Logtalk APIs

Release v3.100.0

Paulo Moura

May 14, 2026

CONTENTS

1 Libraries 1
- 1.1 adaptive_boosting_classifier . . . 1
  - 1.1.1 adaptive_boosting_classifier . . . 1
- 1.2 agglomerative_clusterer . . . 3
  - 1.2.1 agglomerative_clusterer . . . 3
- 1.3 amqp . . . 4
  - 1.3.1 amqp . . . 4
  - 1.3.2 amqp_pool . . . 24
- 1.4 anomaly_detection_protocols . . . 29
  - 1.4.1 anomaly_dataset_protocol . . . 30
  - 1.4.2 anomaly_detector_common . . . 32
  - 1.4.3 anomaly_detector_protocol . . . 37
- 1.5 application . . . 44
  - 1.5.1 application_common . . . 45
  - 1.5.2 application_protocol . . . 46
- 1.6 apriori_pattern_miner . . . 57
  - 1.6.1 apriori_pattern_miner . . . 57
- 1.7 arbitrary . . . 58
  - 1.7.1 arbitrary . . . 59
- 1.8 arrangements . . . 65
  - 1.8.1 arrangements . . . 65
  - 1.8.2 arrangements_protocol . . . 66
- 1.9 assertions . . . 76
  - 1.9.1 assertions . . . 76
  - 1.9.2 assertions(Mode) . . . 78
  - 1.9.3 assertions_messages . . . 80
- 1.10 assignvars . . . 81
  - 1.10.1 assignvars . . . 81
  - 1.10.2 assignvarsp . . . 82
- 1.11 avro . . . 86
  - 1.11.1 avro . . . 86
- 1.12 base32 . . . 88
  - 1.12.1 base32 . . . 89
- 1.13 base58 . . . 90
  - 1.13.1 base58 . . . 91
- 1.14 base64 . . . 92
  - 1.14.1 base64 . . . 93
  - 1.14.2 base64url . . . 94
- 1.15 base85 . . . 96
  - 1.15.1 base85 . . . 96

1.16 bayesian_ridge_regression . . . 98
1.16.1 bayesian_ridge_regression . . . 98
1.17 borda_ranker . . . 101
1.17.1 borda_ranker . . . 101
1.18 bradley_terry_ranker . . . 103
1.18.1 bradley_terry_ranker . . . 103
1.19 c45_classifier . . . 105
1.19.1 c45_classifier . . . 105
1.20 cartesian_products . . . 108
1.20.1 cartesian_products . . . 108
1.20.2 cartesian_products_protocol . . . 109
1.21 cbor . . . 119
1.21.1 cbor . . . 119
1.21.2 cbor(StringRepresentation) . . . 120
1.22 ccsds_frames . . . 122
1.22.1 ccsds_aos_frames(FrameLength,InsertZoneLength,HasOCF,HasFECF) . . . 122
1.22.2 ccsds_frame_protocol . . . 126
1.22.3 ccsds_frames . . . 132
1.22.4 ccsds_frames_types . . . 138
1.22.5 ccsds_tc_frames(FrameLength,SegmentHeaderLength,HasFECF) . . . 139
1.22.6 ccsds_tm_frames(FrameLength,SecondaryHeaderLength,HasFECF) . . . 143
1.23 ccsds_link_profiles . . . 148
1.23.1 ccsds_link_profiles . . . 148
1.24 ccsds_packet_services . . . 158
1.24.1 ccsds_packet_services . . . 158
1.25 ccsds_packetization . . . 173
1.25.1 ccsds_packetization . . . 173
1.26 ccsds_packets . . . 179
1.26.1 ccsds_packets . . . 179
1.26.2 ccsds_packets(SecondaryHeaderLength) . . . 180
1.26.3 ccsds_packets_types . . . 187
1.27 ccsds_tc_services . . . 188
1.27.1 ccsds_tc_services . . . 188
1.28 ccsds_time_codes . . . 201
1.28.1 ccsds_ccs(CalendarVariant,FractionOctets) . . . 201
1.28.2 ccsds_cds(DaySegmentOctets,SubmillisecondOctets,Epoch) . . . 204
1.28.3 ccsds_cuc(CoarseOctets,FineOctets,Epoch) . . . 207
1.28.4 ccsds_time_code_protocol . . . 209
1.28.5 ccsds_time_codes . . . 214
1.28.6 ccsds_time_codes_types . . . 216
1.29 ccsds_time_fields . . . 217
1.29.1 ccsds_time_fields . . . 217
1.30 character_sets . . . 221
1.30.1 character_set . . . 221
1.30.2 character_set_protocol . . . 225
1.30.3 iso_8859_1 . . . 229
1.30.4 iso_8859_10 . . . 230
1.30.5 iso_8859_13 . . . 231
1.30.6 iso_8859_14 . . . 232
1.30.7 iso_8859_15 . . . 233
1.30.8 iso_8859_16 . . . 235
1.30.9 iso_8859_2 . . . 236
1.30.10 iso_8859_3 . . . 237
1.30.11 iso_8859_4 . . . 238

1.30.12 iso_8859_9 . . . 239
1.30.13 mapped_single_byte_character_set . . . 241
1.30.14 single_byte_character_set(MaxCode) . . . 243
1.30.15 us_ascii . . . 244
1.30.16 utf_16_character_set(Endian) . . . 245
1.30.17 utf_16be . . . 246
1.30.18 utf_16le . . . 247
1.30.19 utf_32_character_set(Endian) . . . 249
1.30.20 utf_32be . . . 250
1.30.21 utf_32le . . . 251
1.30.22 utf_8 . . . 252
1.30.23 utf_8_character_set . . . 253
1.30.24 windows_1250 . . . 255
1.30.25 windows_1251 . . . 256
1.30.26 windows_1252 . . . 257
1.30.27 windows_1253 . . . 258
1.30.28 windows_1254 . . . 259
1.30.29 windows_1257 . . . 261
1.31 classification_protocols . . . 262
1.31.1 classifier_common . . . 262
1.31.2 classifier_protocol . . . 272
1.31.3 dataset_protocol . . . 278
1.31.4 probabilistic_classifier_common . . . 281
1.31.5 probabilistic_classifier_protocol . . . 283
1.32 clo_span_pattern_miner . . . 284
1.32.1 clo_span_pattern_miner . . . 285
1.33 clustering_protocols . . . 286
1.33.1 clusterer_common . . . 286
1.33.2 clusterer_protocol . . . 297
1.33.3 clustering_dataset_protocol . . . 303
1.33.4 search_indexing . . . 305
1.34 code_metrics . . . 309
1.34.1 cc_metric . . . 309
1.34.2 code_metric . . . 311
1.34.3 code_metrics . . . 324
1.34.4 code_metrics_messages . . . 325
1.34.5 code_metrics_utilities . . . 326
1.34.6 cogc_metric . . . 331
1.34.7 coupling_metric . . . 333
1.34.8 dit_metric . . . 335
1.34.9 doc_metric . . . 336
1.34.10 halstead_metric . . . 341
1.34.11 halstead_metric(Stroud) . . . 342
1.34.12 lcom_metric . . . 344
1.34.13 lines_metric . . . 345
1.34.14 mi_metric . . . 347
1.34.15 mi_metric(Stroud) . . . 348
1.34.16 noc_metric . . . 350
1.34.17 nor_metric . . . 351
1.34.18 rfc_metric . . . 353
1.34.19 size_metric . . . 354
1.34.20 upn_metric . . . 356
1.34.21 wmc_metric . . . 357
1.35 colley_ranker . . . 359

1.35.1 colley_ranker . . . 359
1.36 combinations . . . 361
1.36.1 combinations . . . 361
1.36.2 combinations_protocol . . . 362
1.37 command_line_options . . . 372
1.37.1 command_line_option . . . 372
1.37.2 command_line_options . . . 376
1.38 copeland_ranker . . . 380
1.38.1 copeland_ranker . . . 380
1.39 core . . . 382
1.39.1 core_messages . . . 382
1.39.2 expanding . . . 383
1.39.3 forwarding . . . 385
1.39.4 logtalk . . . 386
1.39.5 monitoring . . . 403
1.39.6 user . . . 405
1.40 coroutining . . . 406
1.40.1 coroutining . . . 406
1.41 crs_projections . . . 409
1.41.1 crs_projections . . . 410
1.41.2 crs_projections_protocol . . . 411
1.42 csv . . . 419
1.42.1 csv . . . 419
1.42.2 csv(Header,Separator,IgnoreQuotes) . . . 420
1.42.3 csv(Header,Separator,IgnoreQuotes,Comments) . . . 422
1.42.4 csv_guess_questions . . . 423
1.42.5 csv_protocol . . . 424
1.43 cuid2 . . . 435
1.43.1 cuid2 . . . 435
1.43.2 cuid2(Representation,Size,Alphabet) . . . 436
1.43.3 cuid2_protocol . . . 438
1.44 cusum_anomaly_detector . . . 440
1.44.1 cusum_anomaly_detector . . . 440
1.45 datalog . . . 441
1.45.1 datalog . . . 441
1.45.2 datalog_protocol . . . 451
1.46 dates . . . 459
1.46.1 date . . . 459
1.46.2 datep . . . 460
1.46.3 time . . . 471
1.46.4 timep . . . 473
1.47 dates_tz . . . 475
1.47.1 dates_tz . . . 475
1.47.2 dates_tz_protocol . . . 477
1.48 dbscan_clusterer . . . 480
1.48.1 dbscan_clusterer . . . 480
1.49 dead_code_scanner . . . 482
1.49.1 dead_code_scanner . . . 482
1.49.2 dead_code_scanner_messages . . . 490
1.50 debug_messages . . . 492
1.50.1 debug_messages . . . 492
1.51 debugger . . . 496
1.51.1 debugger . . . 496
1.51.2 debugger_messages . . . 506

1.51.3 debuggerp . . . 507
1.51.4 dump_trace . . . 519
1.52 dependents . . . 521
1.52.1 observer . . . 521
1.52.2 subject . . . 522
1.53 deques . . . 526
1.53.1 deque . . . 526
1.53.2 deque_protocol . . . 527
1.54 derangements . . . 533
1.54.1 derangements . . . 533
1.54.2 derangements_protocol . . . 534
1.55 diagrams . . . 544
1.55.1 caller_diagram . . . 544
1.55.2 caller_diagram(Format) . . . 545
1.55.3 cytoscapejs_graph_language . . . 548
1.55.4 d2_graph_language . . . 552
1.55.5 diagram(Format) . . . 554
1.55.6 diagrams . . . 578
1.55.7 diagrams(Format) . . . 579
1.55.8 directory_dependency_diagram . . . 589
1.55.9 directory_dependency_diagram(Format) . . . 591
1.55.10 directory_diagram(Format) . . . 592
1.55.11 directory_load_diagram . . . 596
1.55.12 directory_load_diagram(Format) . . . 597
1.55.13 dot_graph_language . . . 599
1.55.14 entity_diagram . . . 601
1.55.15 entity_diagram(Format) . . . 602
1.55.16 file_dependency_diagram . . . 606
1.55.17 file_dependency_diagram(Format) . . . 607
1.55.18 file_diagram(Format) . . . 609
1.55.19 file_load_diagram . . . 613
1.55.20 file_load_diagram(Format) . . . 614
1.55.21 graph_language_protocol . . . 616
1.55.22 graph_language_registry . . . 620
1.55.23 inheritance_diagram . . . 621
1.55.24 inheritance_diagram(Format) . . . 623
1.55.25 library_dependency_diagram . . . 624
1.55.26 library_dependency_diagram(Format) . . . 626
1.55.27 library_diagram(Format) . . . 627
1.55.28 library_load_diagram . . . 632
1.55.29 library_load_diagram(Format) . . . 633
1.55.30 mermaid_graph_language . . . 635
1.55.31 modules_diagram_support . . . 637
1.55.32 plantuml_graph_language . . . 639
1.55.33 uses_diagram . . . 640
1.55.34 uses_diagram(Format) . . . 642
1.55.35 xref_diagram . . . 643
1.55.36 xref_diagram(Format) . . . 645
1.56 dictionaries . . . 648
1.56.1 avltree . . . 648
1.56.2 bintree . . . 650
1.56.3 dictionaryp . . . 652
1.56.4 rbtree . . . 665
1.56.5 splaytree . . . 666

1.56.6 two3tree . . . 668
1.57 dif . . . 680
1.57.1 dif . . . 680
1.58 dimension_reduction_protocols . . . 682
1.58.1 dimension_reducer_common . . . 682
1.58.2 dimension_reducer_protocol . . . 696
1.58.3 dimension_reduction_dataset_protocol . . . 702
1.58.4 supervised_dimension_reduction_dataset_protocol . . . 704
1.58.5 target_supervised_dimension_reduction_dataset_protocol . . . 706
1.59 doclet . . . 708
1.59.1 doclet . . . 708
1.60 eclat_pattern_miner . . . 710
1.60.1 eclat_pattern_miner . . . 711
1.61 edcg . . . 712
1.61.1 edcg . . . 712
1.62 elastic_net_regression . . . 716
1.62.1 elastic_net_regression . . . 716
1.63 elo_ranker . . . 717
1.63.1 elo_ranker . . . 718
1.64 events . . . 719
1.64.1 after_event_registry . . . 719
1.64.2 before_event_registry . . . 721
1.64.3 event_registry . . . 722
1.64.4 event_registryp . . . 723
1.64.5 monitor . . . 727
1.64.6 monitorp . . . 729
1.65 ewma_anomaly_detector . . . 732
1.65.1 ewma_anomaly_detector . . . 732
1.66 expand_library_alias_paths . . . 734
1.66.1 expand_library_alias_paths . . . 734
1.67 expecteds . . . 735
1.67.1 either . . . 736
1.67.2 expected . . . 739
1.67.3 expected(Expected) . . . 744
1.68 fcube . . . 757
1.68.1 fcube . . . 757
1.69 flags . . . 761
1.69.1 flags . . . 761
1.69.2 flags_validator . . . 769
1.70 format . . . 771
1.70.1 format . . . 771
1.71 fp_growth_pattern_miner . . . 773
1.71.1 fp_growth_pattern_miner . . . 773
1.72 frequent_pattern_mining_protocols . . . 774
1.72.1 frequent_pattern_mining_common . . . 774
1.72.2 transaction_dataset_protocol . . . 778
1.73 gaussian_mixture_clusterer . . . 780
1.73.1 gaussian_mixture_clusterer . . . 780
1.74 gaussian_process_regression . . . 782
1.74.1 gaussian_process_regression . . . 782
1.75 genint . . . 784
1.75.1 genint . . . 784
1.75.2 genint_core . . . 785
1.76 gensym . . . 788

1.76.1 gensym . . . 788
1.76.2 gensym_core . . . 789
1.77 geohash . . . 792
1.77.1 geohash . . . 792
1.77.2 geohash_protocol . . . 794
1.78 geojson . . . 807
1.78.1 geojson . . . 807
1.78.2 geojson(ObjectRepresentation,PairRepresentation,StringRepresentation) . . . 808
1.78.3 geojson_protocol . . . 810
1.79 geospatial . . . 815
1.79.1 geospatial . . . 815
1.79.2 geospatial_protocol . . . 818
1.80 git . . . 844
1.80.1 git . . . 844
1.80.2 git_protocol . . . 845
1.81 glicko2_periodic_ranker . . . 849
1.81.1 glicko2_periodic_ranker . . . 849
1.82 glicko2_ranker . . . 851
1.82.1 glicko2_ranker . . . 851
1.83 gradient_boosting_classifier . . . 853
1.83.1 gradient_boosting_classifier . . . 853
1.84 gradient_boosting_regression . . . 855
1.84.1 gradient_boosting_regression . . . 855
1.85 grammars . . . 856
1.85.1 blank_grammars(Format) . . . 856
1.85.2 ip_grammars(Format) . . . 862
1.85.3 number_grammars(Format) . . . 864
1.85.4 sequence_grammars . . . 870
1.86 graphs . . . 874
1.86.1 directed_graph_common . . . 874
1.86.2 directed_graph_protocol . . . 875
1.86.3 graph_common . . . 881
1.86.4 graph_protocol . . . 884
1.86.5 graph_types . . . 896
1.86.6 undirected_graph_common . . . 898
1.86.7 unweighted_directed_graph . . . 902
1.86.8 unweighted_directed_graph(Dictionary) . . . 904
1.86.9 unweighted_graph_common(Dictionary) . . . 907
1.86.10 unweighted_graph_protocol . . . 908
1.86.11 unweighted_undirected_graph . . . 911
1.86.12 unweighted_undirected_graph(Dictionary) . . . 913
1.86.13 weighted_directed_graph . . . 915
1.86.14 weighted_directed_graph(Dictionary) . . . 916
1.86.15 weighted_graph_common(Dictionary) . . . 918
1.86.16 weighted_graph_protocol . . . 922
1.86.17 weighted_undirected_graph . . . 925
1.86.18 weighted_undirected_graph(Dictionary) . . . 927
1.87 gsp_pattern_miner . . . 930
1.87.1 gsp_pattern_miner . . . 930
1.88 hashes . . . 932
1.88.1 crc32_non_reflected(Polynomial,Initial,FinalXor,AppendLength) . . . 932
1.88.2 crc32_reflected(Polynomial) . . . 933
1.88.3 crc32b . . . 935
1.88.4 crc32bzip2 . . . 936

1.88.5 crc32c . . . 937
1.88.6 crc32mpeg2 . . . 939
1.88.7 crc32posix . . . 940
1.88.8 crc32q . . . 941
1.88.9 djb2_32 . . . 943
1.88.10 djb2_64 . . . 944
1.88.11 fips202_fixed_hash(RateBytes,Suffix,OutputBytes) . . . 945
1.88.12 fips202_hash(RateBytes,Suffix,OutputBytes) . . . 947
1.88.13 fnv1a_32 . . . 949
1.88.14 fnv1a_64 . . . 950
1.88.15 hash_common_32 . . . 951
1.88.16 hash_common_64 . . . 958
1.88.17 hash_digest_protocol . . . 964
1.88.18 hash_protocol . . . 966
1.88.19 md5 . . . 968
1.88.20 murmurhash3_x64_128 . . . 969
1.88.21 murmurhash3_x86_128 . . . 970
1.88.22 murmurhash3_x86_32 . . . 972
1.88.23 sdbm_32 . . . 973
1.88.24 sdbm_64 . . . 974
1.88.25 sha1 . . . 976
1.88.26 sha256 . . . 977
1.88.27 sha3_224 . . . 979
1.88.28 sha3_256 . . . 980
1.88.29 sha3_384 . . . 981
1.88.30 sha3_512 . . . 983
1.88.31 shake128(OutputBytes) . . . 984
1.88.32 shake256(OutputBytes) . . . 985
1.88.33 siphash_2_4 . . . 987
1.88.34 siphash_2_4(Key) . . . 988
1.89 hdbscan_clusterer . . . 989
1.89.1 hdbscan_clusterer . . . 989
1.90 heaps . . . 991
1.90.1 binary_heap(Order) . . . 991
1.90.2 binary_heap_max . . . 992
1.90.3 binary_heap_min . . . 994
1.90.4 heap(Order) . . . 995
1.90.5 heap_protocol . . . 996
1.90.6 heapp . . . 1002
1.90.7 maxheap . . . 1003
1.90.8 minheap . . . 1004
1.90.9 pairing_heap(Order) . . . 1006
1.90.10 pairing_heap_max . . . 1007
1.90.11 pairing_heap_min . . . 1009
1.91 help . . . 1010
1.91.1 help . . . 1010
1.92 hierarchical_clustering . . . 1018
1.92.1 hierarchical_clustering . . . 1019
1.93 hierarchies . . . 1021
1.93.1 class_hierarchy . . . 1021
1.93.2 class_hierarchyp . . . 1022
1.93.3 hierarchyp . . . 1029
1.93.4 proto_hierarchy . . . 1032
1.93.5 proto_hierarchyp . . . 1034

1.94 hmac . . . 1036
1.94.1 hmac . . . 1037
1.94.2 hmac_protocol . . . 1038
1.95 hodge_rank . . . 1041
1.95.1 hodge_rank . . . 1041
1.96 hook_flows . . . 1043
1.96.1 hook_pipeline(Pipeline) . . . 1043
1.96.2 hook_set(Set) . . . 1045
1.97 hook_objects . . . 1046
1.97.1 backend_adapter_hook . . . 1046
1.97.2 default_workflow_hook . . . 1048
1.97.3 grammar_rules_hook . . . 1049
1.97.4 identity_hook . . . 1050
1.97.5 object_wrapper_hook . . . 1052
1.97.6 object_wrapper_hook(Protocol) . . . 1053
1.97.7 object_wrapper_hook(Name,Relations) . . . 1055
1.97.8 print_goal_hook . . . 1056
1.97.9 prolog_module_hook(Module) . . . 1058
1.97.10 suppress_goal_hook . . . 1059
1.97.11 write_to_file_hook(File) . . . 1060
1.97.12 write_to_file_hook(File,Options) . . . 1062
1.97.13 write_to_stream_hook(Stream) . . . 1063
1.97.14 write_to_stream_hook(Stream,Options) . . . 1065
1.98 html . . . 1066
1.98.1 html . . . 1066
1.98.2 html5 . . . 1069
1.98.3 xhtml11 . . . 1070
1.99 ica_projection . . . 1071
1.99.1 ica_projection . . . 1071
1.100 ids . . . 1073
1.100.1 ids . . . 1073
1.100.2 ids(Representation,Bytes) . . . 1074
1.101 ieee_754 . . . 1076
1.101.1 ieee_754(Precision,ByteOrder,NaNRepresentation) . . . 1076
1.101.2 ieee_754_common(Precision,ByteOrder) . . . 1078
1.101.3 ieee_754_fields(Precision,ByteOrder) . . . 1086
1.101.4 ieee_754_fields_protocol . . . 1087
1.101.5 ieee_754_protocol . . . 1093
1.102 intervals . . . 1098
1.102.1 interval . . . 1098
1.102.2 interval_algebra . . . 1100
1.102.3 interval_algebra_protocol . . . 1101
1.102.4 interval_constraint_network . . . 1103
1.102.5 interval_constraint_network_protocol . . . 1105
1.102.6 interval_protocol . . . 1116
1.102.7 interval_relation_set . . . 1123
1.102.8 interval_relation_set_protocol . . . 1125
1.103 iqr_anomaly_detector . . . 1130
1.103.1 iqr_anomaly_detector . . . 1130
1.104 iso8601 . . . 1132
1.104.1 iso8601 . . . 1132
1.105 iso_13616 . . . 1146
1.105.1 iso_13616 . . . 1146
1.105.2 iso_13616_protocol . . . 1148

1.105.3 iso_13616_registry . . . 1150
1.106 iso_3166 . . . 1152
1.106.1 iso_3166 . . . 1152
1.106.2 iso_3166_1 . . . 1153
1.106.3 iso_3166_2 . . . 1155
1.106.4 iso_3166_protocol . . . 1156
1.107 iso_4217 . . . 1158
1.107.1 iso_4217 . . . 1158
1.107.2 iso_4217_protocol . . . 1160
1.108 iso_639 . . . 1162
1.108.1 iso_639 . . . 1162
1.108.2 iso_639_1 . . . 1164
1.108.3 iso_639_2 . . . 1165
1.108.4 iso_639_3 . . . 1167
1.108.5 iso_639_5 . . . 1169
1.108.6 iso_639_protocol . . . 1170
1.109 iso_9362 . . . 1173
1.109.1 iso_9362 . . . 1173
1.109.2 iso_9362_protocol . . . 1174
1.110 isolation_forest_anomaly_detector . . . 1177
1.110.1 isolation_forest_anomaly_detector . . . 1177
1.111 issue_creator . . . 1179
1.111.1 issue_creator . . . 1179
1.112 java . . . 1181
1.112.1 java . . . 1181
1.112.2 java(Reference) . . . 1182
1.112.3 java(Reference,ReturnValue) . . . 1183
1.112.4 java_access_protocol . . . 1185
1.112.5 java_hook . . . 1188
1.112.6 java_utils_protocol . . . 1190
1.113 json . . . 1199
1.113.1 json . . . 1199
1.113.2 json(StringRepresentation) . . . 1200
1.113.3 json(ObjectRepresentation,PairRepresentation,StringRepresentation) . . . 1201
1.113.4 json_protocol . . . 1203
1.114 json_ld . . . 1205
1.114.1 json_ld . . . 1205
1.114.2 json_ld(StringRepresentation) . . . 1206
1.114.3 json_ld(ObjectRepresentation,PairRepresentation,StringRepresentation) . . . 1207
1.114.4 json_ld_protocol . . . 1209
1.115 json_lines . . . 1212
1.115.1 json_lines . . . 1213
1.115.2 json_lines(StringRepresentation) . . . 1214
1.115.3 json_lines(ObjectRepresentation,PairRepresentation,StringRepresentation) . . . 1215
1.115.4 json_lines_protocol . . . 1217
1.116 json_pointer . . . 1219
1.116.1 json_pointer . . . 1219
1.116.2 json_pointer(StringRepresentation) . . . 1220
1.116.3 json_pointer_protocol . . . 1221
1.117 json_rpc . . . 1226
1.117.1 json_rpc . . . 1226
1.118 json_schema . . . 1239
1.118.1 json_schema . . . 1239
1.118.2 json_schema(StringRepresentation) . . . 1240

1.118.3 json_schema(ObjectRepresentation,PairRepresentation,StringRepresentation) 1242
1.118.4 json_schema_protocol 1243
1.119 kcenters_clusterer 1245
1.119.1 kcenters_clusterer 1245
1.120 kemeny_young_ranker 1247
1.120.1 kemeny_young_ranker 1247
1.121 kernel_pca_projection 1249
1.121.1 kernel_pca_projection 1250
1.122 kernel_svm_classifier 1251
1.122.1 kernel_svm_classifier 1251
1.123 kmeans_clusterer 1253
1.123.1 kmeans_clusterer 1253
1.124 kmedians_clusterer 1254
1.124.1 kmedians_clusterer 1254
1.125 kmedoids_clusterer 1256
1.125.1 kmedoids_clusterer 1256
1.126 kmodes_clusterer 1257
1.126.1 kmodes_clusterer 1258
1.127 knn_classifier 1259
1.127.1 knn_classifier 1259
1.128 knn_distance_anomaly_detector 1261
1.128.1 knn_distance_anomaly_detector 1262
1.129 knn_regression 1263
1.129.1 knn_regression 1263
1.130 kprototypes_clusterer 1265
1.130.1 kprototypes_clusterer 1265
1.131 ksuid 1266
1.131.1 ksuid 1266
1.131.2 ksuid(Representation,Alphabet) 1268
1.131.3 ksuid_protocol 1269
1.132 lasso_regression 1271
1.132.1 lasso_regression 1271
1.133 lda_classifier 1272
1.133.1 lda_classifier 1272
1.134 lda_projection 1274
1.134.1 lda_projection 1274
1.135 lgtdoc 1276
1.135.1 lgtdoc 1276
1.135.2 lgtdoc_messages 1280
1.135.3 lgtdocp 1281
1.136 lgtunit 1293
1.136.1 automation_report 1293
1.136.2 cobertura_report 1294
1.136.3 coverage_report 1298
1.136.4 ctrf_output 1300
1.136.5 ctrf_report 1301
1.136.6 lcov_report 1303
1.136.7 lgtunit 1305
1.136.8 lgtunit_messages 1355
1.136.9 minimal_output 1357
1.136.10subunit_v1_output 1358
1.136.11subunit_v1_report 1359
1.136.12subunit_v2_output 1361
1.136.13subunit_v2_report 1362

1.136.14tap_output . . . 1363
1.136.15tap_report . . . 1366
1.136.16xunit_net_v2_output . . . 1368
1.136.17xunit_net_v2_report . . . 1369
1.136.18xunit_output . . . 1371
1.136.19xunit_report . . . 1373
1.137 library . . . 1374
1.137.1 cloning . . . 1374
1.137.2 counters . . . 1376
1.137.3 streamvars . . . 1379
1.138 linda . . . 1382
1.138.1 linda . . . 1382
1.138.2 linda_client . . . 1384
1.138.3 linda_server . . . 1397
1.139 linear_algebra . . . 1405
1.139.1 linear_algebra . . . 1405
1.139.2 linear_algebra_protocol . . . 1407
1.140 linear_regression . . . 1444
1.140.1 linear_regression . . . 1444
1.141 linear_svm_classifier . . . 1446
1.141.1 linear_svm_classifier . . . 1446
1.142 linter_reporter . . . 1447
1.142.1 linter_reporter . . . 1447
1.143 listing . . . 1453
1.143.1 listing . . . 1453
1.144 lof_anomaly_detector . . . 1456
1.144.1 lof_anomaly_detector . . . 1456
1.145 logging . . . 1458
1.145.1 logger . . . 1458
1.145.2 logging . . . 1460
1.145.3 loggingp . . . 1462
1.146 logistic_regression_classifier . . . 1466
1.146.1 logistic_regression_classifier . . . 1466
1.147 loops . . . 1468
1.147.1 loop . . . 1468
1.147.2 loopp . . . 1469
1.148 massey_ranker . . . 1475
1.148.1 massey_ranker . . . 1475
1.149 mcp_server . . . 1476
1.149.1 mcp_prompt_protocol . . . 1477
1.149.2 mcp_resource_protocol . . . 1479
1.149.3 mcp_server . . . 1481
1.149.4 mcp_tool_protocol . . . 1488
1.150 memcached . . . 1492
1.150.1 memcached . . . 1492
1.151 message_pack . . . 1505
1.151.1 message_pack(StringRepresentation) . . . 1505
1.152 meta . . . 1507
1.152.1 meta . . . 1507
1.152.2 metap . . . 1509
1.153 meta_compiler . . . 1519
1.153.1 meta_compiler . . . 1520
1.154 metagol . . . 1521
1.154.1 metagol . . . 1522

1.154.2 metagol_example_protocol . . . 1529
1.155 mime_types . . . 1531
1.155.1 mime_types . . . 1531
1.156 modified_z_score_anomaly_detector . . . 1539
1.156.1 modified_z_score_anomaly_detector . . . 1539
1.157 multisets . . . 1541
1.157.1 multisets . . . 1541
1.157.2 multisets_protocol . . . 1542
1.158 mutation_testing . . . 1552
1.158.1 arithmetic_operator_replacement(Entity,Predicate,ClauseIndex,Occurrence, PrintMutation) . . . 1552
1.158.2 body_goal_negation(Entity,Predicate,ClauseIndex,Occurrence,PrintMutation) . . . 1554
1.158.3 clause_mutator_protocol . . . 1555
1.158.4 clauses_reordering(Entity,Predicate,ClauseIndex,Occurrence,PrintMutation) . . . 1556
1.158.5 directive_mutator_protocol . . . 1558
1.158.6 fail_insertion(Entity,Predicate,ClauseIndex,Occurrence,PrintMutation) . . . 1559
1.158.7 head_arguments_mutation(Entity,Predicate,ClauseIndex,Occurrence,PrintMutation) 1561
1.158.8 head_arguments_reordering(Entity,Predicate,ClauseIndex,Occurrence, PrintMutation) . . . 1562
1.158.9 mutation_testing . . . 1564
1.158.10 mutation_testing_messages . . . 1575
1.158.11 mutator_common . . . 1577
1.158.12 mutator_protocol . . . 1585
1.158.13 predicate_directive_suppression(Entity,Predicate,DirectiveIndex,Occurrence, PrintMutation) . . . 1587
1.158.14 relational_operator_replacement(Entity,Predicate,ClauseIndex,Occurrence, PrintMutation) . . . 1589
1.158.15 scope_directive_replacement(Entity,Predicate,DirectiveIndex,Occurrence, PrintMutation) . . . 1590
1.158.16 subprocess_coverage_hook . . . 1592
1.158.17 subprocess_mutation_hook . . . 1594
1.158.18 truth_literal_flip(Entity,Predicate,ClauseIndex,Occurrence,PrintMutation) . . . 1596
1.158.19 uses_directive_resource_deletion(Entity,Predicate,DirectiveIndex,Occurrence, PrintMutation) . . . 1598
1.159 mutations . . . 1599
1.159.1 default_atom_mutations . . . 1599
1.159.2 default_compound_mutations . . . 1601
1.159.3 default_float_mutations . . . 1602
1.159.4 default_integer_mutations . . . 1603
1.159.5 default_list_mutations . . . 1605
1.159.6 mutations . . . 1606
1.159.7 mutations_store . . . 1608
1.160 naive_bayes_classifier . . . 1611
1.160.1 naive_bayes_classifier . . . 1611
1.161 nanoid . . . 1612
1.161.1 nanoid . . . 1613
1.161.2 nanoid(Representation,Size,Alphabet) . . . 1614
1.161.3 nanoid_protocol . . . 1615
1.162 nearest_centroid_classifier . . . 1617
1.162.1 nearest_centroid_classifier . . . 1617
1.163 nested_dictionaries . . . 1619
1.163.1 navltree . . . 1620
1.163.2 nbintree . . . 1621
1.163.3 nested_dictionary_protocol . . . 1622

1.163.4 nrbtree . . . 1627
1.164 nmea . . . 1628
1.164.1 nmea . . . 1629
1.164.2 nmea_protocol . . . 1630
1.165 nmf_projection . . . 1634
1.165.1 nmf_projection . . . 1634
1.166 optics_clusterer . . . 1635
1.166.1 optics_clusterer . . . 1636
1.167 optionals . . . 1637
1.167.1 maybe . . . 1637
1.167.2 optional . . . 1640
1.167.3 optional(Optional) . . . 1645
1.168 options . . . 1655
1.168.1 options . . . 1655
1.168.2 options_protocol . . . 1656
1.169 os . . . 1662
1.169.1 os . . . 1662
1.169.2 os_types . . . 1664
1.169.3 osp . . . 1666
1.170 packs . . . 1684
1.170.1 pack_protocol . . . 1685
1.170.2 packs . . . 1689
1.170.3 packs_common . . . 1725
1.170.4 packs_messages . . . 1737
1.170.5 packs_specs_hook . . . 1738
1.170.6 registries . . . 1740
1.170.7 registry_loader_hook . . . 1752
1.170.8 registry_protocol . . . 1753
1.171 partitions . . . 1757
1.171.1 partitions . . . 1757
1.171.2 partitions_protocol . . . 1758
1.172 pattern_mining_protocols . . . 1776
1.172.1 pattern_miner_common . . . 1776
1.172.2 pattern_miner_protocol . . . 1780
1.173 pca_projection . . . 1785
1.173.1 pca_projection . . . 1786
1.174 pddl_parser . . . 1787
1.174.1 pddl . . . 1787
1.174.2 read_file . . . 1790
1.175 permutations . . . 1792
1.175.1 permutations . . . 1792
1.175.2 permutations_protocol . . . 1793
1.176 plackett_luce_last_ranker . . . 1804
1.176.1 plackett_luce_last_ranker . . . 1804
1.177 plackett_luce_ranker . . . 1806
1.177.1 plackett_luce_ranker . . . 1806
1.178 pls_projection . . . 1808
1.178.1 pls_projection . . . 1808
1.179 ports_profiler . . . 1810
1.179.1 ports_profiler . . . 1810
1.180 prefix_span_pattern_miner . . . 1816
1.180.1 prefix_span_pattern_miner . . . 1816
1.181 probabilistic_pca_projection . . . 1818
1.181.1 probabilistic_pca_projection . . . 1818

1.182 process . . . 1819
1.182.1 process . . . 1819
1.183 protobuf . . . 1822
1.183.1 protobuf . . . 1822
1.184 qda_classifier . . . 1825
1.184.1 qda_classifier . . . 1825
1.185 queues . . . 1827
1.185.1 queue . . . 1827
1.185.2 queuep . . . 1828
1.186 random . . . 1835
1.186.1 backend_random . . . 1835
1.186.2 fast_random . . . 1837
1.186.3 fast_random(Algorithm) . . . 1838
1.186.4 pseudo_random_protocol . . . 1841
1.186.5 random . . . 1843
1.186.6 random(Algorithm) . . . 1845
1.186.7 random_protocol . . . 1848
1.186.8 sampling_protocol . . . 1857
1.187 random_forest_classifier . . . 1870
1.187.1 random_forest_classifier . . . 1870
1.188 random_forest_regression . . . 1871
1.188.1 random_forest_regression . . . 1872
1.189 random_projection . . . 1873
1.189.1 random_projection . . . 1873
1.190 rank_centrality . . . 1875
1.190.1 rank_centrality . . . 1875
1.191 ranked_pairs . . . 1876
1.191.1 ranked_pairs . . . 1877
1.192 ranking_protocols . . . 1879
1.192.1 condorcet_victory_common . . . 1879
1.192.2 glicko2_common . . . 1882
1.192.3 grouped_strength_ranker_common . . . 1885
1.192.4 pairwise_measurement_dataset_protocol . . . 1889
1.192.5 pairwise_ranking_dataset_protocol . . . 1891
1.192.6 pairwise_strength_ranker_common . . . 1893
1.192.7 plackett_luce_common . . . 1899
1.192.8 ranker_common . . . 1902
1.192.9 ranker_protocol . . . 1906
1.192.10 ranking_dataset_common . . . 1912
1.192.11 ranking_dataset_protocol . . . 1927
1.192.12 score_ranker_common . . . 1929
1.192.13 score_ranker_model_common . . . 1931
1.192.14 temporal_pairwise_ranking_dataset_protocol . . . 1937
1.193 reader . . . 1939
1.193.1 reader . . . 1939
1.194 recorded_database . . . 1948
1.194.1 recorded_database . . . 1948
1.194.2 recorded_database_core . . . 1949
1.195 redis . . . 1955
1.195.1 redis . . . 1955
1.196 regression_protocols . . . 1976
1.196.1 regression_dataset_protocol . . . 1976
1.196.2 regressor_common . . . 1979
1.196.3 regressor_protocol . . . 2008

1.197 regression_tree . . . 2013
1.197.1 regression_tree . . . 2013
1.198 regularized_bradley_terry_ranker . . . 2015
1.198.1 regularized_bradley_terry_ranker . . . 2015
1.199 ridge_regression . . . 2016
1.199.1 ridge_regression . . . 2017
1.200 sarif . . . 2018
1.200.1 sarif . . . 2018
1.201 sbom . . . 2021
1.201.1 sbom . . . 2021
1.202 schulze_ranker . . . 2024
1.202.1 schulze_ranker . . . 2025
1.203 sequential_pattern_mining_protocols . . . 2027
1.203.1 sequence_dataset_protocol . . . 2027
1.203.2 sequential_pattern_mining_common . . . 2029
1.204 sets . . . 2033
1.204.1 set . . . 2033
1.204.2 set(Type) . . . 2035
1.204.3 setp . . . 2037
1.204.4 treap_set . . . 2047
1.205 sgd_classifier . . . 2048
1.205.1 sgd_classifier . . . 2049
1.206 simulated_annealing . . . 2050
1.206.1 simulated_annealing(Problem) . . . 2050
1.206.2 simulated_annealing(Problem,RandomAlgorithm) . . . 2052
1.206.3 simulated_annealing_protocol . . . 2056
1.207 snowflakeid . . . 2061
1.207.1 snowflakeid . . . 2061
1.207.2 snowflakeid(Representation,EpochMilliseconds,TimeUnitMilliseconds, TimestampBits,NodeBits,SequenceBits,Node) . . . 2062
1.207.3 snowflakeid_instagram . . . 2064
1.207.4 snowflakeid_instagram(Representation) . . . 2066
1.207.5 snowflakeid_protocol . . . 2067
1.207.6 snowflakeid_sonyflake . . . 2069
1.207.7 snowflakeid_sonyflake(Representation) . . . 2070
1.207.8 snowflakeid_twitter . . . 2071
1.207.9 snowflakeid_twitter(Representation) . . . 2073
1.208 sockets . . . 2074
1.208.1 socket . . . 2074
1.209 spade_pattern_miner . . . 2081
1.209.1 spade_pattern_miner . . . 2081
1.210 statistics . . . 2082
1.210.1 population . . . 2082
1.210.2 sample . . . 2084
1.210.3 statistics . . . 2085
1.210.4 statisticsp . . . 2091
1.211 stemming . . . 2107
1.211.1 lovins_stemmer(Representation) . . . 2108
1.211.2 porter_stemmer(Representation) . . . 2109
1.211.3 stemmer_protocol . . . 2111
1.212 stomp . . . 2113
1.212.1 stomp . . . 2113
1.213 string_distance . . . 2122
1.213.1 string_distance(Representation) . . . 2122

1.214 strings . . . 2132
1.214.1 string(Representation) . . . 2132
1.215 subsequences . . . 2140
1.215.1 subsequences . . . 2140
1.215.2 subsequences_protocol . . . 2142
1.216 term_io . . . 2164
1.216.1 term_io . . . 2165
1.216.2 term_io_protocol . . . 2166
1.217 thurstone_mosteller_ranker . . . 2177
1.217.1 thurstone_mosteller_ranker . . . 2177
1.218 time_scales . . . 2179
1.218.1 time_scales . . . 2179
1.218.2 time_scales_data . . . 2180
1.218.3 time_scales_protocol . . . 2189
1.219 timeout . . . 2201
1.219.1 timeout . . . 2201
1.220 tle_orbits . . . 2203
1.220.1 tle_orbits . . . 2204
1.220.2 tle_orbits_protocol . . . 2205
1.221 toml . . . 2210
1.221.1 toml . . . 2210
1.221.2 toml(StringRepresentation) . . . 2211
1.221.3 toml(ObjectRepresentation,PairRepresentation,StringRepresentation) . . . 2213
1.221.4 toml_protocol . . . 2214
1.222 tool_diagnostics . . . 2217
1.222.1 tool_diagnostics_common . . . 2217
1.222.2 tool_diagnostics_protocol . . . 2219
1.223 toon . . . 2225
1.223.1 toon . . . 2225
1.223.2 toon(StringRepresentation) . . . 2226
1.223.3 toon(ObjectRepresentation,PairRepresentation,StringRepresentation) . . . 2227
1.223.4 toon_protocol . . . 2229
1.224 toychr . . . 2231
1.224.1 toychrdb . . . 2231
1.225 truncated_svd_projection . . . 2236
1.225.1 truncated_svd_projection . . . 2237
1.226 tsv . . . 2238
1.226.1 tsv . . . 2239
1.226.2 tsv(Header) . . . 2240
1.226.3 tsv(Header,Comments) . . . 2241
1.226.4 tsv_protocol . . . 2243
1.227 tutor . . . 2252
1.227.1 tutor . . . 2252
1.227.2 tutor_explanations . . . 2253
1.228 types . . . 2255
1.228.1 atom . . . 2255
1.228.2 atomic . . . 2257
1.228.3 callable . . . 2258
1.228.4 character . . . 2259
1.228.5 characterp . . . 2261
1.228.6 comparingp . . . 2270
1.228.7 compound . . . 2273
1.228.8 difflist . . . 2274
1.228.9 float . . . 2277

1.228.10integer . . . 2279
1.228.11list . . . 2283
1.228.12list(Type) . . . 2285
1.228.13listp . . . 2286
1.228.14natural . . . 2309
1.228.15number . . . 2316
1.228.16numberlist . . . 2319
1.228.17numberlistp . . . 2321
1.228.18pairs . . . 2333
1.228.19term . . . 2338
1.228.20termp . . . 2339
1.228.21type . . . 2346
1.228.22varlist . . . 2352
1.228.23varlistp . . . 2353
1.229 tzif . . . 2363
1.229.1 tzif . . . 2363
1.229.2 tzif_protocol . . . 2365
1.229.3 tzif_zone_ids . . . 2391
1.230 ulid . . . 2393
1.230.1 ulid . . . 2393
1.230.2 ulid(Representation) . . . 2394
1.230.3 ulid_protocol . . . 2396
1.230.4 ulid_types . . . 2399
1.231 union_find . . . 2400
1.231.1 union_find . . . 2400
1.231.2 union_find_protocol . . . 2402
1.232 url . . . 2406
1.232.1 url(Representation) . . . 2406
1.233 uuid . . . 2410
1.233.1 uuid . . . 2410
1.233.2 uuid(Representation) . . . 2411
1.233.3 uuid_protocol . . . 2413
1.234 validations . . . 2417
1.234.1 validated . . . 2417
1.234.2 validation . . . 2422
1.234.3 validation(Validation) . . . 2428
1.235 verdi_neruda . . . 2441
1.235.1 a_star_interpreter(W) . . . 2442
1.235.2 benchmark_generators . . . 2443
1.235.3 best_first . . . 2444
1.235.4 bfs_interpreter . . . 2446
1.235.5 bup_interpreter . . . 2447
1.235.6 counter . . . 2448
1.235.7 databasep . . . 2452
1.235.8 debug_expansion(Mode) . . . 2454
1.235.9 demodb . . . 2456
1.235.10dfs_interpreter . . . 2457
1.235.11flatting . . . 2458
1.235.12heuristic_expansion(Mode) . . . 2460
1.235.13iddfs_interpreter(Increment) . . . 2461
1.235.14interpreterp . . . 2462
1.235.15magic . . . 2464
1.235.16magic_expansion(Mode) . . . 2466
1.235.17rule_expansion(Mode) . . . 2467

1.235.18 shell . . . 2468
1.235.19 shell(Interpreters) . . . 2470
1.235.20 shell_expansion(Mode) . . . 2471
1.236 wkt_wkb . . . 2473
1.236.1 wkt_wkb . . . 2473
1.236.2 wkt_wkb_protocol . . . 2474
1.237 wrapper . . . 2478
1.237.1 wrapper . . . 2478
1.238 xml_parser . . . 2490
1.238.1 xml . . . 2491
1.239 yaml . . . 2498
1.239.1 yaml . . . 2499
1.239.2 yaml_protocol . . . 2500
1.240 z_score_anomaly_detector . . . 2504
1.240.1 z_score_anomaly_detector . . . 2504
1.241 zippers . . . 2505
1.241.1 zipperp . . . 2506
1.241.2 zlist . . . 2516

2 Directories 2519
2.1 contributions/flags/ . . . 2521
2.2 contributions/iso8601/ . . . 2521
2.3 contributions/pddl_parser/ . . . 2521
2.4 contributions/verdi_neruda/ . . . 2521
2.5 contributions/xml_parser/ . . . 2521
2.6 core/ . . . 2521
2.7 library/ . . . 2521
2.8 library/adaptive_boosting_classifier/ . . . 2521
2.9 library/agglomerative_clusterer/ . . . 2521
2.10 library/amqp/ . . . 2521
2.11 library/anomaly_detection_protocols/ . . . 2521
2.12 library/application/ . . . 2521
2.13 library/apriori_pattern_miner/ . . . 2521
2.14 library/arbitrary/ . . . 2521
2.15 library/arrangements/ . . . 2521
2.16 library/assignvars/ . . . 2521
2.17 library/avro/ . . . 2521
2.18 library/base32/ . . . 2521
2.19 library/base58/ . . . 2521
2.20 library/base64/ . . . 2521
2.21 library/base85/ . . . 2521
2.22 library/bayesian_ridge_regression/ . . . 2521
2.23 library/borda_ranker/ . . . 2521
2.24 library/bradley_terry_ranker/ . . . 2521
2.25 library/c45_classifier/ . . . 2521
2.26 library/cartesian_products/ . . . 2521
2.27 library/cbor/ . . . 2521
2.28 library/ccsds_frames/ . . . 2521
2.29 library/ccsds_link_profiles/ . . . 2521
2.30 library/ccsds_packet_services/ . . . 2521
2.31 library/ccsds_packetization/ . . . 2521
2.32 library/ccsds_packets/ . . . 2521
2.33 library/ccsds_tc_services/ . . . 2521
2.34 library/ccsds_time_codes/ . . . 2521

2.35 library/ccsds_time_fields/ . . . 2521
2.36 library/character_sets/ . . . 2521
2.37 library/classification_protocols/ . . . 2521
2.38 library/clo_span_pattern_miner/ . . . 2521
2.39 library/clustering_protocols/ . . . 2521
2.40 library/colley_ranker/ . . . 2521
2.41 library/combinations/ . . . 2521
2.42 library/command_line_options/ . . . 2521
2.43 library/copeland_ranker/ . . . 2521
2.44 library/coroutining/ . . . 2521
2.45 library/crs_projections/ . . . 2521
2.46 library/csv/ . . . 2521
2.47 library/cuid2/ . . . 2521
2.48 library/cusum_anomaly_detector/ . . . 2521
2.49 library/datalog/ . . . 2521
2.50 library/dates/ . . . 2521
2.51 library/dates_tz/ . . . 2521
2.52 library/dbscan_clusterer/ . . . 2521
2.53 library/dependents/ . . . 2521
2.54 library/deques/ . . . 2521
2.55 library/derangements/ . . . 2521
2.56 library/dictionaries/ . . . 2521
2.57 library/dif/ . . . 2521
2.58 library/dimension_reduction_protocols/ . . . 2521
2.59 library/eclat_pattern_miner/ . . . 2521
2.60 library/edcg/ . . . 2521
2.61 library/elastic_net_regression/ . . . 2521
2.62 library/elo_ranker/ . . . 2521
2.63 library/events/ . . . 2521
2.64 library/ewma_anomaly_detector/ . . . 2521
2.65 library/expand_library_alias_paths/ . . . 2521
2.66 library/expecteds/ . . . 2521
2.67 library/format/ . . . 2521
2.68 library/fp_growth_pattern_miner/ . . . 2521
2.69 library/frequent_pattern_mining_protocols/ . . . 2521
2.70 library/gaussian_mixture_clusterer/ . . . 2521
2.71 library/gaussian_process_regression/ . . . 2521
2.72 library/genint/ . . . 2521
2.73 library/gensym/ . . . 2521
2.74 library/geohash/ . . . 2521
2.75 library/geojson/ . . . 2521
2.76 library/geospatial/ . . . 2521
2.77 library/git/ . . . 2521
2.78 library/glicko2_periodic_ranker/ . . . 2521
2.79 library/glicko2_ranker/ . . . 2521
2.80 library/gradient_boosting_classifier/ . . . 2521
2.81 library/gradient_boosting_regression/ . . . 2521
2.82 library/grammars/ . . . 2521
2.83 library/graphs/ . . . 2521
2.84 library/gsp_pattern_miner/ . . . 2521
2.85 library/hashes/ . . . 2521
2.86 library/hdbscan_clusterer/ . . . 2521
2.87 library/heaps/ . . . 2521
2.88 library/hierarchical_clustering/ . . . 2521

2.89 library/hierarchies/ . . . 2521
2.90 library/hmac/ . . . 2521
2.91 library/hodge_rank/ . . . 2521
2.92 library/hook_flows/ . . . 2521
2.93 library/hook_objects/ . . . 2521
2.94 library/html/ . . . 2521
2.95 library/ica_projection/ . . . 2521
2.96 library/ids/ . . . 2521
2.97 library/ieee_754/ . . . 2521
2.98 library/intervals/ . . . 2521
2.99 library/iqr_anomaly_detector/ . . . 2521
2.100 library/iso_13616/ . . . 2521
2.101 library/iso_3166/ . . . 2521
2.102 library/iso_4217/ . . . 2521
2.103 library/iso_639/ . . . 2521
2.104 library/iso_9362/ . . . 2521
2.105 library/isolation_forest_anomaly_detector/ . . . 2521
2.106 library/java/ . . . 2521
2.107 library/json/ . . . 2521
2.108 library/json_ld/ . . . 2521
2.109 library/json_lines/ . . . 2521
2.110 library/json_pointer/ . . . 2521
2.111 library/json_rpc/ . . . 2521
2.112 library/json_schema/ . . . 2521
2.113 library/kcenters_clusterer/ . . . 2521
2.114 library/kemeny_young_ranker/ . . . 2521
2.115 library/kernel_pca_projection/ . . . 2521
2.116 library/kernel_svm_classifier/ . . . 2521
2.117 library/kmeans_clusterer/ . . . 2521
2.118 library/kmedians_clusterer/ . . . 2521
2.119 library/kmedoids_clusterer/ . . . 2521
2.120 library/kmodes_clusterer/ . . . 2521
2.121 library/knn_classifier/ . . . 2521
2.122 library/knn_distance_anomaly_detector/ . . . 2521
2.123 library/knn_regression/ . . . 2521
2.124 library/kprototypes_clusterer/ . . . 2521
2.125 library/ksuid/ . . . 2521
2.126 library/lasso_regression/ . . . 2521
2.127 library/lda_classifier/ . . . 2521
2.128 library/lda_projection/ . . . 2521
2.129 library/linda/ . . . 2521
2.130 library/linear_algebra/ . . . 2521
2.131 library/linear_regression/ . . . 2521
2.132 library/linear_svm_classifier/ . . . 2521
2.133 library/listing/ . . . 2521
2.134 library/lof_anomaly_detector/ . . . 2521
2.135 library/logging/ . . . 2521
2.136 library/logistic_regression_classifier/ . . . 2521
2.137 library/loops/ . . . 2521
2.138 library/massey_ranker/ . . . 2521
2.139 library/mcp_server/ . . . 2521
2.140 library/memcached/ . . . 2521
2.141 library/message_pack/ . . . 2521
2.142 library/meta/ . . . 2521

2.143 library/meta_compiler/ . . . 2521
2.144 library/mime_types/ . . . 2521
2.145 library/modified_z_score_anomaly_detector/ . . . 2521
2.146 library/multisets/ . . . 2521
2.147 library/mutations/ . . . 2521
2.148 library/naive_bayes_classifier/ . . . 2521
2.149 library/nanoid/ . . . 2521
2.150 library/nearest_centroid_classifier/ . . . 2521
2.151 library/nested_dictionaries/ . . . 2521
2.152 library/nmea/ . . . 2521
2.153 library/nmf_projection/ . . . 2521
2.154 library/optics_clusterer/ . . . 2521
2.155 library/optionals/ . . . 2521
2.156 library/options/ . . . 2521
2.157 library/os/ . . . 2521
2.158 library/partitions/ . . . 2521
2.159 library/pattern_mining_protocols/ . . . 2521
2.160 library/pca_projection/ . . . 2521
2.161 library/permutations/ . . . 2521
2.162 library/plackett_luce_last_ranker/ . . . 2521
2.163 library/plackett_luce_ranker/ . . . 2521
2.164 library/pls_projection/ . . . 2521
2.165 library/prefix_span_pattern_miner/ . . . 2521
2.166 library/probabilistic_pca_projection/ . . . 2521
2.167 library/process/ . . . 2521
2.168 library/protobuf/ . . . 2521
2.169 library/qda_classifier/ . . . 2521
2.170 library/queues/ . . . 2521
2.171 library/random/ . . . 2521
2.172 library/random_forest_classifier/ . . . 2521
2.173 library/random_forest_regression/ . . . 2521
2.174 library/random_projection/ . . . 2521
2.175 library/rank_centrality/ . . . 2521
2.176 library/ranked_pairs/ . . . 2521
2.177 library/ranking_protocols/ . . . 2521
2.178 library/reader/ . . . 2521
2.179 library/recorded_database/ . . . 2521
2.180 library/redis/ . . . 2521
2.181 library/regression_protocols/ . . . 2521
2.182 library/regression_tree/ . . . 2521
2.183 library/regularized_bradley_terry_ranker/ . . . 2521
2.184 library/ridge_regression/ . . . 2521
2.185 library/schulze_ranker/ . . . 2521
2.186 library/sequential_pattern_mining_protocols/ . . . 2521
2.187 library/sets/ . . . 2521
2.188 library/sgd_classifier/ . . . 2521
2.189 library/simulated_annealing/ . . . 2521
2.190 library/snowflakeid/ . . . 2521
2.191 library/sockets/ . . . 2521
2.192 library/spade_pattern_miner/ . . . 2521
2.193 library/statistics/ . . . 2521
2.194 library/stemming/ . . . 2521
2.195 library/stomp/ . . . 2521
2.196 library/string_distance/ . . . 2521

2.197 library/strings/ . . . 2521
2.198 library/subsequences/ . . . 2521
2.199 library/term_io/ . . . 2521
2.200 library/thurstone_mosteller_ranker/ . . . 2521
2.201 library/time_scales/ . . . 2521
2.202 library/timeout/ . . . 2521
2.203 library/tle_orbits/ . . . 2521
2.204 library/toml/ . . . 2521
2.205 library/toon/ . . . 2521
2.206 library/truncated_svd_projection/ . . . 2521
2.207 library/tsv/ . . . 2521
2.208 library/types/ . . . 2521
2.209 library/tzif/ . . . 2521
2.210 library/ulid/ . . . 2521
2.211 library/union_find/ . . . 2521
2.212 library/url/ . . . 2521
2.213 library/uuid/ . . . 2521
2.214 library/validations/ . . . 2521
2.215 library/wkt_wkb/ . . . 2521
2.216 library/yaml/ . . . 2521
2.217 library/z_score_anomaly_detector/ . . . 2521
2.218 library/zippers/ . . . 2521
2.219 ports/fcube/ . . . 2521
2.220 ports/metagol/ . . . 2521
2.221 ports/toychr/ . . . 2521
2.222 tools/assertions/ . . . 2521
2.223 tools/code_metrics/ . . . 2521
2.224 tools/dead_code_scanner/ . . . 2521
2.225 tools/debug_messages/ . . . 2521
2.226 tools/debugger/ . . . 2521
2.227 tools/diagrams/ . . . 2521
2.228 tools/doclet/ . . . 2521
2.229 tools/help/ . . . 2521
2.230 tools/issue_creator/ . . . 2521
2.231 tools/lgtdoc/ . . . 2521
2.232 tools/lgtunit/ . . . 2521
2.233 tools/linter_reporter/ . . . 2521
2.234 tools/mutation_testing/ . . . 2521
2.235 tools/mutation_testing/mutators/ . . . 2521
2.236 tools/packs/ . . . 2521
2.237 tools/ports_profiler/ . . . 2521
2.238 tools/sarif/ . . . 2521
2.239 tools/sbom/ . . . 2521
2.240 tools/tool_diagnostics/ . . . 2521
2.241 tools/tutor/ . . . 2521
2.242 tools/wrapper/ . . . 2521

3 Entities 2523
3.1 Categories . . . 2523
3.2 Objects . . . 2523
3.3 Protocols . . . 2523

4 Predicates 2525
4.1 (/)/2 . . . 2525

4.2 (//)/2 . . . 2525
4.3 (<)/2 . . . 2525
4.4 (<=)/2 . . . 2525
4.5 (=:=)/2 . . . 2525
4.6 (=<)/2 . . . 2526
4.7 (=>)/2 . . . 2526
4.8 (=\=)/2 . . . 2526
4.9 =~= / 2 . . . 2526
4.10 (>)/2 . . . 2526
4.11 (>=)/2 . . . 2526
4.12 abbreviation/2 . . . 2526
4.13 abbreviation/3 . . . 2526
4.14 abbreviation/4 . . . 2527
4.15 abort_transaction/3 . . . 2527
4.16 absolute_file_name/2 . . . 2527
4.17 ack/3 . . . 2527
4.18 acquire/1 . . . 2527
4.19 activate_debug_handler/1 . . . 2527
4.20 activate_monitor/0 . . . 2527
4.21 active_debug_handler/1 . . . 2527
4.22 add/1 . . . 2528
4.23 add/2 . . . 2528
4.24 add/3 . . . 2528
4.25 add/5 . . . 2528
4.26 add32/3 . . . 2528
4.27 add32/4 . . . 2528
4.28 add32/5 . . . 2528
4.29 add64/3 . . . 2528
4.30 addDependent/1 . . . 2529
4.31 add_duration/3 . . . 2529
4.32 add_edge/4 . . . 2529
4.33 add_edge/5 . . . 2529
4.34 add_edges/3 . . . 2529
4.35 add_matrices/3 . . . 2529
4.36 add_rule/3 . . . 2529
4.37 add_scaled_outer_product/4 . . . 2529
4.38 add_scaled_vector/4 . . . 2530
4.39 add_type/2 . . . 2530
4.40 add_type/3 . . . 2530
4.41 add_vectors/3 . . . 2530
4.42 add_vertex/3 . . . 2530
4.43 add_vertices/3 . . . 2530
4.44 adjacent/3 . . . 2530
4.45 adjacent_bits/4 . . . 2530
4.46 adjacent_int/4 . . . 2531
4.47 after/2 . . . 2531
4.48 after/3 . . . 2531
4.49 alias/1 . . . 2531
4.50 all/0 . . . 2531
4.51 all/1 . . . 2531
4.52 all_files/0 . . . 2531
4.53 all_files/1 . . . 2532
4.54 all_libraries/0 . . . 2532
4.55 all_libraries/1 . . . 2532

4.56 all_pairs_min_paths/2 . . . 2532
4.57 all_pairs_min_predecessors/2 . . . 2532
4.58 all_score/1 . . . 2532
4.59 along_track_distance/4 . . . 2532
4.60 alternating_subsequence/2 . . . 2533
4.61 alternating_subsequences/2 . . . 2533
4.62 ancestor/1 . . . 2533
4.63 ancestors/1 . . . 2533
4.64 and64/3 . . . 2533
4.65 anomaly_detector_options/2 . . . 2533
4.66 apid/2 . . . 2533
4.67 apis/0 . . . 2533
4.68 apis/1 . . . 2534
4.69 append/2 . . . 2534
4.70 append/3 . . . 2534
4.71 append/4 . . . 2534
4.72 apply/2 . . . 2534
4.73 apply/4 . . . 2534
4.74 approximately_equal/2 . . . 2534
4.75 approximately_equal/3 . . . 2535
4.76 arbitrary/1 . . . 2535
4.77 arbitrary/2 . . . 2535
4.78 archive/1 . . . 2535
4.79 arithmetic_mean/2 . . . 2535
4.80 arrangement/3 . . . 2535
4.81 arrangement/4 . . . 2535
4.82 arrangement_index/4 . . . 2535
4.83 arrangement_index/5 . . . 2536
4.84 arrangements/3 . . . 2536
4.85 arrangements/4 . . . 2536
4.86 array_list/2 . . . 2536
4.87 array_to_list/2 . . . 2536
4.88 array_to_terms/2 . . . 2536
4.89 array_to_terms/3 . . . 2536
4.90 articulation_points/2 . . . 2536
4.91 as_curly_bracketed/2 . . . 2537
4.92 as_deque/2 . . . 2537
4.93 as_dictionary/2 . . . 2537
4.94 as_difflist/2 . . . 2537
4.95 as_heap/2 . . . 2537
4.96 as_list/2 . . . 2537
4.97 as_nested_dictionary/2 . . . 2537
4.98 as_set/2 . . . 2538
4.99 ask_question/5 . . . 2538
4.100 assert_fact/1 . . . 2538
4.101 assertion/1 . . . 2538
4.102 assertion/2 . . . 2538
4.103 assignable/1 . . . 2538
4.104 assignable/2 . . . 2538
4.105 atom_string/2 . . . 2538
4.106 atomics_to_string/2 . . . 2539
4.107 atomics_to_string/3 . . . 2539
4.108 attribute_values/2 . . . 2539
4.109 available/0 . . . 2539

4.110 available/1 . . . 2539
4.111 available/2 . . . 2539
4.112 average/2 . . . 2539
4.113 average_deviation/3 . . . 2540
4.114 backward_substitution/3 . . . 2540
4.115 basic_ack/3 . . . 2540
4.116 basic_cancel/3 . . . 2540
4.117 basic_consume/3 . . . 2540
4.118 basic_get/3 . . . 2540
4.119 basic_nack/3 . . . 2540
4.120 basic_publish/4 . . . 2540
4.121 basic_qos/2 . . . 2541
4.122 basic_recover/2 . . . 2541
4.123 basic_reject/3 . . . 2541
4.124 basis_vector/3 . . . 2541
4.125 bbox_contains/2 . . . 2541
4.126 bbox_contains_polygon/2 . . . 2541
4.127 bbox_expand/3 . . . 2541
4.128 bbox_from_coordinates/2 . . . 2541
4.129 bbox_intersects/2 . . . 2542
4.130 bbox_intersects_polygon/2 . . . 2542
4.131 bbox_intersects_polyline/2 . . . 2542
4.132 bbox_overlaps/2 . . . 2542
4.133 bbox_union/3 . . . 2542
4.134 before/2 . . . 2542
4.135 before/3 . . . 2542
4.136 begin/0 . . . 2542
4.137 begin_transaction/3 . . . 2543
4.138 bell/2 . . . 2543
4.139 bench_goal/1 . . . 2543
4.140 benchmark/2 . . . 2543
4.141 benchmark/3 . . . 2543
4.142 benchmark/4 . . . 2543
4.143 benchmark_reified/3 . . . 2543
4.144 bernoulli/2 . . . 2543
4.145 beta/3 . . . 2544
4.146 between/3 . . . 2544
4.147 between/4 . . . 2544
4.148 bic/5 . . . 2544
4.149 big_endian_word32/2 . . . 2544
4.150 binomial/3 . . . 2544
4.151 bit//1 . . . 2544
4.152 bits//1 . . . 2544
4.153 blank//0 . . . 2545
4.154 blanks//0 . . . 2545
4.155 block_size/1 . . . 2545
4.156 body_pred/1 . . . 2545
4.157 bounding_box/2 . . . 2545
4.158 bounding_box/3 . . . 2545
4.159 bounding_box_bits/3 . . . 2545
4.160 bounding_box_int/3 . . . 2545
4.161 branch/2 . . . 2546
4.162 breadth_first_order/3 . . . 2546
4.163 bridges/2 . . . 2546

4.164 built_date/1 . . . 2546
4.165 built_in_directive/4 . . . 2546
4.166 built_in_flag/2 . . . 2546
4.167 built_in_method/4 . . . 2546
4.168 built_in_non_terminal/4 . . . 2546
4.169 built_in_predicate/4 . . . 2547
4.170 bypass_flag/2 . . . 2547
4.171 byte_count/1 . . . 2547
4.172 bytes_hex/2 . . . 2547
4.173 bytes_to_codes/2 . . . 2547
4.174 cache/1 . . . 2547
4.175 cache_source/1 . . . 2547
4.176 cached_tzif/1 . . . 2547
4.177 calendar_month/3 . . . 2548
4.178 call_with_timeout/2 . . . 2548
4.179 call_with_timeout/3 . . . 2548
4.180 canonical_bic/2 . . . 2548
4.181 canonical_iban/2 . . . 2548
4.182 capabilities/1 . . . 2548
4.183 cartesian_product/2 . . . 2548
4.184 cartesian_product/3 . . . 2548
4.185 cartesian_tuple/2 . . . 2549
4.186 cartesian_tuple/3 . . . 2549
4.187 cartesian_tuple_index/3 . . . 2549
4.188 cartesian_tuple_index/4 . . . 2549
4.189 cas/6 . . . 2549
4.190 cat/2 . . . 2549
4.191 catalan/2 . . . 2549
4.192 cell_dimensions/3 . . . 2549
4.193 center_gram_matrix/4 . . . 2550
4.194 center_gram_vector/4 . . . 2550
4.195 central_moment/3 . . . 2550
4.196 change_directory/1 . . . 2550
4.197 changed/0 . . . 2550
4.198 changed/1 . . . 2550
4.199 channel_close/1 . . . 2550
4.200 channel_close/3 . . . 2550
4.201 channel_open/3 . . . 2551
4.202 chebyshev_distance/3 . . . 2551
4.203 chebyshev_norm/2 . . . 2551
4.204 check/0 . . . 2551
4.205 check/1 . . . 2551
4.206 check/2 . . . 2551
4.207 check/3 . . . 2551
4.208 check_anomaly_detector/1 . . . 2551
4.209 check_classifier/1 . . . 2552
4.210 check_clusterer/1 . . . 2552
4.211 check_conversion/3 . . . 2552
4.212 check_convert/4 . . . 2552
4.213 check_dimension_reducer/1 . . . 2552
4.214 check_instant/1 . . . 2552
4.215 check_offset/3 . . . 2552
4.216 check_option/1 . . . 2552
4.217 check_options/1 . . . 2553

4.218 check_pattern_miner/1 . . . 2553
4.219 check_ranker/1 . . . 2553
4.220 check_regressor/1 . . . 2553
4.221 checksum/2 . . . 2553
4.222 chi_squared/2 . . . 2553
4.223 children/2 . . . 2553
4.224 cholesky_decomposition/2 . . . 2553
4.225 chr_is/2 . . . 2554
4.226 chr_no_spy/1 . . . 2554
4.227 chr_nospy/0 . . . 2554
4.228 chr_notrace/0 . . . 2554
4.229 chr_option/2 . . . 2554
4.230 chr_spy/1 . . . 2554
4.231 chr_trace/0 . . . 2554
4.232 circular_uniform_cartesian/3 . . . 2554
4.233 circular_uniform_polar/3 . . . 2555
4.234 class/1 . . . 2555
4.235 class_values/1 . . . 2555
4.236 classes/1 . . . 2555
4.237 classifier_options/2 . . . 2555
4.238 classify/2 . . . 2555
4.239 clause/5 . . . 2555
4.240 clause_location/6 . . . 2556
4.241 clean/0 . . . 2556
4.242 clean/1 . . . 2556
4.243 clean/2 . . . 2556
4.244 clear/0 . . . 2556
4.245 clear_cache/0 . . . 2556
4.246 clear_dut1_override/0 . . . 2556
4.247 clear_leap_seconds_override/0 . . . 2557
4.248 client_open/4 . . . 2557
4.249 client_open/5 . . . 2557
4.250 clockwise_polygon/2 . . . 2557
4.251 clone/1 . . . 2557
4.252 clone/3 . . . 2557
4.253 clone/4 . . . 2557
4.254 close/1 . . . 2557
4.255 close/2 . . . 2558
4.256 close/3 . . . 2558
4.257 close_client/1 . . . 2558
4.258 close_polygon/2 . . . 2558
4.259 cluster/3 . . . 2558
4.260 cluster_probabilities/3 . . . 2558
4.261 clusterer_options/2 . . . 2558
4.262 coarse/2 . . . 2558
4.263 codes_to_bytes/2 . . . 2559
4.264 coefficient_of_variation/2 . . . 2559
4.265 combination/3 . . . 2559
4.266 combination/4 . . . 2559
4.267 combination_index/4 . . . 2559
4.268 combination_index/5 . . . 2559
4.269 combinations/3 . . . 2559
4.270 combinations/4 . . . 2559
4.271 command_line_arguments/1 . . . 2560

4.272 commit/0 . . . 2560
4.273 commit_author/2 . . . 2560
4.274 commit_date/2 . . . 2560
4.275 commit_hash/2 . . . 2560
4.276 commit_hash_abbreviated/2 . . . 2560
4.277 commit_log/3 . . . 2560
4.278 commit_message/2 . . . 2560
4.279 commit_transaction/3 . . . 2561
4.280 common_prefix/3 . . . 2561
4.281 common_subsequence/3 . . . 2561
4.282 common_subsequences/3 . . . 2561
4.283 compact/3 . . . 2561
4.284 compare_date_time/3 . . . 2561
4.285 compile_aux_clauses/1 . . . 2561
4.286 compile_predicate_heads/4 . . . 2561
4.287 compile_predicate_indicators/3 . . . 2562
4.288 complement/2 . . . 2562
4.289 completion/2 . . . 2562
4.290 completions/2 . . . 2562
4.291 compose/3 . . . 2562
4.292 compress/2 . . . 2562
4.293 confirm_select/1 . . . 2562
4.294 connect/1 . . . 2562
4.295 connect/2 . . . 2563
4.296 connect/3 . . . 2563
4.297 connect/4 . . . 2563
4.298 connected_components/2 . . . 2563
4.299 connection_alive/1 . . . 2563
4.300 consensus_ranking/2 . . . 2563
4.301 consensus_score/2 . . . 2563
4.302 consistent/1 . . . 2564
4.303 console/1 . . . 2564
4.304 constraints/2 . . . 2564
4.305 contains/2 . . . 2564
4.306 contradiction/2 . . . 2564
4.307 contradiction_explanations/2 . . . 2564
4.308 control//0 . . . 2564
4.309 control_command_flag/2 . . . 2564
4.310 control_construct/4 . . . 2565
4.311 controls//0 . . . 2565
4.312 converse/2 . . . 2565
4.313 convert/4 . . . 2565
4.314 convert_zones/4 . . . 2565
4.315 convert_zones_with_resolution/5 . . . 2565
4.316 cooling_schedule/3 . . . 2565
4.317 coordinates_bounding_box/2 . . . 2565
4.318 copy_file/2 . . . 2566
4.319 correlation/3 . . . 2566
4.320 cosine_similarity/3 . . . 2566
4.321 count_arrangements/3 . . . 2566
4.322 count_cartesian_product/2 . . . 2566
4.323 count_combinations/3 . . . 2566
4.324 count_derangements/2 . . . 2566
4.325 count_distinct_arrangements/3 . . . 2566

4.326 count_distinct_cartesian_product/2 . . . 2567
4.327 count_distinct_combinations/3 . . . 2567
4.328 count_distinct_derangements/2 . . . 2567
4.329 count_distinct_multisets/3 . . . 2567
4.330 count_distinct_partitions/2 . . . 2567
4.331 count_distinct_partitions/3 . . . 2567
4.332 count_distinct_permutations/2 . . . 2567
4.333 count_distinct_subsequences/3 . . . 2567
4.334 count_multisets/3 . . . 2568
4.335 count_partial_derangements/3 . . . 2568
4.336 count_partitions/2 . . . 2568
4.337 count_partitions/3 . . . 2568
4.338 count_permutations/2 . . . 2568
4.339 count_subsequences/2 . . . 2568
4.340 counter/2 . . . 2568
4.341 counterclockwise_polygon/2 . . . 2568
4.342 country/4 . . . 2569
4.343 country_spec/3 . . . 2569
4.344 covariance/3 . . . 2569
4.345 covariance_matrix/2 . . . 2569
4.346 cover/1 . . . 2569
4.347 coverage_clause_mutator/0 . . . 2569
4.348 covering/3 . . . 2569
4.349 covering/4 . . . 2569
4.350 cpu_time/1 . . . 2570
4.351 create/3 . . . 2570
4.352 creators/1 . . . 2570
4.353 cross_track_distance/4 . . . 2570
4.354 crs_dimensions/2 . . . 2570
4.355 crs_epsg/2 . . . 2570
4.356 crs_kind/2 . . . 2570
4.357 crs_name/2 . . . 2570
4.358 crs_units/2 . . . 2571
4.359 currency/5 . . . 2571
4.360 current/2 . . . 2571
4.361 current_host/1 . . . 2571
4.362 cut/3 . . . 2571
4.363 cycle/2 . . . 2571
4.364 damerau_levenshtein/3 . . . 2571
4.365 data/0 . . . 2571
4.366 data/1 . . . 2572
4.367 data/2 . . . 2572
4.368 data_field/2 . . . 2572
4.369 data_length/2 . . . 2572
4.370 date/4 . . . 2572
4.371 date/5 . . . 2572
4.372 date/6 . . . 2572
4.373 date/7 . . . 2572
4.374 date_string/3 . . . 2573
4.375 date_time/7 . . . 2573
4.376 date_time_string/3 . . . 2573
4.377 date_time_to_unix/2 . . . 2573
4.378 day/2 . . . 2573
4.379 day_of_year/2 . . . 2573

4.380 day_of_year_date/3 . . . 2573
4.381 daylight_saving_time/2 . . . 2573
4.382 daylight_saving_time/3 . . . 2574
4.383 daylight_saving_time/4 . . . 2574
4.384 days/2 . . . 2574
4.385 days_in_month/3 . . . 2574
4.386 deactivate_debug_handler/0 . . . 2574
4.387 debug/0 . . . 2574
4.388 debug_handler/1 . . . 2574
4.389 debug_handler/3 . . . 2574
4.390 debugging/0 . . . 2575
4.391 debugging/1 . . . 2575
4.392 decide/1 . . . 2575
4.393 decide/2 . . . 2575
4.394 decode/2 . . . 2575
4.395 decode_bits/3 . . . 2575
4.396 decode_exception/2 . . . 2575
4.397 decode_exception/3 . . . 2575
4.398 decode_frame/2 . . . 2576
4.399 decode_int/3 . . . 2576
4.400 decompile_predicate_heads/4 . . . 2576
4.401 decompile_predicate_indicators/4 . . . 2576
4.402 decompose_file_name/3 . . . 2576
4.403 decompose_file_name/4 . . . 2576
4.404 decr/3 . . . 2576
4.405 decr/4 . . . 2576
4.406 decrby/4 . . . 2577
4.407 decrement_counter/1 . . . 2577
4.408 default/1 . . . 2577
4.409 default_option/1 . . . 2577
4.410 default_options/1 . . . 2577
4.411 define_log_file/2 . . . 2577
4.412 defined/4 . . . 2577
4.413 defined_flag/6 . . . 2577
4.414 degree/3 . . . 2578
4.415 del/3 . . . 2578
4.416 del_monitors/0 . . . 2578
4.417 del_monitors/4 . . . 2578
4.418 del_spy_points/4 . . . 2578
4.419 delete/0 . . . 2578
4.420 delete/1 . . . 2578
4.421 delete/2 . . . 2578
4.422 delete/3 . . . 2579
4.423 delete/4 . . . 2579
4.424 delete_all_after/2 . . . 2579
4.425 delete_all_after_and_unzip/2 . . . 2579
4.426 delete_all_before/2 . . . 2579
4.427 delete_all_before_and_unzip/2 . . . 2579
4.428 delete_and_next/2 . . . 2579
4.429 delete_and_previous/2 . . . 2579
4.430 delete_and_unzip/2 . . . 2580
4.431 delete_directory/1 . . . 2580
4.432 delete_directory_and_contents/1 . . . 2580
4.433 delete_directory_contents/1 . . . 2580

4.434 delete_edge/4 . . . 2580
4.435 delete_edge/5 . . . 2580
4.436 delete_edges/3 . . . 2580
4.437 delete_file/1 . . . 2580
4.438 delete_in/4 . . . 2581
4.439 delete_matches/3 . . . 2581
4.440 delete_max/4 . . . 2581
4.441 delete_min/4 . . . 2581
4.442 delete_vertex/3 . . . 2581
4.443 delete_vertices/3 . . . 2581
4.444 dependents/1 . . . 2581
4.445 dependents/2 . . . 2581
4.446 dependents/3 . . . 2582
4.447 depth/2 . . . 2582
4.448 depth_first_order/3 . . . 2582
4.449 derangement/2 . . . 2582
4.450 derangement/3 . . . 2582
4.451 derangement_index/3 . . . 2582
4.452 derangement_index/4 . . . 2582
4.453 derangements/2 . . . 2582
4.454 derangements/3 . . . 2583
4.455 descendant/1 . . . 2583
4.456 descendant_class/1 . . . 2583
4.457 descendant_classes/1 . . . 2583
4.458 descendant_instance/1 . . . 2583
4.459 descendant_instances/1 . . . 2583
4.460 descendants/1 . . . 2583
4.461 describe/1 . . . 2583
4.462 describe/2 . . . 2584
4.463 description/1 . . . 2584
4.464 destination_point/4 . . . 2584
4.465 destroy/0 . . . 2584
4.466 determinant/2 . . . 2584
4.467 deterministic/1 . . . 2584
4.468 deterministic/2 . . . 2584
4.469 diagnostic/2 . . . 2584
4.470 diagnostic/3 . . . 2585
4.471 diagnostic_rule/5 . . . 2585
4.472 diagnostic_rules/1 . . . 2585
4.473 diagnostic_target/1 . . . 2585
4.474 diagnostics/2 . . . 2585
4.475 diagnostics/3 . . . 2586
4.476 diagnostics_preflight/2 . . . 2586
4.477 diagnostics_preflight/3 . . . 2586
4.478 diagnostics_summary/2 . . . 2586
4.479 diagnostics_summary/3 . . . 2586
4.480 diagnostics_tool/5 . . . 2586
4.481 diagonal_matrix/2 . . . 2586
4.482 diagonal_matrix/3 . . . 2586
4.483 diagonal_matrix/4 . . . 2587
4.484 diagram_description/1 . . . 2587
4.485 diagram_name_suffix/1 . . . 2587
4.486 dif/1 . . . 2587
4.487 dif/2 . . . 2587

4.488 difference_norm/3 . . . 2587
4.489 digest/2 . . . 2587
4.490 digest/4 . . . 2587
4.491 digest/5 . . . 2588
4.492 digest_size/1 . . . 2588
4.493 digit//1 . . . 2588
4.494 digits//1 . . . 2588
4.495 dimension_reducer_options/2 . . . 2588
4.496 directories/1 . . . 2588
4.497 directories/2 . . . 2588
4.498 directories/3 . . . 2589
4.499 directory/1 . . . 2589
4.500 directory/2 . . . 2589
4.501 directory/3 . . . 2589
4.502 directory_exists/1 . . . 2590
4.503 directory_files/2 . . . 2590
4.504 directory_files/3 . . . 2590
4.505 directory_mutants/2 . . . 2590
4.506 directory_mutants/3 . . . 2590
4.507 directory_score/2 . . . 2590
4.508 dirichlet/2 . . . 2590
4.509 disable/0 . . . 2590
4.510 disable/1 . . . 2591
4.511 disable/2 . . . 2591
4.512 disable_logging/1 . . . 2591
4.513 disconnect/1 . . . 2591
4.514 disconnect/2 . . . 2591
4.515 disjoint/2 . . . 2591
4.516 disjoint_sets/2 . . . 2591
4.517 dispatch_map_id/2 . . . 2591
4.518 dispatch_service_units/2 . . . 2592
4.519 dispatch_service_units_by_map/2 . . . 2592
4.520 dispatch_service_units_by_map/3 . . . 2592
4.521 distance/4 . . . 2592
4.522 distance/5 . . . 2592
4.523 distinct_arrangement/3 . . . 2592
4.524 distinct_arrangement/4 . . . 2592
4.525 distinct_arrangement_index/4 . . . 2592
4.526 distinct_arrangements/3 . . . 2593
4.527 distinct_arrangements/4 . . . 2593
4.528 distinct_cartesian_product/2 . . . 2593
4.529 distinct_cartesian_product/3 . . . 2593
4.530 distinct_cartesian_tuple/2 . . . 2593
4.531 distinct_cartesian_tuple/3 . . . 2593
4.532 distinct_cartesian_tuple_index/3 . . . 2593
4.533 distinct_combination/3 . . . 2593
4.534 distinct_combination/4 . . . 2594
4.535 distinct_combination_index/4 . . . 2594
4.536 distinct_combinations/3 . . . 2594
4.537 distinct_combinations/4 . . . 2594
4.538 distinct_derangement/2 . . . 2594
4.539 distinct_derangement/3 . . . 2594
4.540 distinct_derangement_index/3 . . . 2594
4.541 distinct_derangements/2 . . . 2594

4.542 distinct_derangements/3 . . . 2595
4.543 distinct_multiset/3 . . . 2595
4.544 distinct_multiset/4 . . . 2595
4.545 distinct_multiset_index/4 . . . 2595
4.546 distinct_multisets/3 . . . 2595
4.547 distinct_multisets/4 . . . 2595
4.548 distinct_partition/2 . . . 2595
4.549 distinct_partition/3 . . . 2595
4.550 distinct_partition/4 . . . 2596
4.551 distinct_partition_index/3 . . . 2596
4.552 distinct_partition_index/4 . . . 2596
4.553 distinct_partition_index/5 . . . 2596
4.554 distinct_partitions/2 . . . 2596
4.555 distinct_partitions/3 . . . 2596
4.556 distinct_partitions/4 . . . 2596
4.557 distinct_permutation/2 . . . 2596
4.558 distinct_permutation/3 . . . 2597
4.559 distinct_permutation_index/3 . . . 2597
4.560 distinct_permutations/2 . . . 2597
4.561 distinct_permutations/3 . . . 2597
4.562 distribution/1 . . . 2597
4.563 doc_goal/1 . . . 2597
4.564 document/1 . . . 2597
4.565 document/2 . . . 2597
4.566 dot//1 . . . 2598
4.567 dot_product/3 . . . 2598
4.568 double_metaphone/3 . . . 2598
4.569 double_metaphone_match/2 . . . 2598
4.570 dowhile/2 . . . 2598
4.571 drop/3 . . . 2598
4.572 duration_between/3 . . . 2598
4.573 duration_string/2 . . . 2598
4.574 during/2 . . . 2599
4.575 dut1_entries/1 . . . 2599
4.576 dut1_offset_at_utc_unix/3 . . . 2599
4.577 dut1_source/1 . . . 2599
4.578 easter_day/3 . . . 2599
4.579 edge/3 . . . 2599
4.580 edge/4 . . . 2599
4.581 edge/6 . . . 2599
4.582 edge_case/2 . . . 2600
4.583 edges/2 . . . 2600
4.584 edit_similarity/3 . . . 2600
4.585 edit_similarity/4 . . . 2600
4.586 either/3 . . . 2600
4.587 elicit_request/5 . . . 2600
4.588 empty/1 . . . 2600
4.589 enable/0 . . . 2601
4.590 enable/1 . . . 2601
4.591 enable/2 . . . 2601
4.592 enable_logging/1 . . . 2601
4.593 enabled/1 . . . 2601
4.594 enabled/2 . . . 2601
4.595 encode/2 . . . 2601

4.596 encode/3 . . . 2601
4.597 encode_bits/3 . . . 2602
4.598 encode_frame/2 . . . 2602
4.599 encode_int/3 . . . 2602
4.600 encoding_suffix/2 . . . 2602
4.601 ensure_directory/1 . . . 2602
4.602 ensure_file/1 . . . 2602
4.603 entailment_explanations/5 . . . 2602
4.604 entails/4 . . . 2602
4.605 entails/5 . . . 2603
4.606 entity/1 . . . 2603
4.607 entity/2 . . . 2603
4.608 entity_info_pair_score_hook/3 . . . 2603
4.609 entity_info_score_hook/2 . . . 2603
4.610 entity_mutants/2 . . . 2603
4.611 entity_mutants/3 . . . 2603
4.612 entity_predicates_weights_hook/2 . . . 2604
4.613 entity_prefix/2 . . . 2604
4.614 entity_score/2 . . . 2604
4.615 enumerate/2 . . . 2604
4.616 environment_variable/2 . . . 2604
4.617 epoch/1 . . . 2604
4.618 epoch/2 . . . 2604
4.619 epsg_crs/2 . . . 2604
4.620 epsilon/1 . . . 2605
4.621 equal/2 . . . 2605
4.622 equirectangular_inverse/4 . . . 2605
4.623 equirectangular_projection/4 . . . 2605
4.624 equivalent/2 . . . 2605
4.625 erase/1 . . . 2605
4.626 error/2 . . . 2605
4.627 error_code/2 . . . 2605
4.628 error_data/2 . . . 2606
4.629 error_message/2 . . . 2606
4.630 error_response/4 . . . 2606
4.631 error_response/5 . . . 2606
4.632 essentially_equal/3 . . . 2606
4.633 estimate_temperature/1 . . . 2606
4.634 estimate_temperature/2 . . . 2606
4.635 euclidean_distance/3 . . . 2606
4.636 euclidean_norm/2 . . . 2607
4.637 evaluate/3 . . . 2607
4.638 exactly_representable/1 . . . 2607
4.639 example/2 . . . 2607
4.640 example/3 . . . 2607
4.641 exchange_bind/4 . . . 2607
4.642 exchange_declare/3 . . . 2607
4.643 exchange_delete/3 . . . 2608
4.644 exchange_unbind/4 . . . 2608
4.645 exclude/3 . . . 2608
4.646 excluded/4 . . . 2608
4.647 execution_context/7 . . . 2608
4.648 exists/3 . . . 2608
4.649 expand/2 . . . 2608

4.650 expand_int/3 . . . 2608
4.651 expand_library_path/2 . . . 2609
4.652 expected/1 . . . 2609
4.653 expecteds/2 . . . 2609
4.654 expire/4 . . . 2609
4.655 explain/2 . . . 2609
4.656 explain//1 . . . 2609
4.657 exponential/2 . . . 2609
4.658 export/1 . . . 2609
4.659 export/2 . . . 2610
4.660 export_to_clauses/4 . . . 2610
4.661 export_to_file/4 . . . 2610
4.662 extension/1 . . . 2610
4.663 extension_type/2 . . . 2610
4.664 extension_type/3 . . . 2611
4.665 extensions/1 . . . 2611
4.666 external_reference/2 . . . 2611
4.667 extract_aos_packets/3 . . . 2611
4.668 extract_packets/3 . . . 2611
4.669 extract_packets/4 . . . 2611
4.670 extract_tc_segment/2 . . . 2611
4.671 extract_tm_packets/3 . . . 2611
4.672 factorial/2 . . . 2612
4.673 facts/1 . . . 2612
4.674 failed_test_reason//1 . . . 2612
4.675 falling_factorial/3 . . . 2612
4.676 false/1 . . . 2612
4.677 fcube/0 . . . 2612
4.678 fecf/2 . . . 2612
4.679 fibonacci/2 . . . 2612
4.680 fields/2 . . . 2613
4.681 fields/5 . . . 2613
4.682 file/1 . . . 2613
4.683 file/2 . . . 2613
4.684 file_exists/1 . . . 2613
4.685 file_footer/3 . . . 2613
4.686 file_header/3 . . . 2614
4.687 file_modification_time/2 . . . 2614
4.688 file_path_components/2 . . . 2614
4.689 file_permission/2 . . . 2614
4.690 file_score/2 . . . 2614
4.691 file_size/2 . . . 2614
4.692 file_to_bytes/2 . . . 2614
4.693 file_to_bytes/3 . . . 2614
4.694 file_to_chars/2 . . . 2615
4.695 file_to_chars/3 . . . 2615
4.696 file_to_codes/2 . . . 2615
4.697 file_to_codes/3 . . . 2615
4.698 file_to_terms/2 . . . 2615
4.699 file_to_terms/3 . . . 2615
4.700 file_type_extension/2 . . . 2615
4.701 files/1 . . . 2615
4.702 files/2 . . . 2616
4.703 files/3 . . . 2616

4.704 filter/2 . . . 2616
4.705 filter/3 . . . 2616
4.706 final_bearing/3 . . . 2616
4.707 find/4 . . . 2616
4.708 find/5 . . . 2616
4.709 findall_in_noblock/3 . . . 2617
4.710 findall_in_noblock/4 . . . 2617
4.711 findall_member/4 . . . 2617
4.712 findall_member/5 . . . 2617
4.713 findall_rd_noblock/3 . . . 2617
4.714 findall_rd_noblock/4 . . . 2617
4.715 fine/2 . . . 2617
4.716 finished_by/2 . . . 2617
4.717 finishes/2 . . . 2618
4.718 finite_binary_rational/4 . . . 2618
4.719 first_header_pointer/2 . . . 2618
4.720 first_significant_component/2 . . . 2618
4.721 first_significant_component/3 . . . 2618
4.722 fisher/3 . . . 2618
4.723 flag_group_chk/1 . . . 2618
4.724 flag_groups/1 . . . 2618
4.725 flat_map/2 . . . 2619
4.726 flatten/1 . . . 2619
4.727 flatten/2 . . . 2619
4.728 float//1 . . . 2619
4.729 flush_all/1 . . . 2619
4.730 flush_all/2 . . . 2619
4.731 fold_left/4 . . . 2619
4.732 fold_left_1/3 . . . 2620
4.733 fold_right/4 . . . 2620
4.734 fold_right_1/3 . . . 2620
4.735 fordownto/3 . . . 2620
4.736 fordownto/4 . . . 2620
4.737 fordownto/5 . . . 2620
4.738 foreach/3 . . . 2620
4.739 foreach/4 . . . 2620
4.740 format/1 . . . 2621
4.741 format/2 . . . 2621
4.742 format/3 . . . 2621
4.743 format_date_time/4 . . . 2621
4.744 format_entity_score//2 . . . 2621
4.745 format_object/1 . . . 2621
4.746 format_report/1 . . . 2621
4.747 format_report/2 . . . 2621
4.748 format_report/3 . . . 2622
4.749 format_to_atom/3 . . . 2622
4.750 format_to_chars/3 . . . 2622
4.751 format_to_chars/4 . . . 2622
4.752 format_to_codes/3 . . . 2622
4.753 format_to_codes/4 . . . 2622
4.754 formatted_iban/2 . . . 2622
4.755 forto/3 . . . 2622
4.756 forto/4 . . . 2623
4.757 forto/5 . . . 2623

4.758 forward/1 . . . 2623
4.759 forward/2 . . . 2623
4.760 forward/3 . . . 2623
4.761 forward_substitution/3 . . . 2623
4.762 fractile/3 . . . 2623
4.763 frame_body/2 . . . 2623
4.764 frame_command/2 . . . 2624
4.765 frame_header/3 . . . 2624
4.766 frame_headers/2 . . . 2624
4.767 frame_type/2 . . . 2624
4.768 freeze/2 . . . 2624
4.769 frequency_distribution/2 . . . 2624
4.770 frobenius_norm/2 . . . 2624
4.771 from_expected/2 . . . 2624
4.772 from_generator/2 . . . 2625
4.773 from_generator/3 . . . 2625
4.774 from_generator/4 . . . 2625
4.775 from_goal/2 . . . 2625
4.776 from_goal/3 . . . 2625
4.777 from_goal/4 . . . 2625
4.778 from_goal_or_throw/2 . . . 2626
4.779 from_goal_or_throw/3 . . . 2626
4.780 from_optional/3 . . . 2626
4.781 from_unix_seconds/2 . . . 2626
4.782 frozen/2 . . . 2626
4.783 full_device_path/1 . . . 2626
4.784 func_test/3 . . . 2626
4.785 functional/0 . . . 2626
4.786 fund_currency/5 . . . 2627
4.787 game/4 . . . 2627
4.788 gamma/3 . . . 2627
4.789 gat/4 . . . 2627
4.790 gats/5 . . . 2627
4.791 generate/1 . . . 2627
4.792 generate/2 . . . 2627
4.793 generate/3 . . . 2628
4.794 generate/4 . . . 2629
4.795 generate/8 . . . 2629
4.796 generate_all/2 . . . 2629
4.797 generate_fragment/2 . . . 2629
4.798 generate_frame/3 . . . 2629
4.799 generate_frames/3 . . . 2629
4.800 generate_idle_packet/4 . . . 2629
4.801 genint/2 . . . 2629
4.802 gensym/2 . . . 2630
4.803 geohash_to_int/2 . . . 2630
4.804 geojson_to_json/2 . . . 2630
4.805 geometric/2 . . . 2630
4.806 geometric_mean/2 . . . 2630
4.807 get/1 . . . 2630
4.808 get/3 . . . 2630
4.809 get/4 . . . 2630
4.810 get_field/2 . . . 2631
4.811 get_flag_value/2 . . . 2631

4.812 get_seed/1 . . . 2631
4.813 getrange/5 . . . 2631
4.814 gets/4 . . . 2631
4.815 gets/5 . . . 2631
4.816 git_object_identifier/1 . . . 2631
4.817 gnu/0 . . . 2631
4.818 goal_expansion/2 . . . 2632
4.819 gram_matrix/2 . . . 2632
4.820 graph_coloring/3 . . . 2632
4.821 graph_footer/5 . . . 2632
4.822 graph_header/5 . . . 2632
4.823 ground/1 . . . 2632
4.824 ground_track/5 . . . 2632
4.825 ground_track/6 . . . 2632
4.826 group/1 . . . 2633
4.827 group_by_key/2 . . . 2633
4.828 group_consecutive_by_key/2 . . . 2633
4.829 group_sorted_by_key/2 . . . 2633
4.830 grouped_dataset_groups/2 . . . 2633
4.831 grouped_dataset_items/2 . . . 2633
4.832 grouped_dataset_items/3 . . . 2633
4.833 grouped_dataset_summary/2 . . . 2633
4.834 guess_all_extensions/2 . . . 2634
4.835 guess_all_extensions/3 . . . 2634
4.836 guess_arity/2 . . . 2634
4.837 guess_extension/2 . . . 2634
4.838 guess_extension/3 . . . 2634
4.839 guess_file_type/3 . . . 2634
4.840 guess_file_type/4 . . . 2634
4.841 guess_separator/2 . . . 2634
4.842 guess_type/3 . . . 2635
4.843 guess_type/4 . . . 2635
4.844 gumbel/3 . . . 2635
4.845 halt/0 . . . 2635
4.846 halt/1 . . . 2635
4.847 halt_hook/2 . . . 2635
4.848 hamming/3 . . . 2635
4.849 hamming_distance/3 . . . 2635
4.850 handbook/0 . . . 2636
4.851 harmonic_mean/2 . . . 2636
4.852 has_cycle/1 . . . 2636
4.853 has_negative_cycle/1 . . . 2636
4.854 has_path/3 . . . 2636
4.855 hash/2 . . . 2636
4.856 haversine_distance/3 . . . 2636
4.857 hdel/4 . . . 2636
4.858 head/2 . . . 2637
4.859 head_pred/1 . . . 2637
4.860 help/0 . . . 2637
4.861 help/1 . . . 2637
4.862 help/2 . . . 2637
4.863 help/3 . . . 2637
4.864 hex_digest/4 . . . 2637
4.865 hex_digest/5 . . . 2637

4.866 hex_digit//1 . . . 2638
4.867 hex_digits//1 . . . 2638
4.868 hexists/4 . . . 2638
4.869 hget/4 . . . 2638
4.870 hgetall/3 . . . 2638
4.871 hkeys/3 . . . 2638
4.872 hlen/3 . . . 2638
4.873 home/1 . . . 2638
4.874 homepage/1 . . . 2639
4.875 hset/5 . . . 2639
4.876 hvals/3 . . . 2639
4.877 hypergeometric/4 . . . 2639
4.878 iban/4 . . . 2639
4.879 ibk/3 . . . 2639
4.880 id/2 . . . 2639
4.881 identity_matrix/2 . . . 2639
4.882 if_empty/1 . . . 2640
4.883 if_expected/1 . . . 2640
4.884 if_expected_or_else/2 . . . 2640
4.885 if_invalid/1 . . . 2640
4.886 if_present/1 . . . 2640
4.887 if_present_or_else/2 . . . 2640
4.888 if_unexpected/1 . . . 2640
4.889 if_valid/1 . . . 2640
4.890 if_valid_or_else/2 . . . 2641
4.891 in/1 . . . 2641
4.892 in/2 . . . 2641
4.893 in_degree/3 . . . 2641
4.894 in_list/2 . . . 2641
4.895 in_list/3 . . . 2641
4.896 in_noblock/1 . . . 2641
4.897 in_noblock/2 . . . 2641
4.898 include/3 . . . 2642
4.899 incr/3 . . . 2642
4.900 incr/4 . . . 2642
4.901 incrby/4 . . . 2642
4.902 increase/1 . . . 2642
4.903 increment/0 . . . 2642
4.904 increment_counter/1 . . . 2642
4.905 init/0 . . . 2642
4.906 init/2 . . . 2643
4.907 init1/2 . . . 2643
4.908 init_log_file/2 . . . 2643
4.909 init_tail/2 . . . 2643
4.910 init_tails/2 . . . 2643
4.911 initial_bearing/3 . . . 2643
4.912 initial_channel_reassembly_state/1 . . . 2643
4.913 initial_reassembly_state/1 . . . 2643
4.914 initial_state/1 . . . 2644
4.915 initial_temperature/1 . . . 2644
4.916 initialize/1 . . . 2644
4.917 inits/2 . . . 2644
4.918 inits1/2 . . . 2644
4.919 inorder/2 . . . 2644

4.920 insert/3 . . . 2644
4.921 insert/4 . . . 2644
4.922 insert_after/3 . . . 2645
4.923 insert_all/3 . . . 2645
4.924 insert_aos_packets/4 . . . 2645
4.925 insert_before/3 . . . 2645
4.926 insert_in/4 . . . 2645
4.927 insert_packets/4 . . . 2645
4.928 insert_packets/5 . . . 2645
4.929 insert_tc_segment/3 . . . 2645
4.930 insert_tm_packets/4 . . . 2646
4.931 insert_zone/2 . . . 2646
4.932 install/1 . . . 2646
4.933 install/2 . . . 2646
4.934 install/3 . . . 2646
4.935 install/4 . . . 2646
4.936 installed/0 . . . 2646
4.937 installed/1 . . . 2646
4.938 installed/3 . . . 2647
4.939 installed/4 . . . 2647
4.940 instance/1 . . . 2647
4.941 instance/2 . . . 2647
4.942 instances/1 . . . 2647
4.943 instant_to_utc_date_time/2 . . . 2647
4.944 int_to_geohash/3 . . . 2647
4.945 integer//1 . . . 2647
4.946 integer_to_big_endian_bytes32/2 . . . 2648
4.947 integer_to_big_endian_bytes64/2 . . . 2648
4.948 integer_to_little_endian_bytes32/2 . . . 2648
4.949 internal_error/2 . . . 2648
4.950 internal_os_path/2 . . . 2648
4.951 interpolate_great_circle/4 . . . 2648
4.952 interpolate_rhumb/4 . . . 2648
4.953 interquartile_range/2 . . . 2648
4.954 intersect/2 . . . 2649
4.955 intersection/2 . . . 2649
4.956 intersection/3 . . . 2649
4.957 intersection/4 . . . 2649
4.958 interval_string/2 . . . 2649
4.959 invalid/1 . . . 2649
4.960 invalid_params/2 . . . 2649
4.961 invalid_request/1 . . . 2649
4.962 invalids/2 . . . 2650
4.963 inverse_matrix/2 . . . 2650
4.964 invert_from_cholesky/2 . . . 2650
4.965 invoke/1 . . . 2650
4.966 invoke/2 . . . 2650
4.967 ipv4//1 . . . 2650
4.968 ipv6//1 . . . 2650
4.969 is_absolute_file_name/1 . . . 2650
4.970 is_acyclic/1 . . . 2651
4.971 is_alpha/1 . . . 2651
4.972 is_alphanumeric/1 . . . 2651
4.973 is_ascii/1 . . . 2651

4.974 is_batch/1 . . . 2651
4.975 is_bin_digit/1 . . . 2651
4.976 is_bipartite/1 . . . 2651
4.977 is_clockwise_polygon/1 . . . 2651
4.978 is_complete/1 . . . 2652
4.979 is_connected/1 . . . 2652
4.980 is_control/1 . . . 2652
4.981 is_dec_digit/1 . . . 2652
4.982 is_empty/0 . . . 2652
4.983 is_end_of_line/1 . . . 2652
4.984 is_error_response/1 . . . 2652
4.985 is_expected/0 . . . 2652
4.986 is_false/1 . . . 2653
4.987 is_hex_digit/1 . . . 2653
4.988 is_invalid/0 . . . 2653
4.989 is_layout/1 . . . 2653
4.990 is_letter/1 . . . 2653
4.991 is_lower_case/1 . . . 2653
4.992 is_newline/1 . . . 2653
4.993 is_notification/1 . . . 2653
4.994 is_null/1 . . . 2654
4.995 is_object/1 . . . 2654
4.996 is_octal_digit/1 . . . 2654
4.997 is_period/1 . . . 2654
4.998 is_prefix_of/2 . . . 2654
4.999 is_present/0 . . . 2654
4.1000 is_punctuation/1 . . . 2654
4.1001 is_quote/1 . . . 2654
4.1002 is_request/1 . . . 2655
4.1003 is_response/1 . . . 2655
4.1004 is_sparse/1 . . . 2655
4.1005 is_subsequence_of/2 . . . 2655
4.1006 is_suffix_of/2 . . . 2655
4.1007 is_tree/1 . . . 2655
4.1008 is_true/1 . . . 2655
4.1009 is_unexpected/0 . . . 2655
4.1010 is_upper_case/1 . . . 2656
4.1011 is_valid/0 . . . 2656
4.1012 is_valid_polygon/1 . . . 2656
4.1013 is_void/1 . . . 2656
4.1014 is_vowel/1 . . . 2656
4.1015 is_white_space/1 . . . 2656
4.1016 item/1 . . . 2656
4.1017 item/2 . . . 2656
4.1018 items/1 . . . 2657
4.1019 iterator_element/2 . . . 2657
4.1020 jaccard_index/3 . . . 2657
4.1021 jaro/3 . . . 2657
4.1022 jaro_winkler/3 . . . 2657
4.1023 join/3 . . . 2657
4.1024 join_all/3 . . . 2657
4.1025 join_aos_packet_zone/3 . . . 2657
4.1026 join_packet_zone/4 . . . 2658
4.1027 json_to_geojson/2 . . . 2658

4.1028 jump/3 . . . 2658
4.1029 jump_all/3 . . . 2658
4.1030 jump_all_block/3 . . . 2658
4.1031 k_distinct_subsequence/3 . . . 2658
4.1032 k_distinct_subsequences/3 . . . 2658
4.1033 k_permutation/3 . . . 2658
4.1034 k_permutation/4 . . . 2659
4.1035 k_permutations/3 . . . 2659
4.1036 k_permutations/4 . . . 2659
4.1037 key/2 . . . 2659
4.1038 keys/2 . . . 2659
4.1039 keys/3 . . . 2659
4.1040 keys_values/3 . . . 2659
4.1041 keysort/2 . . . 2659
4.1042 kill/1 . . . 2660
4.1043 kill/2 . . . 2660
4.1044 known_zone_id/1 . . . 2660
4.1045 kurtosis/2 . . . 2660
4.1046 lambert_azimuthal_equal_area/3 . . . 2660
4.1047 lambert_azimuthal_equal_area_inverse/3 . . . 2660
4.1048 language/3 . . . 2660
4.1049 language/5 . . . 2660
4.1050 language_code/5 . . . 2661
4.1051 language_group/2 . . . 2661
4.1052 language_object/2 . . . 2661
4.1053 last/2 . . . 2661
4.1054 leaf/1 . . . 2661
4.1055 leaf_class/1 . . . 2661
4.1056 leaf_classes/1 . . . 2661
4.1057 leaf_instance/1 . . . 2662
4.1058 leaf_instances/1 . . . 2662
4.1059 leap_effective_date/2 . . . 2662
4.1060 leap_offset_at_utc_unix/2 . . . 2662
4.1061 leap_second_date/2 . . . 2662
4.1062 leap_seconds_entries/1 . . . 2662
4.1063 leap_seconds_source/1 . . . 2662
4.1064 leap_year/1 . . . 2662
4.1065 learn/0 . . . 2663
4.1066 learn/1 . . . 2663
4.1067 learn/2 . . . 2663
4.1068 learn/3 . . . 2663
4.1069 learn_seq/2 . . . 2663
4.1070 learn_with_timeout/4 . . . 2664
4.1071 leash/1 . . . 2664
4.1072 leashing/1 . . . 2664
4.1073 least_common_multiple/2 . . . 2664
4.1074 least_squares/3 . . . 2664
4.1075 leaves/1 . . . 2664
4.1076 leaves/2 . . . 2664
4.1077 length/2 . . . 2664
4.1078 levenshtein/3 . . . 2665
4.1079 libraries/0 . . . 2665
4.1080 libraries/1 . . . 2665
4.1081 libraries/2 . . . 2665

4.1082 libraries/3 . . . 2665
4.1083 library/1 . . . 2665
4.1084 library/2 . . . 2666
4.1085 library_mutants/2 . . . 2666
4.1086 library_mutants/3 . . . 2666
4.1087 library_score/2 . . . 2666
4.1088 license/1 . . . 2666
4.1089 linda/0 . . . 2666
4.1090 linda/1 . . . 2666
4.1091 linda_client/1 . . . 2667
4.1092 linda_client/2 . . . 2667
4.1093 linda_timeout/2 . . . 2667
4.1094 line_to_chars/2 . . . 2667
4.1095 line_to_chars/3 . . . 2667
4.1096 line_to_codes/2 . . . 2667
4.1097 line_to_codes/3 . . . 2667
4.1098 linear_regression/4 . . . 2667
4.1099 lint/0 . . . 2668
4.1100 lint/1 . . . 2668
4.1101 lint/2 . . . 2668
4.1102 list/0 . . . 2668
4.1103 list_to_array/2 . . . 2668
4.1104 listing/0 . . . 2668
4.1105 listing/1 . . . 2668
4.1106 little_endian_word32/2 . . . 2668
4.1107 llen/3 . . . 2669
4.1108 load/1 . . . 2669
4.1109 load/2 . . . 2669
4.1110 load_config/1 . . . 2669
4.1111 load_coverage_config/1 . . . 2669
4.1112 load_dut1_override/1 . . . 2669
4.1113 load_leap_seconds_override/1 . . . 2669
4.1114 load_program/1 . . . 2670
4.1115 loaded_file/1 . . . 2670
4.1116 loaded_file_property/2 . . . 2670
4.1117 loaded_files_topological_sort/1 . . . 2670
4.1118 loaded_files_topological_sort/2 . . . 2670
4.1119 loaded_pack/3 . . . 2670
4.1120 loaded_pack_dependency/6 . . . 2670
4.1121 loaded_pack_file/4 . . . 2670
4.1122 loader_file/1 . . . 2671
4.1123 local_abbreviation/2 . . . 2671
4.1124 local_abbreviation/3 . . . 2671
4.1125 local_abbreviation/4 . . . 2671
4.1126 local_abbreviation_reified/2 . . . 2671
4.1127 local_abbreviation_reified/3 . . . 2671
4.1128 local_abbreviation_reified/4 . . . 2671
4.1129 local_abbreviation_with_resolution/3 . . . 2671
4.1130 local_abbreviation_with_resolution/4 . . . 2672
4.1131 local_abbreviation_with_resolution/5 . . . 2672
4.1132 local_daylight_saving_time/2 . . . 2672
4.1133 local_daylight_saving_time/3 . . . 2672
4.1134 local_daylight_saving_time/4 . . . 2672
4.1135 local_daylight_saving_time_reified/2 . . . 2672

4.1136 local_daylight_saving_time_reified/3 . . . 2672
4.1137 local_daylight_saving_time_reified/4 . . . 2672
4.1138 local_daylight_saving_time_with_resolution/3 . . . 2673
4.1139 local_daylight_saving_time_with_resolution/4 . . . 2673
4.1140 local_daylight_saving_time_with_resolution/5 . . . 2673
4.1141 local_offset/2 . . . 2673
4.1142 local_offset/3 . . . 2673
4.1143 local_offset/4 . . . 2673
4.1144 local_offset_reified/2 . . . 2673
4.1145 local_offset_reified/3 . . . 2673
4.1146 local_offset_reified/4 . . . 2674
4.1147 local_offset_with_resolution/3 . . . 2674
4.1148 local_offset_with_resolution/4 . . . 2674
4.1149 local_offset_with_resolution/5 . . . 2674
4.1150 local_tangent_plane/3 . . . 2674
4.1151 local_tangent_plane_inverse/3 . . . 2674
4.1152 local_time_type/2 . . . 2674
4.1153 local_time_type/3 . . . 2674
4.1154 local_time_type/4 . . . 2675
4.1155 local_time_type_reified/2 . . . 2675
4.1156 local_time_type_reified/3 . . . 2675
4.1157 local_time_type_reified/4 . . . 2675
4.1158 local_time_type_with_resolution/3 . . . 2675
4.1159 local_time_type_with_resolution/4 . . . 2675
4.1160 local_time_type_with_resolution/5 . . . 2675
4.1161 local_to_utc/3 . . . 2675
4.1162 local_to_utc_tz/3 . . . 2676
4.1163 local_to_utc_tz_with_resolution/4 . . . 2676
4.1164 locked_pairs/2 . . . 2676
4.1165 log/3 . . . 2676
4.1166 log_event/2 . . . 2676
4.1167 log_file/2 . . . 2676
4.1168 logging/1 . . . 2676
4.1169 logging/3 . . . 2676
4.1170 logistic/3 . . . 2677
4.1171 lognormal/3 . . . 2677
4.1172 logseries/2 . . . 2677
4.1173 logtalk_packs/0 . . . 2677
4.1174 logtalk_packs/1 . . . 2677
4.1175 long_flags/1 . . . 2677
4.1176 longest_common_increasing_subsequence/3 . . . 2677
4.1177 longest_common_subsequence/3 . . . 2677
4.1178 longest_common_subsequence_length/3 . . . 2678
4.1179 longest_common_substring/3 . . . 2678
4.1180 longest_decreasing_subsequence/2 . . . 2678
4.1181 longest_increasing_subsequence/2 . . . 2678
4.1182 longest_repeating_subsequence/2 . . . 2678
4.1183 lookup/2 . . . 2678
4.1184 lookup/3 . . . 2678
4.1185 lookup/4 . . . 2678
4.1186 lookup_in/3 . . . 2679
4.1187 lower_triangular_part/2 . . . 2679
4.1188 lower_triangular_part/3 . . . 2679
4.1189 lower_upper/2 . . . 2679

4.1190 lpop/3 . . . 2679
4.1191 lpush/4 . . . 2679
4.1192 lrange/5 . . . 2679
4.1193 lrem/5 . . . 2679
4.1194 ltrim/5 . . . 2680
4.1195 lucas/2 . . . 2680
4.1196 magic/2 . . . 2680
4.1197 magicise/4 . . . 2680
4.1198 make_directory/1 . . . 2680
4.1199 make_directory_path/1 . . . 2680
4.1200 make_set/3 . . . 2680
4.1201 man/1 . . . 2680
4.1202 manhattan_distance/3 . . . 2681
4.1203 manhattan_norm/2 . . . 2681
4.1204 map/2 . . . 2681
4.1205 map/3 . . . 2681
4.1206 map/4 . . . 2681
4.1207 map/5 . . . 2682
4.1208 map/6 . . . 2682
4.1209 map/7 . . . 2682
4.1210 map/8 . . . 2682
4.1211 map_both/3 . . . 2682
4.1212 map_catching/2 . . . 2682
4.1213 map_element/2 . . . 2682
4.1214 map_id/2 . . . 2682
4.1215 map_invalid/2 . . . 2683
4.1216 map_or_else/3 . . . 2683
4.1217 map_reduce/5 . . . 2683
4.1218 map_unexpected/2 . . . 2683
4.1219 mask32/1 . . . 2683
4.1220 mask64/1 . . . 2683
4.1221 master_channel_frame_count/2 . . . 2683
4.1222 materialize/0 . . . 2683
4.1223 matrix_column/3 . . . 2684
4.1224 matrix_column_means/2 . . . 2684
4.1225 matrix_diagonal/2 . . . 2684
4.1226 matrix_diagonal/3 . . . 2684
4.1227 matrix_matrix_product/3 . . . 2684
4.1228 matrix_rank/2 . . . 2684
4.1229 matrix_rank/3 . . . 2684
4.1230 matrix_row/3 . . . 2684
4.1231 matrix_row_means/2 . . . 2685
4.1232 matrix_trace/2 . . . 2685
4.1233 matrix_value/4 . . . 2685
4.1234 matrix_vector_product/3 . . . 2685
4.1235 max/2 . . . 2685
4.1236 max/3 . . . 2685
4.1237 max_clauses/1 . . . 2685
4.1238 max_inv_preds/1 . . . 2685
4.1239 max_path/5 . . . 2686
4.1240 max_size/1 . . . 2686
4.1241 max_tree/3 . . . 2686
4.1242 maximal_cliques/2 . . . 2686
4.1243 maximum_cliques/2 . . . 2686

4.1244 maybe/0 . . . 2686
4.1245 maybe/1 . . . 2686
4.1246 maybe/2 . . . 2686
4.1247 maybe_call/1 . . . 2687
4.1248 maybe_call/2 . . . 2687
4.1249 mean_center/2 . . . 2687
4.1250 mean_deviation/2 . . . 2687
4.1251 mean_squared_error/3 . . . 2687
4.1252 measurement/4 . . . 2687
4.1253 median/2 . . . 2687
4.1254 median_deviation/2 . . . 2687
4.1255 meets/2 . . . 2688
4.1256 member/2 . . . 2688
4.1257 memberchk/2 . . . 2688
4.1258 merge/3 . . . 2688
4.1259 message_body/2 . . . 2688
4.1260 message_delivery_tag/2 . . . 2688
4.1261 message_exchange/2 . . . 2688
4.1262 message_hook/4 . . . 2689
4.1263 message_prefix_file/6 . . . 2689
4.1264 message_prefix_stream/4 . . . 2689
4.1265 message_properties/2 . . . 2689
4.1266 message_property/3 . . . 2689
4.1267 message_routing_key/2 . . . 2689
4.1268 message_tokens//2 . . . 2689
4.1269 met_by/2 . . . 2689
4.1270 meta/1 . . . 2690
4.1271 meta_type/3 . . . 2690
4.1272 metaphone/2 . . . 2690
4.1273 metaphone_match/2 . . . 2690
4.1274 metarule/6 . . . 2690
4.1275 metarule_next_id/1 . . . 2690
4.1276 method/2 . . . 2690
4.1277 method_not_found/2 . . . 2690
4.1278 mget/3 . . . 2691
4.1279 mibenum/1 . . . 2691
4.1280 midpoint/3 . . . 2691
4.1281 milliseconds/2 . . . 2691
4.1282 min/2 . . . 2691
4.1283 min/3 . . . 2691
4.1284 min_clauses/1 . . . 2691
4.1285 min_distances/3 . . . 2692
4.1286 min_max/3 . . . 2692
4.1287 min_max_normalization/2 . . . 2692
4.1288 min_path/5 . . . 2692
4.1289 min_path_bellman_ford/5 . . . 2692
4.1290 min_paths/3 . . . 2692
4.1291 min_predecessors/3 . . . 2692
4.1292 min_tree/3 . . . 2692
4.1293 mine/2 . . . 2693
4.1294 mine/3 . . . 2693
4.1295 minimum_enclosing_circle/3 . . . 2693
4.1296 minkowski_distance/4 . . . 2693
4.1297 modes/2 . . . 2693

4.1298 module_property/2 . . . 2693
4.1299 monitor/1 . . . 2693
4.1300 monitor/4 . . . 2693
4.1301 monitor_activated/0 . . . 2694
4.1302 monitored/1 . . . 2694
4.1303 monitors/1 . . . 2694
4.1304 month/2 . . . 2694
4.1305 month_weekday_date/5 . . . 2694
4.1306 mset/3 . . . 2694
4.1307 msort/2 . . . 2694
4.1308 msort/3 . . . 2694
4.1309 mul32/3 . . . 2695
4.1310 mul64/3 . . . 2695
4.1311 multinomial/2 . . . 2695
4.1312 multiset/3 . . . 2695
4.1313 multiset/4 . . . 2695
4.1314 multiset_index/4 . . . 2695
4.1315 multiset_index/5 . . . 2695
4.1316 multisets/3 . . . 2695
4.1317 multisets/4 . . . 2696
4.1318 mutation/2 . . . 2696
4.1319 mutation/3 . . . 2696
4.1320 nack/3 . . . 2696
4.1321 name/1 . . . 2696
4.1322 name_of_day/3 . . . 2696
4.1323 name_of_month/3 . . . 2696
4.1324 nan_kind/2 . . . 2697
4.1325 nan_payload/2 . . . 2697
4.1326 nan_representation/1 . . . 2697
4.1327 natural//1 . . . 2697
4.1328 nearest_coordinate/5 . . . 2697
4.1329 nearest_point_on_polyline/4 . . . 2697
4.1330 nearest_point_on_segment/4 . . . 2697
4.1331 neighbor_state/2 . . . 2697
4.1332 neighbor_state/3 . . . 2698
4.1333 neighbors/2 . . . 2698
4.1334 neighbors/3 . . . 2698
4.1335 neighbors_bits/3 . . . 2698
4.1336 neighbors_int/3 . . . 2698
4.1337 network/1 . . . 2698
4.1338 new/1 . . . 2698
4.1339 new/2 . . . 2699
4.1340 new/3 . . . 2699
4.1341 new_line//0 . . . 2699
4.1342 new_lines//0 . . . 2699
4.1343 new_matrix/4 . . . 2699
4.1344 new_vector/3 . . . 2699
4.1345 new_vector_like/2 . . . 2699
4.1346 next/2 . . . 2700
4.1347 next/3 . . . 2700
4.1348 next/4 . . . 2700
4.1349 next_arrangement/3 . . . 2700
4.1350 next_cartesian_tuple/3 . . . 2700
4.1351 next_combination/3 . . . 2700

4.1352 next_derangement/3 . . . 2700
4.1353 next_multiset/3 . . . 2700
4.1354 next_partition/3 . . . 2701
4.1355 next_partition/4 . . . 2701
4.1356 next_permutation/2 . . . 2701
4.1357 nextto/3 . . . 2701
4.1358 node/7 . . . 2701
4.1359 nodebug/0 . . . 2701
4.1360 nodes/2 . . . 2701
4.1361 nolog/3 . . . 2701
4.1362 nologall/0 . . . 2702
4.1363 non_blank//1 . . . 2702
4.1364 non_blanks//1 . . . 2702
4.1365 nonempty_subsequence/2 . . . 2702
4.1366 nonempty_subsequences/2 . . . 2702
4.1367 normal/3 . . . 2702
4.1368 normal_element/2 . . . 2702
4.1369 normalize/2 . . . 2702
4.1370 normalize_coordinate/2 . . . 2703
4.1371 normalize_date_time/2 . . . 2703
4.1372 normalize_polygon_orientation/3 . . . 2703
4.1373 normalize_range/2 . . . 2703
4.1374 normalize_range/4 . . . 2703
4.1375 normalize_scalar/2 . . . 2703
4.1376 normalize_unit/2 . . . 2703
4.1377 normalize_vector/2 . . . 2703
4.1378 normalize_vector/3 . . . 2704
4.1379 nospy/1 . . . 2704
4.1380 nospy/3 . . . 2704
4.1381 nospy/4 . . . 2704
4.1382 nospyall/0 . . . 2704
4.1383 not64/2 . . . 2704
4.1384 note/2 . . . 2704
4.1385 note/3 . . . 2704
4.1386 notification/2 . . . 2705
4.1387 notification/3 . . . 2705
4.1388 notrace/0 . . . 2705
4.1389 now/3 . . . 2705
4.1390 nth0/3 . . . 2705
4.1391 nth0/4 . . . 2705
4.1392 nth1/3 . . . 2705
4.1393 nth1/4 . . . 2705
4.1394 nth_arrangement/4 . . . 2706
4.1395 nth_arrangement/5 . . . 2706
4.1396 nth_cartesian_tuple/3 . . . 2706
4.1397 nth_cartesian_tuple/4 . . . 2706
4.1398 nth_combination/4 . . . 2706
4.1399 nth_combination/5 . . . 2706
4.1400 nth_derangement/3 . . . 2706
4.1401 nth_derangement/4 . . . 2706
4.1402 nth_distinct_arrangement/4 . . . 2707
4.1403 nth_distinct_cartesian_tuple/3 . . . 2707
4.1404 nth_distinct_combination/4 . . . 2707
4.1405 nth_distinct_derangement/3 . . . 2707

4.1406 nth_distinct_multiset/4 . . . 2707
4.1407 nth_distinct_partition/3 . . . 2707
4.1408 nth_distinct_partition/4 . . . 2707
4.1409 nth_distinct_partition/5 . . . 2707
4.1410 nth_distinct_permutation/3 . . . 2708
4.1411 nth_multiset/4 . . . 2708
4.1412 nth_multiset/5 . . . 2708
4.1413 nth_partition/3 . . . 2708
4.1414 nth_partition/4 . . . 2708
4.1415 nth_partition/5 . . . 2708
4.1416 nth_permutation/3 . . . 2708
4.1417 nth_permutation/4 . . . 2708
4.1418 null/1 . . . 2709
4.1419 null_device_path/1 . . . 2709
4.1420 null_space/2 . . . 2709
4.1421 null_space/3 . . . 2709
4.1422 number//1 . . . 2709
4.1423 number_of_edges/2 . . . 2709
4.1424 number_of_tests/1 . . . 2709
4.1425 number_of_vertices/2 . . . 2709
4.1426 number_string/2 . . . 2710
4.1427 numbervars/1 . . . 2710
4.1428 numbervars/3 . . . 2710
4.1429 occurrences/2 . . . 2710
4.1430 occurrences/3 . . . 2710
4.1431 occurs/2 . . . 2710
4.1432 ocf/2 . . . 2710
4.1433 ocf_flag/2 . . . 2710
4.1434 of/2 . . . 2711
4.1435 of_expected/2 . . . 2711
4.1436 of_invalid/2 . . . 2711
4.1437 of_invalids/2 . . . 2711
4.1438 of_unexpected/2 . . . 2711
4.1439 of_valid/2 . . . 2711
4.1440 offset/2 . . . 2711
4.1441 offset/3 . . . 2711
4.1442 offset/4 . . . 2712
4.1443 one_or_more//0 . . . 2712
4.1444 one_or_more//1 . . . 2712
4.1445 one_or_more//2 . . . 2712
4.1446 operating_system_machine/1 . . . 2712
4.1447 operating_system_name/1 . . . 2712
4.1448 operating_system_release/1 . . . 2712
4.1449 operating_system_type/1 . . . 2712
4.1450 option/2 . . . 2713
4.1451 option/3 . . . 2713
4.1452 or/2 . . . 2713
4.1453 or64/3 . . . 2713
4.1454 or_else/2 . . . 2713
4.1455 or_else_call/2 . . . 2713
4.1456 or_else_fail/1 . . . 2713
4.1457 or_else_get/2 . . . 2714
4.1458 or_else_throw/1 . . . 2714
4.1459 or_else_throw/2 . . . 2714

4.1460 order/1 . . . 2714
4.1461 originator/1 . . . 2714
4.1462 orphaned/0 . . . 2714
4.1463 orphaned/2 . . . 2714
4.1464 out/1 . . . 2715
4.1465 out/2 . . . 2715
4.1466 out_degree/3 . . . 2715
4.1467 outdated/0 . . . 2715
4.1468 outdated/1 . . . 2715
4.1469 outdated/2 . . . 2715
4.1470 outdated/4 . . . 2715
4.1471 outdated/5 . . . 2715
4.1472 outer_product/3 . . . 2716
4.1473 output_file_name/2 . . . 2716
4.1474 output_schema/2 . . . 2716
4.1475 overlapped_by/2 . . . 2716
4.1476 overlaps/2 . . . 2716
4.1477 pack_dependency/6 . . . 2716
4.1478 pack_metadata/4 . . . 2716
4.1479 pack_object/3 . . . 2716
4.1480 pack_property/4 . . . 2717
4.1481 package/1 . . . 2717
4.1482 packet_order_flag/2 . . . 2717
4.1483 packetize_aos_frames/7 . . . 2717
4.1484 packetize_aos_frames/8 . . . 2717
4.1485 packetize_aos_packets/6 . . . 2717
4.1486 packetize_aos_packets/7 . . . 2717
4.1487 packetize_tm_frames/7 . . . 2717
4.1488 packetize_tm_frames/8 . . . 2718
4.1489 packetize_tm_packets/6 . . . 2718
4.1490 packetize_tm_packets/7 . . . 2718
4.1491 packets/2 . . . 2718
4.1492 pad_md/4 . . . 2718
4.1493 pairwise_dataset_connected_components/2 . . . 2718
4.1494 pairwise_dataset_declared_items/2 . . . 2718
4.1495 pairwise_dataset_items/2 . . . 2718
4.1496 pairwise_dataset_matchups/2 . . . 2719
4.1497 pairwise_dataset_preferences/2 . . . 2719
4.1498 pairwise_dataset_summary/2 . . . 2719
4.1499 pairwise_dataset_win_totals/2 . . . 2719
4.1500 pairwise_measurement_dataset_connected_components/2 . . . 2719
4.1501 pairwise_measurement_dataset_declared_items/2 . . . 2719
4.1502 pairwise_measurement_dataset_items/2 . . . 2719
4.1503 pairwise_measurement_dataset_measurements/2 . . . 2719
4.1504 pairwise_measurement_dataset_summary/2 . . . 2720
4.1505 params/2 . . . 2720
4.1506 parent/1 . . . 2720
4.1507 parent/2 . . . 2720
4.1508 parenthesis/2 . . . 2720
4.1509 parents/1 . . . 2720
4.1510 parse/2 . . . 2720
4.1511 parse/3 . . . 2721
4.1512 parse/4 . . . 2721
4.1513 parse/5 . . . 2721

4.1514 parse_all/2 . . . 2722
4.1515 parse_domain/2 . . . 2722
4.1516 parse_domain/3 . . . 2722
4.1517 parse_error/1 . . . 2722
4.1518 parse_fragment/2 . . . 2722
4.1519 parse_frame/3 . . . 2722
4.1520 parse_frames/3 . . . 2722
4.1521 parse_lines/4 . . . 2722
4.1522 parse_problem/2 . . . 2723
4.1523 parse_problem/3 . . . 2723
4.1524 partial_map/4 . . . 2723
4.1525 partition/2 . . . 2723
4.1526 partition/3 . . . 2723
4.1527 partition/4 . . . 2723
4.1528 partition/5 . . . 2723
4.1529 partition/6 . . . 2724
4.1530 partition_index/3 . . . 2724
4.1531 partition_index/4 . . . 2724
4.1532 partition_index/5 . . . 2724
4.1533 partition_number/2 . . . 2724
4.1534 partitions/2 . . . 2724
4.1535 partitions/3 . . . 2724
4.1536 partitions/4 . . . 2724
4.1537 path/3 . . . 2725
4.1538 path_concat/3 . . . 2725
4.1539 path_consistency/2 . . . 2725
4.1540 pattern_miner_options/2 . . . 2725
4.1541 peek_back/2 . . . 2725
4.1542 peek_front/2 . . . 2725
4.1543 pending_data/2 . . . 2725
4.1544 pending_fragments/2 . . . 2725
4.1545 pending_packets/2 . . . 2726
4.1546 percentile/3 . . . 2726
4.1547 period/1 . . . 2726
4.1548 permutation/2 . . . 2726
4.1549 permutation/3 . . . 2726
4.1550 permutation_index/3 . . . 2726
4.1551 permutation_index/4 . . . 2726
4.1552 permutations/2 . . . 2727
4.1553 permutations/3 . . . 2727
4.1554 persist/3 . . . 2727
4.1555 pid/1 . . . 2727
4.1556 pin/0 . . . 2727
4.1557 pin/1 . . . 2727
4.1558 pinned/1 . . . 2727
4.1559 plus/3 . . . 2727
4.1560 point_in_polygon/2 . . . 2728
4.1561 point_to_polyline_distance/3 . . . 2728
4.1562 poisson/2 . . . 2728
4.1563 polygon_area/2 . . . 2728
4.1564 polygon_bounding_box/2 . . . 2728
4.1565 polygon_centroid/2 . . . 2728
4.1566 polygon_covering/4 . . . 2728
4.1567 polygon_orientation/2 . . . 2728

4.1568 polygon_perimeter/2 . . . 2729
4.1569 polygon_perimeter/3 . . . 2729
4.1570 polygons_intersect/2 . . . 2729
4.1571 polyline_covering/4 . . . 2729
4.1572 polyline_length/2 . . . 2729
4.1573 polyline_length/3 . . . 2729
4.1574 polyline_resample/3 . . . 2729
4.1575 polyline_simplify/3 . . . 2729
4.1576 polyline_split_at_distance/4 . . . 2730
4.1577 pop_back/3 . . . 2730
4.1578 pop_front/3 . . . 2730
4.1579 port/5 . . . 2730
4.1580 portray_clause/1 . . . 2730
4.1581 possible/4 . . . 2730
4.1582 postorder/2 . . . 2730
4.1583 power/2 . . . 2730
4.1584 power_sequence/4 . . . 2731
4.1585 power_set/2 . . . 2731
4.1586 powerset/2 . . . 2731
4.1587 pp/1 . . . 2731
4.1588 pprint/1 . . . 2731
4.1589 precision/1 . . . 2731
4.1590 precision/3 . . . 2731
4.1591 predicate/1 . . . 2731
4.1592 predicate/2 . . . 2732
4.1593 predicate/3 . . . 2732
4.1594 predicate_info_pair_score_hook/4 . . . 2732
4.1595 predicate_info_score_hook/3 . . . 2732
4.1596 predicate_mode_score_hook/3 . . . 2732
4.1597 predicate_mode_score_hook/5 . . . 2732
4.1598 predicate_mutants/3 . . . 2732
4.1599 predicate_mutants/4 . . . 2733
4.1600 predicate_stratum/3 . . . 2733
4.1601 predicates/2 . . . 2733
4.1602 predicates/3 . . . 2733
4.1603 predict/3 . . . 2733
4.1604 predict/4 . . . 2733
4.1605 predict_distribution/3 . . . 2733
4.1606 predict_probabilities/3 . . . 2734
4.1607 predict_probabilities/4 . . . 2734
4.1608 predict_scores/3 . . . 2734
4.1609 preference/3 . . . 2734
4.1610 preferred_mime_name/1 . . . 2734
4.1611 prefix/0 . . . 2734
4.1612 prefix/1 . . . 2734
4.1613 prefix/2 . . . 2734
4.1614 prefix/3 . . . 2735
4.1615 prefix_data/2 . . . 2735
4.1616 preorder/2 . . . 2735
4.1617 prepend/3 . . . 2735
4.1618 previous/2 . . . 2735
4.1619 previous/3 . . . 2735
4.1620 previous/4 . . . 2735
4.1621 previous_arrangement/3 . . . 2735

4.1622previous_cartesian_tuple/3 . . . 2736
4.1623previous_combination/3 . . . 2736
4.1624previous_derangement/3 . . . 2736
4.1625previous_multiset/3 . . . 2736
4.1626previous_partition/3 . . . 2736
4.1627previous_partition/4 . . . 2736
4.1628previous_permutation/2 . . . 2736
4.1629primary_office_bic/1 . . . 2736
4.1630print_anomaly_detector/1 . . . 2737
4.1631print_classifier/1 . . . 2737
4.1632print_clusterer/1 . . . 2737
4.1633print_dimension_reducer/1 . . . 2737
4.1634print_flags/0 . . . 2737
4.1635print_flags/1 . . . 2737
4.1636print_message/3 . . . 2737
4.1637print_message_token/4 . . . 2737
4.1638print_message_tokens/3 . . . 2738
4.1639print_pattern_miner/1 . . . 2738
4.1640print_ranker/1 . . . 2738
4.1641print_regressor/1 . . . 2738
4.1642product/2 . . . 2738
4.1643product/3 . . . 2738
4.1644program_to_clauses/2 . . . 2738
4.1645progress/5 . . . 2738
4.1646prompt_get/3 . . . 2739
4.1647prompts/1 . . . 2739
4.1648propagate/2 . . . 2739
4.1649propagate/3 . . . 2739
4.1650propagate/4 . . . 2739
4.1651propagate/5 . . . 2739
4.1652propagate_state/4 . . . 2739
4.1653propagate_state/5 . . . 2739
4.1654propagation_triple/2 . . . 2740
4.1655propagation_triples/2 . . . 2740
4.1656proper_prefix/2 . . . 2740
4.1657proper_prefix/3 . . . 2740
4.1658proper_subsequence/2 . . . 2740
4.1659proper_suffix/2 . . . 2740
4.1660proper_suffix/3 . . . 2740
4.1661property/1 . . . 2740
4.1662prove/2 . . . 2741
4.1663prove/3 . . . 2741
4.1664provides/2 . . . 2741
4.1665prune/3 . . . 2741
4.1666prune/5 . . . 2741
4.1667pseudo_inverse/2 . . . 2741
4.1668pseudo_inverse/3 . . . 2741
4.1669push_back/3 . . . 2741
4.1670push_front/3 . . . 2742
4.1671qr_decomposition/3 . . . 2742
4.1672quartiles/4 . . . 2742
4.1673query/1 . . . 2742
4.1674query/2 . . . 2742
4.1675question_hook/6 . . . 2742

4.1676 question_prompt_stream/4 . . . 2742
4.1677 queue_bind/4 . . . 2742
4.1678 queue_declare/3 . . . 2743
4.1679 queue_delete/3 . . . 2743
4.1680 queue_purge/2 . . . 2743
4.1681 queue_unbind/4 . . . 2743
4.1682 quick_check/1 . . . 2743
4.1683 quick_check/2 . . . 2743
4.1684 quick_check/3 . . . 2743
4.1685 random/1 . . . 2743
4.1686 random/3 . . . 2744
4.1687 random_arrangement/3 . . . 2744
4.1688 random_cartesian_tuple/2 . . . 2744
4.1689 random_combination/3 . . . 2744
4.1690 random_derangement/2 . . . 2744
4.1691 random_distinct_arrangement/3 . . . 2744
4.1692 random_distinct_cartesian_tuple/2 . . . 2744
4.1693 random_distinct_combination/3 . . . 2744
4.1694 random_distinct_derangement/2 . . . 2745
4.1695 random_distinct_multiset/3 . . . 2745
4.1696 random_distinct_partition/2 . . . 2745
4.1697 random_distinct_partition/3 . . . 2745
4.1698 random_distinct_permutation/2 . . . 2745
4.1699 random_multiset/3 . . . 2745
4.1700 random_node/1 . . . 2745
4.1701 random_partition/2 . . . 2745
4.1702 random_partition/3 . . . 2746
4.1703 random_permutation/2 . . . 2746
4.1704 random_subsequence/2 . . . 2746
4.1705 random_tree/1 . . . 2746
4.1706 randomize/1 . . . 2746
4.1707 randseq/4 . . . 2746
4.1708 randset/4 . . . 2746
4.1709 range/2 . . . 2746
4.1710 rank/3 . . . 2747
4.1711 rank_correlation/3 . . . 2747
4.1712 ranker_options/2 . . . 2747
4.1713 rd/1 . . . 2747
4.1714 rd/2 . . . 2747
4.1715 rd_list/2 . . . 2747
4.1716 rd_list/3 . . . 2747
4.1717 rd_noblock/1 . . . 2747
4.1718 rd_noblock/2 . . . 2748
4.1719 rdirectories/1 . . . 2748
4.1720 rdirectories/2 . . . 2748
4.1721 rdirectory/1 . . . 2748
4.1722 rdirectory/2 . . . 2748
4.1723 rdirectory/3 . . . 2748
4.1724 rdirectory_score/2 . . . 2749
4.1725 reachable/3 . . . 2749
4.1726 read_file/2 . . . 2749
4.1727 read_file/3 . . . 2749
4.1728 read_file_by_line/2 . . . 2749
4.1729 read_file_by_line/3 . . . 2749

4.1730 read_framed_message/2 . . . 2749
4.1731 read_from_atom/2 . . . 2750
4.1732 read_from_chars/2 . . . 2750
4.1733 read_from_codes/2 . . . 2750
4.1734 read_message/2 . . . 2750
4.1735 read_mime_types/2 . . . 2750
4.1736 read_only_device_path/1 . . . 2750
4.1737 read_stream/2 . . . 2750
4.1738 read_stream/3 . . . 2750
4.1739 read_stream_by_line/2 . . . 2751
4.1740 read_stream_by_line/3 . . . 2751
4.1741 read_term_from_atom/3 . . . 2751
4.1742 read_term_from_chars/3 . . . 2751
4.1743 read_term_from_chars/4 . . . 2751
4.1744 read_term_from_codes/3 . . . 2751
4.1745 read_term_from_codes/4 . . . 2751
4.1746 readme/1 . . . 2751
4.1747 readme/2 . . . 2752
4.1748 reassemble_aos_frames/5 . . . 2752
4.1749 reassemble_aos_frames/6 . . . 2752
4.1750 reassemble_aos_frames/7 . . . 2752
4.1751 reassemble_aos_packets/5 . . . 2752
4.1752 reassemble_aos_packets/6 . . . 2752
4.1753 reassemble_aos_packets/7 . . . 2752
4.1754 reassemble_frames/6 . . . 2752
4.1755 reassemble_frames/7 . . . 2753
4.1756 reassemble_frames/8 . . . 2753
4.1757 reassemble_packet_zone/5 . . . 2753
4.1758 reassemble_packets/6 . . . 2753
4.1759 reassemble_packets/7 . . . 2753
4.1760 reassemble_packets/8 . . . 2753
4.1761 reassemble_tc_frame/4 . . . 2753
4.1762 reassemble_tc_frame/5 . . . 2753
4.1763 reassemble_tc_frame/6 . . . 2754
4.1764 reassemble_tc_frame_with_provenance/5 . . . 2754
4.1765 reassemble_tc_frame_with_provenance/6 . . . 2754
4.1766 reassemble_tc_frames/4 . . . 2754
4.1767 reassemble_tc_frames/5 . . . 2754
4.1768 reassemble_tc_frames/6 . . . 2754
4.1769 reassemble_tc_frames_with_provenance/5 . . . 2754
4.1770 reassemble_tc_frames_with_provenance/6 . . . 2754
4.1771 reassemble_tm_frames/5 . . . 2755
4.1772 reassemble_tm_frames/6 . . . 2755
4.1773 reassemble_tm_frames/7 . . . 2755
4.1774 reassemble_tm_packets/5 . . . 2755
4.1775 reassemble_tm_packets/6 . . . 2755
4.1776 reassemble_tm_packets/7 . . . 2755
4.1777 receive/3 . . . 2755
4.1778 recorda/2 . . . 2755
4.1779 recorda/3 . . . 2756
4.1780 recorded/2 . . . 2756
4.1781 recorded/3 . . . 2756
4.1782 recordz/2 . . . 2756
4.1783 recordz/3 . . . 2756

4.1784 refine/5 . . . 2756
4.1785 refine_propagate/3 . . . 2756
4.1786 refine_propagate/4 . . . 2756
4.1787 refine_propagate/5 . . . 2757
4.1788 refine_propagate/6 . . . 2757
4.1789 regressor_options/2 . . . 2757
4.1790 relation/1 . . . 2757
4.1791 relation/3 . . . 2757
4.1792 relation/4 . . . 2757
4.1793 relation_set/1 . . . 2757
4.1794 relative_standard_deviation/2 . . . 2757
4.1795 release/1 . . . 2758
4.1796 release_date/1 . . . 2758
4.1797 relevance/3 . . . 2758
4.1798 removeDependent/1 . . . 2758
4.1799 remove_duplicates/2 . . . 2758
4.1800 remove_rule/1 . . . 2758
4.1801 rename/4 . . . 2758
4.1802 rename_file/2 . . . 2758
4.1803 replace/3 . . . 2759
4.1804 replace/5 . . . 2759
4.1805 replace_sub_atom/4 . . . 2759
4.1806 report_directory/3 . . . 2759
4.1807 report_entity/3 . . . 2759
4.1808 report_library/3 . . . 2759
4.1809 report_predicate/4 . . . 2759
4.1810 repository/1 . . . 2759
4.1811 repository_branch/1 . . . 2760
4.1812 repository_commit/1 . . . 2760
4.1813 repository_commit_abbreviated/1 . . . 2760
4.1814 repository_commit_author/1 . . . 2760
4.1815 repository_commit_date/1 . . . 2760
4.1816 repository_commit_message/1 . . . 2760
4.1817 request/3 . . . 2760
4.1818 request/4 . . . 2760
4.1819 rescale/3 . . . 2761
4.1820 reset/0 . . . 2761
4.1821 reset/1 . . . 2761
4.1822 reset_counter/1 . . . 2761
4.1823 reset_counters/0 . . . 2761
4.1824 reset_flags/0 . . . 2761
4.1825 reset_flags/1 . . . 2761
4.1826 reset_genint/0 . . . 2762
4.1827 reset_genint/1 . . . 2762
4.1828 reset_gensym/0 . . . 2762
4.1829 reset_gensym/1 . . . 2762
4.1830 reset_monitor/0 . . . 2762
4.1831 reset_seed/0 . . . 2762
4.1832 residuals/2 . . . 2762
4.1833 resize/2 . . . 2762
4.1834 resource_read/3 . . . 2763
4.1835 resources/1 . . . 2763
4.1836 response/3 . . . 2763
4.1837 restore/1 . . . 2763

4.1838 restore/2 . . . 2763
4.1839 result/2 . . . 2763
4.1840 retract_fact/1 . . . 2763
4.1841 reverse/2 . . . 2763
4.1842 rewind/2 . . . 2764
4.1843 rewind/3 . . . 2764
4.1844 rhumb_bearing/3 . . . 2764
4.1845 rhumb_destination_point/4 . . . 2764
4.1846 rhumb_distance/3 . . . 2764
4.1847 rhumb_midpoint/3 . . . 2764
4.1848 rising_factorial/3 . . . 2764
4.1849 rlibraries/1 . . . 2764
4.1850 rlibraries/2 . . . 2765
4.1851 rlibrary/1 . . . 2765
4.1852 rlibrary/2 . . . 2765
4.1853 rlibrary_score/2 . . . 2765
4.1854 rol32/3 . . . 2765
4.1855 rol64/3 . . . 2765
4.1856 rollback/0 . . . 2766
4.1857 root_mean_squared_error/3 . . . 2766
4.1858 ror32/3 . . . 2766
4.1859 route_distance/2 . . . 2766
4.1860 route_distance/3 . . . 2766
4.1861 route_distance/4 . . . 2766
4.1862 rpop/3 . . . 2766
4.1863 rpush/4 . . . 2766
4.1864 rule/2 . . . 2767
4.1865 rule/3 . . . 2767
4.1866 rule/4 . . . 2767
4.1867 rules/1 . . . 2767
4.1868 run/0 . . . 2767
4.1869 run/1 . . . 2767
4.1870 run/2 . . . 2767
4.1871 run/3 . . . 2767
4.1872 run/4 . . . 2768
4.1873 run_test_sets/1 . . . 2768
4.1874 sadd/4 . . . 2768
4.1875 same_instant/2 . . . 2768
4.1876 same_length/2 . . . 2768
4.1877 same_length/3 . . . 2768
4.1878 sample_arrangements/4 . . . 2768
4.1879 sample_cartesian_tuples/3 . . . 2768
4.1880 sample_combinations/4 . . . 2769
4.1881 sample_derangements/3 . . . 2769
4.1882 sample_distinct_arrangements/4 . . . 2769
4.1883 sample_distinct_cartesian_tuples/3 . . . 2769
4.1884 sample_distinct_combinations/4 . . . 2769
4.1885 sample_distinct_derangements/3 . . . 2769
4.1886 sample_distinct_multisets/4 . . . 2769
4.1887 sample_distinct_partitions/3 . . . 2769
4.1888 sample_distinct_partitions/4 . . . 2770
4.1889 sample_distinct_permutations/3 . . . 2770
4.1890 sample_multisets/4 . . . 2770
4.1891 sample_partitions/3 . . . 2770

4.1892 sample_partitions/4 . . . 2770
4.1893 sample_permutations/3 . . . 2770
4.1894 save/0 . . . 2770
4.1895 save/1 . . . 2770
4.1896 save/2 . . . 2771
4.1897 save_dut1_entries/1 . . . 2771
4.1898 save_leap_seconds_entries/1 . . . 2771
4.1899 scalar_product/3 . . . 2771
4.1900 scale_matrix/3 . . . 2771
4.1901 scale_vector/3 . . . 2771
4.1902 scan_left/4 . . . 2771
4.1903 scan_left_1/3 . . . 2772
4.1904 scan_right/4 . . . 2772
4.1905 scan_right_1/3 . . . 2772
4.1906 scard/3 . . . 2772
4.1907 score/3 . . . 2772
4.1908 score_all/3 . . . 2772
4.1909 scores/2 . . . 2772
4.1910 search/1 . . . 2772
4.1911 secondary_header/2 . . . 2773
4.1912 secondary_header_flag/2 . . . 2773
4.1913 secondary_header_time/3 . . . 2773
4.1914 segment_data/2 . . . 2773
4.1915 segment_header/2 . . . 2773
4.1916 segment_header_suffix/2 . . . 2773
4.1917 segment_header_suffixes/2 . . . 2773
4.1918 segment_length_identifier/2 . . . 2773
4.1919 select/3 . . . 2774
4.1920 select/4 . . . 2774
4.1921 selectchk/3 . . . 2774
4.1922 selectchk/4 . . . 2774
4.1923 send/3 . . . 2774
4.1924 send/4 . . . 2774
4.1925 send_heartbeat/1 . . . 2774
4.1926 sentence_type/2 . . . 2775
4.1927 sequence/2 . . . 2775
4.1928 sequence/3 . . . 2775
4.1929 sequence/4 . . . 2775
4.1930 sequence/5 . . . 2775
4.1931 sequence_count/2 . . . 2775
4.1932 sequence_flags/2 . . . 2775
4.1933 sequence_number/2 . . . 2776
4.1934 sequential_occurrences/2 . . . 2776
4.1935 sequential_occurrences/3 . . . 2776
4.1936 serve/3 . . . 2776
4.1937 server_accept/4 . . . 2776
4.1938 server_accept/5 . . . 2776
4.1939 server_close/1 . . . 2776
4.1940 server_open/2 . . . 2776
4.1941 server_open/3 . . . 2777
4.1942 server_open/4 . . . 2777
4.1943 set/1 . . . 2777
4.1944 set/3 . . . 2777
4.1945 set/4 . . . 2777

4.1946 set/5 . . . 2777
4.1947 set_element/2 . . . 2777
4.1948 set_field/2 . . . 2777
4.1949 set_flag_value/2 . . . 2778
4.1950 set_flag_value/3 . . . 2778
4.1951 set_monitor/4 . . . 2778
4.1952 set_seed/1 . . . 2778
4.1953 set_spy_point/4 . . . 2778
4.1954 set_write_max_depth/1 . . . 2778
4.1955 setrange/5 . . . 2778
4.1956 setup/0 . . . 2778
4.1957 shell/1 . . . 2779
4.1958 shell/2 . . . 2779
4.1959 shell_command/1 . . . 2779
4.1960 shift_matrix_diagonal/3 . . . 2779
4.1961 shl64/3 . . . 2779
4.1962 short_flags/1 . . . 2779
4.1963 shr64/3 . . . 2779
4.1964 shrink/3 . . . 2779
4.1965 shrink_sequence/3 . . . 2780
4.1966 shrinker/1 . . . 2780
4.1967 shutdown_server/1 . . . 2780
4.1968 sign//1 . . . 2780
4.1969 signaling_field/2 . . . 2780
4.1970 singleton/2 . . . 2780
4.1971 singletons/2 . . . 2780
4.1972 ismember/4 . . . 2780
4.1973 size/2 . . . 2781
4.1974 skewness/2 . . . 2781
4.1975 sleep/1 . . . 2781
4.1976 sliding_window/3 . . . 2781
4.1977 smembers/3 . . . 2781
4.1978 softmax/2 . . . 2781
4.1979 softmax/3 . . . 2781
4.1980 software_heritage_identifier/1 . . . 2781
4.1981 solve_cholesky/3 . . . 2782
4.1982 solve_linear_system/3 . . . 2782
4.1983 solve_linear_systems/3 . . . 2782
4.1984 solve_lower_triangular_matrix/3 . . . 2782
4.1985 solve_upper_triangular/3 . . . 2782
4.1986 solve_upper_triangular_matrix/3 . . . 2782
4.1987 sort/2 . . . 2782
4.1988 sort/3 . . . 2782
4.1989 sort/4 . . . 2783
4.1990 soundex/2 . . . 2783
4.1991 soundex_match/2 . . . 2783
4.1992 source_file_extension/1 . . . 2783
4.1993 space//0 . . . 2783
4.1994 spacecraft_id/2 . . . 2783
4.1995 spaces//0 . . . 2783
4.1996 split/3 . . . 2783
4.1997 split/4 . . . 2784
4.1998 split_aos_packet_zone/3 . . . 2784
4.1999 split_packet_zone/4 . . . 2784

4.2000 split_string/4 . . . 2784
4.2001 spy/1 . . . 2784
4.2002 spy/3 . . . 2784
4.2003 spy/4 . . . 2784
4.2004 spy_point/4 . . . 2784
4.2005 spying/1 . . . 2785
4.2006 spying/3 . . . 2785
4.2007 spying/4 . . . 2785
4.2008 srem/4 . . . 2785
4.2009 stabilize_vector_sign/2 . . . 2785
4.2010 stabilize_vector_sign/3 . . . 2785
4.2011 standard_cauchy/3 . . . 2785
4.2012 standard_deviation/2 . . . 2785
4.2013 standard_error/2 . . . 2786
4.2014 standard_exponential/1 . . . 2786
4.2015 standard_gamma/2 . . . 2786
4.2016 standard_normal/1 . . . 2786
4.2017 standard_t/2 . . . 2786
4.2018 start/0 . . . 2786
4.2019 start/2 . . . 2786
4.2020 start/3 . . . 2786
4.2021 start/4 . . . 2787
4.2022 start/5 . . . 2787
4.2023 start_redirect_to_file/2 . . . 2787
4.2024 started_by/2 . . . 2787
4.2025 starts/2 . . . 2787
4.2026 state_energy/2 . . . 2787
4.2027 stats/1 . . . 2787
4.2028 stats/2 . . . 2787
4.2029 stats/3 . . . 2788
4.2030 stem/2 . . . 2788
4.2031 stems/2 . . . 2788
4.2032 stirling_first/3 . . . 2788
4.2033 stirling_second/3 . . . 2788
4.2034 stop/0 . . . 2788
4.2035 stop_condition/3 . . . 2788
4.2036 stop_redirect_to_file/0 . . . 2788
4.2037 strata/1 . . . 2789
4.2038 stream_to_bytes/2 . . . 2789
4.2039 stream_to_bytes/3 . . . 2789
4.2040 stream_to_chars/2 . . . 2789
4.2041 stream_to_chars/3 . . . 2789
4.2042 stream_to_codes/2 . . . 2789
4.2043 stream_to_codes/3 . . . 2789
4.2044 stream_to_terms/2 . . . 2789
4.2045 stream_to_terms/3 . . . 2790
4.2046 string_chars/2 . . . 2790
4.2047 string_codes/2 . . . 2790
4.2048 string_concat/3 . . . 2790
4.2049 string_length/2 . . . 2790
4.2050 string_lower/2 . . . 2790
4.2051 string_upper/2 . . . 2790
4.2052 strlen/3 . . . 2790
4.2053 strongest_paths/2 . . . 2791

4.2054 strongly_connected_components/2 . . . 2791
4.2055 sub_string/5 . . . 2791
4.2056 subclass/1 . . . 2791
4.2057 subclasses/1 . . . 2791
4.2058 subdivision/4 . . . 2791
4.2059 subfactorial/2 . . . 2791
4.2060 sublist/2 . . . 2791
4.2061 submilliseconds/2 . . . 2792
4.2062 subscribe/4 . . . 2792
4.2063 subsequence/2 . . . 2792
4.2064 subsequence/3 . . . 2792
4.2065 subsequence/4 . . . 2792
4.2066 subsequence_at_indices/3 . . . 2792
4.2067 subsequence_length/2 . . . 2792
4.2068 subsequences/2 . . . 2792
4.2069 subsequences/3 . . . 2793
4.2070 subsequences_with_min_span/3 . . . 2793
4.2071 subset/2 . . . 2793
4.2072 subslices/2 . . . 2793
4.2073 substitute/4 . . . 2793
4.2074 subsumes/2 . . . 2793
4.2075 subterm/2 . . . 2793
4.2076 subtract/3 . . . 2794
4.2077 subtract_duration/3 . . . 2794
4.2078 subtract_matrices/3 . . . 2794
4.2079 subtract_vectors/3 . . . 2794
4.2080 succ/2 . . . 2794
4.2081 suffix/2 . . . 2794
4.2082 suffix/3 . . . 2794
4.2083 suffix_alias/2 . . . 2795
4.2084 suffix_data/2 . . . 2795
4.2085 sum/2 . . . 2795
4.2086 sum_of_squares/2 . . . 2795
4.2087 summary/1 . . . 2795
4.2088 superclass/1 . . . 2795
4.2089 superclasses/1 . . . 2795
4.2090 supplier/1 . . . 2795
4.2091 supported_range/2 . . . 2796
4.2092 suspend_monitor/0 . . . 2796
4.2093 swap/1 . . . 2796
4.2094 swap/2 . . . 2796
4.2095 swap_consecutive/2 . . . 2796
4.2096 symdiff/3 . . . 2796
4.2097 symmetric_closure/2 . . . 2796
4.2098 symmetric_eigen/3 . . . 2796
4.2099 symmetric_eigen/4 . . . 2797
4.2100 symmetric_eigen/5 . . . 2797
4.2101 synchronization_flag/2 . . . 2797
4.2102 tab//0 . . . 2797
4.2103 tabs//0 . . . 2797
4.2104 tai_minus_utc_for_tai_unix/2 . . . 2797
4.2105 tail/2 . . . 2797
4.2106 tail1/2 . . . 2797
4.2107 tails/2 . . . 2798

4.2108 tails1/2 . . . 2798
4.2109 take/3 . . . 2798
4.2110 take/4 . . . 2798
4.2111 talker/2 . . . 2798
4.2112 target/1 . . . 2798
4.2113 tcb_minus_tdb_approx/3 . . . 2798
4.2114 tcg_minus_tt_approx/3 . . . 2798
4.2115 tdb_minus_tt_approx/3 . . . 2799
4.2116 temporal_pairwise_dataset_connected_components/2 . . . 2799
4.2117 temporal_pairwise_dataset_declared_items/2 . . . 2799
4.2118 temporal_pairwise_dataset_games/2 . . . 2799
4.2119 temporal_pairwise_dataset_games/3 . . . 2799
4.2120 temporal_pairwise_dataset_items/2 . . . 2799
4.2121 temporal_pairwise_dataset_periods/2 . . . 2799
4.2122 temporal_pairwise_dataset_summary/2 . . . 2799
4.2123 temporary_directory/1 . . . 2800
4.2124 term/2 . . . 2800
4.2125 term/4 . . . 2800
4.2126 term_expansion/2 . . . 2800
4.2127 terms_to_array/2 . . . 2800
4.2128 test/1 . . . 2800
4.2129 time_of_day/4 . . . 2800
4.2130 time_stamp/1 . . . 2800
4.2131 time_string/3 . . . 2801
4.2132 time_type/2 . . . 2801
4.2133 time_type/3 . . . 2801
4.2134 time_type/4 . . . 2801
4.2135 timeout/1 . . . 2801
4.2136 timestamp/2 . . . 2801
4.2137 timestamp/8 . . . 2801
4.2138 to_expected/1 . . . 2801
4.2139 to_expected/2 . . . 2802
4.2140 to_optional/1 . . . 2802
4.2141 today/3 . . . 2802
4.2142 tolerance_equal/4 . . . 2802
4.2143 tool/1 . . . 2802
4.2144 tool_call/3 . . . 2802
4.2145 tool_call/4 . . . 2802
4.2146 tools/0 . . . 2802
4.2147 tools/1 . . . 2803
4.2148 top/3 . . . 2803
4.2149 top_next/5 . . . 2803
4.2150 topological_sort/2 . . . 2803
4.2151 topological_sort/3 . . . 2803
4.2152 touch/3 . . . 2803
4.2153 trace/0 . . . 2803
4.2154 trace_event/2 . . . 2803
4.2155 transaction/2 . . . 2804
4.2156 transform/3 . . . 2804
4.2157 transform/4 . . . 2804
4.2158 transitive_closure/2 . . . 2804
4.2159 transitive_reduction/2 . . . 2804
4.2160 transpose/2 . . . 2804
4.2161 transpose_matrix/2 . . . 2804

4.2162 traverse/3 . . . 2804
4.2163 triangular/4 . . . 2805
4.2164 trim/2 . . . 2805
4.2165 trim/3 . . . 2805
4.2166 trim_left/2 . . . 2805
4.2167 trim_left/3 . . . 2805
4.2168 trim_right/2 . . . 2805
4.2169 trim_right/3 . . . 2805
4.2170 trimmed_mean/3 . . . 2805
4.2171 true/1 . . . 2806
4.2172 tt_minus_tai/2 . . . 2806
4.2173 ttl/3 . . . 2806
4.2174 tx_commit/1 . . . 2806
4.2175 tx_rollback/1 . . . 2806
4.2176 tx_select/1 . . . 2806
4.2177 type/1 . . . 2806
4.2178 type/2 . . . 2806
4.2179 type/3 . . . 2807
4.2180 tzdb_version/1 . . . 2807
4.2181 unexpected/1 . . . 2807
4.2182 unexpecteds/2 . . . 2807
4.2183 uniform/1 . . . 2807
4.2184 uniform/3 . . . 2807
4.2185 uninstall/0 . . . 2807
4.2186 uninstall/1 . . . 2807
4.2187 uninstall/2 . . . 2808
4.2188 union/3 . . . 2808
4.2189 union/4 . . . 2808
4.2190 union_all/3 . . . 2808
4.2191 universal/1 . . . 2808
4.2192 unix_seconds/2 . . . 2808
4.2193 unix_to_date_time/2 . . . 2808
4.2194 unpin/0 . . . 2809
4.2195 unpin/1 . . . 2809
4.2196 unsubscribe/3 . . . 2809
4.2197 unzip/2 . . . 2809
4.2198 update/0 . . . 2809
4.2199 update/1 . . . 2809
4.2200 update/2 . . . 2809
4.2201 update/3 . . . 2810
4.2202 update/4 . . . 2810
4.2203 update/5 . . . 2810
4.2204 update_fecf/2 . . . 2810
4.2205 update_in/4 . . . 2810
4.2206 update_in/5 . . . 2810
4.2207 upper_triangular_part/2 . . . 2810
4.2208 upper_triangular_part/3 . . . 2811
4.2209 user_data/2 . . . 2811
4.2210 utc_date_time_to_instant/2 . . . 2811
4.2211 utc_to_local/3 . . . 2811
4.2212 utc_to_local_tz/3 . . . 2811
4.2213 utm_crs/2 . . . 2811
4.2214 utm_zone/2 . . . 2811
4.2215 uuid_max/1 . . . 2811

4.2216 uuid_nil/1 . . . 2812
4.2217 uuid_null/1 . . . 2812
4.2218 uuid_v1/2 . . . 2812
4.2219 uuid_v3/3 . . . 2812
4.2220 uuid_v4/1 . . . 2812
4.2221 uuid_v5/3 . . . 2812
4.2222 uuid_v7/1 . . . 2812
4.2223 valid/0 . . . 2812
4.2224 valid/1 . . . 2813
4.2225 valid/2 . . . 2813
4.2226 valid/3 . . . 2813
4.2227 valid_anomaly_detector/1 . . . 2813
4.2228 valid_channel_reassembly_state/1 . . . 2813
4.2229 valid_classifier/1 . . . 2814
4.2230 valid_clusterer/1 . . . 2814
4.2231 valid_conversion/3 . . . 2814
4.2232 valid_coordinate/1 . . . 2814
4.2233 valid_coordinate/2 . . . 2814
4.2234 valid_crs/1 . . . 2814
4.2235 valid_date/3 . . . 2814
4.2236 valid_date_time/1 . . . 2814
4.2237 valid_descriptor/1 . . . 2815
4.2238 valid_dimension_reducer/1 . . . 2815
4.2239 valid_discontinuity_policy/1 . . . 2815
4.2240 valid_geohash/1 . . . 2815
4.2241 valid_instant/1 . . . 2815
4.2242 valid_map_dispatch/1 . . . 2815
4.2243 valid_option/1 . . . 2815
4.2244 valid_options/1 . . . 2815
4.2245 valid_pattern_miner/1 . . . 2816
4.2246 valid_profile/1 . . . 2816
4.2247 valid_ranker/1 . . . 2816
4.2248 valid_reassembled_segment/1 . . . 2816
4.2249 valid_reassembly_state/1 . . . 2816
4.2250 valid_regressor/1 . . . 2816
4.2251 valid_scale/1 . . . 2816
4.2252 valid_segment/1 . . . 2816
4.2253 valid_until_date/1 . . . 2817
4.2254 validate/1 . . . 2817
4.2255 validate/2 . . . 2817
4.2256 validate/3 . . . 2817
4.2257 validate_grouped_dataset/1 . . . 2817
4.2258 validate_grouped_dataset/2 . . . 2817
4.2259 validate_pairwise_dataset/1 . . . 2817
4.2260 validate_pairwise_dataset/2 . . . 2818
4.2261 validate_pairwise_measurement_dataset/1 . . . 2818
4.2262 validate_pairwise_measurement_dataset/2 . . . 2818
4.2263 validate_temporal_pairwise_dataset/1 . . . 2818
4.2264 validate_temporal_pairwise_dataset/2 . . . 2818
4.2265 valids/2 . . . 2818
4.2266 value/1 . . . 2818
4.2267 value/3 . . . 2818
4.2268 value_reference/2 . . . 2819
4.2269 values/2 . . . 2819

4.2270 variables/2 . . . 2819
4.2271 variance/2 . . . 2819
4.2272 variant/2 . . . 2819
4.2273 varnumbers/2 . . . 2819
4.2274 varnumbers/3 . . . 2819
4.2275 vector_norm/3 . . . 2819
4.2276 vector_value/3 . . . 2820
4.2277 verify_commands_availability/0 . . . 2820
4.2278 verify_fecf/1 . . . 2820
4.2279 version/1 . . . 2820
4.2280 version/2 . . . 2820
4.2281 version/6 . . . 2820
4.2282 versions/3 . . . 2821
4.2283 vertices/2 . . . 2821
4.2284 vincenty_distance/3 . . . 2821
4.2285 virtual_channel_frame_count/2 . . . 2821
4.2286 virtual_channel_id/2 . . . 2821
4.2287 void/1 . . . 2821
4.2288 void_element/1 . . . 2821
4.2289 von_mises/3 . . . 2821
4.2290 wait/2 . . . 2822
4.2291 wald/3 . . . 2822
4.2292 wall_time/1 . . . 2822
4.2293 warning/1 . . . 2822
4.2294 warnings/1 . . . 2822
4.2295 weakly_connected_components/2 . . . 2822
4.2296 week_of_year_iso/2 . . . 2822
4.2297 weekday/2 . . . 2822
4.2298 weibull/3 . . . 2823
4.2299 weight_variances/2 . . . 2823
4.2300 weighted_mean/3 . . . 2823
4.2301 welcome/0 . . . 2823
4.2302 when/2 . . . 2823
4.2303 whiledo/2 . . . 2823
4.2304 white_space//0 . . . 2823
4.2305 white_spaces//0 . . . 2823
4.2306 with_connection/1 . . . 2824
4.2307 with_output_to/2 . . . 2824
4.2308 within_distance/4 . . . 2824
4.2309 without//2 . . . 2824
4.2310 wneighbors/3 . . . 2824
4.2311 word32_hex/2 . . . 2824
4.2312 word64_hex/2 . . . 2824
4.2313 working_directory/1 . . . 2824
4.2314 write_file/3 . . . 2825
4.2315 write_framed_message/2 . . . 2825
4.2316 write_max_depth/1 . . . 2825
4.2317 write_message/2 . . . 2825
4.2318 write_stream/3 . . . 2825
4.2319 write_term_to_atom/3 . . . 2825
4.2320 write_term_to_chars/3 . . . 2825
4.2321 write_term_to_chars/4 . . . 2825
4.2322 write_term_to_codes/3 . . . 2826
4.2323 write_term_to_codes/4 . . . 2826

4.2324 write_to_atom/2 . . . 2826
4.2325 write_to_chars/2 . . . 2826
4.2326 write_to_codes/2 . . . 2826
4.2327 xor64/3 . . . 2826
4.2328 year/2 . . . 2826
4.2329 z_normalization/2 . . . 2826
4.2330 zadd/5 . . . 2827
4.2331 zcard/3 . . . 2827
4.2332 zero_or_more//0 . . . 2827
4.2333 zero_or_more//1 . . . 2827
4.2334 zero_or_more//2 . . . 2827
4.2335 zip/2 . . . 2827
4.2336 zip/3 . . . 2827
4.2337 zip_at_index/4 . . . 2827
4.2338 zone/3 . . . 2828
4.2339 zone_id_kind/2 . . . 2828
4.2340 zones/1 . . . 2828
4.2341 zones/2 . . . 2828
4.2342 zrange/5 . . . 2828
4.2343 zrank/4 . . . 2828
4.2344 zrem/4 . . . 2828
4.2345 zscore/4 . . . 2828

5 Indices and tables 2829

Index 2831

CHAPTER

ONE

LIBRARIES

To load any library (including developer tools, ports, and contributions), use the goal logtalk_load(library_name(loader)). To run the library tests, use the goal logtalk_load(library_name(tester)). To load an entity, always load the loader file of the library that includes it to ensure that all required dependencies are also loaded and that any required flags are used. The loading goal can be found in the entity documentation.

1.1 adaptive_boosting_classifier

object

1.1.1 adaptive_boosting_classifier

Adaptive Boosting classifier using C4.5 decision trees as base learners. Implements the SAMME (Stagewise Additive Modeling using a Multi-class Exponential loss function) variant, which supports multi-class classification. Builds an ensemble of weighted decision trees where each subsequent tree focuses on the examples misclassified by previous trees.

Availability:
    logtalk_load(adaptive_boosting_classifier(loader))

Author: Paulo Moura
Version: 2:0:0
Date: 2026-05-11

Compilation flags:
    static, context_switching_calls

Imports:
    public probabilistic_classifier_common
Uses:
    c45_classifier
    fast_random(Algorithm)
    format

list
pairs
type

Remarks:
(none)

Inherited public predicates:
check_classifier/1 check_option/1 check_options/1 classifier_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 predict_probabilities/3 print_classifier/1 valid_classifier/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

dataset_protocol, c45_classifier, isolation_forest_anomaly_detector, knn_classifier, naive_bayes_classifier, nearest_centroid_classifier, random_forest_classifier

1.2 agglomerative_clusterer

object

1.2.1 agglomerative_clusterer

Agglomerative clusterer for continuous datasets. Learns from a dataset object implementing the clustering_dataset_protocol protocol and returns a clusterer term that can be used for assigning new instances to clusters and exported as predicate clauses.

Availability:
logtalk_load(agglomerative_clusterer(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-07

Compilation flags:
static, context_switching_calls

Imports:
public clusterer_common
Uses:
avltree
binary_heap_min
format
list
numberlist
pairs
type

Remarks:
(none)

Inherited public predicates:
check_clusterer/1 check_option/1 check_options/1 cluster/3 clusterer_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 print_clusterer/1 valid_clusterer/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

clusterer_protocol, clustering_dataset_protocol, kcenters_clusterer, kmeans_clusterer, kmedoids_clusterer

1.3 amqp

object

1.3.1 amqp

Portable AMQP 0-9-1 (Advanced Message Queuing Protocol) client. Uses the sockets library for TCP communication.

Availability:
logtalk_load(amqp(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-05-07

Compilation flags:
static, context_switching_calls

Uses:
list
os
socket
term_io

Remarks:

- Supported backends: ECLiPSe, GNU Prolog, SICStus Prolog, SWI-Prolog, Trealla Prolog, and XVM (same as the sockets library).
- Protocol version: Implements AMQP 0-9-1 specification.
- Binary protocol: AMQP is a binary protocol with typed frame encoding.
- Channels: Supports multiple concurrent channels over a single connection.
- Exchanges and queues: Full support for declaring exchanges, queues, and bindings.
- Content: Supports message publishing and consuming with content headers.
- Transactions: Supports AMQP transactions with tx.select, tx.commit, and tx.rollback.
- Publisher confirms: Support for publisher confirms can be added.
- Heartbeat: Supports heartbeat negotiation to keep connections alive.
- Reconnection: Automatic reconnection with configurable retry attempts and delays.

Inherited public predicates:
(none)

- Public predicates
  - connect/4
  - close/1
  - close/3
  - connection_alive/1
  - channel_open/3
  - channel_close/1
  - channel_close/3
  - exchange_declare/3
  - exchange_delete/3
  - exchange_bind/4

  - exchange_unbind/4
  - queue_declare/3
  - queue_delete/3
  - queue_bind/4
  - queue_unbind/4
  - queue_purge/2
  - basic_publish/4
  - basic_consume/3
  - basic_cancel/3
  - basic_get/3
  - basic_ack/3
  - basic_nack/3
  - basic_reject/3
  - basic_qos/2
  - basic_recover/2
  - receive/3
  - tx_select/1
  - tx_commit/1
  - tx_rollback/1
  - confirm_select/1
  - send_heartbeat/1
  - message_body/2
  - message_properties/2
  - message_property/3
  - message_delivery_tag/2
  - message_exchange/2
  - message_routing_key/2
  - encode_frame/2
  - decode_frame/2
- Protected predicates
- Private predicates
  - decode_field_value/3
- Operators

Public predicates

connect/4

Connects to an AMQP 0-9-1 server and performs the protocol handshake. Returns a connection handle for subsequent operations. Supports automatic reconnection on connection failures.

Compilation flags:
    static

Template:
    connect(Host,Port,Connection,Options)
Mode and number of proofs:
    connect(+atom,+integer,--compound,+list) - one_or_error

Exceptions:
    Connection refused or network error:
        amqp_error(connection_failed)
    Server rejected connection:
        amqp_error(protocol_error(Message))
    Authentication failed:
        amqp_error(auth_failed)
    All reconnection attempts failed:
        amqp_error(reconnect_failed)

Remarks:

- Option username(Username): Username for authentication. Default is guest.
- Option password(Password): Password for authentication. Default is guest.
- Option virtual_host(VHost): Virtual host name. Default is /.
- Option heartbeat(Seconds): Heartbeat interval in seconds. Default is 60.
- Option channel_max(Max): Maximum number of channels. Default is 0 (no limit).
- Option frame_max(Max): Maximum frame size. Default is 131072.
- Option reconnect(Boolean): Enable automatic reconnection on connection failure. Default is false.
- Option reconnect_attempts(N): Maximum number of reconnection attempts. Default is 3. Only used when reconnect(true).
- Option reconnect_delay(Seconds): Delay between reconnection attempts in seconds. Default is 1. Only used when reconnect(true).

close/1

Gracefully closes the AMQP connection. Closes all channels and the connection itself.

Compilation flags:
    static

Template:
    close(Connection)
Mode and number of proofs:
    close(+compound) - one_or_error

---

close/3

Closes the AMQP connection with a specific reply code and reason.

Compilation flags:
    static

Template:
    close(Connection,ReplyCode,ReplyText)
Mode and number of proofs:
    close(+compound,+integer,+atom) - one_or_error

---

connection_alive/1

Checks if the connection is still open and valid.

Compilation flags:
    static

Template:
    connection_alive(Connection)
Mode and number of proofs:
    connection_alive(+compound) - zero_or_one

---

channel_open/3

Opens a new channel on the connection. Returns a channel handle.

Compilation flags:
    static

Template:
    channel_open(Connection,ChannelNumber,Channel)
Mode and number of proofs:
    channel_open(+compound,+integer,--compound) - one_or_error

Exceptions:
    Channel already open:
        amqp_error(channel_error(Message))

---

channel_close/1

Closes a channel.

Compilation flags:
    static

Template:
    channel_close(Channel)
Mode and number of proofs:
    channel_close(+compound) - one_or_error

---

channel_close/3

Closes a channel with a specific reply code and reason.

Compilation flags:
    static

Template:

channel_close(Channel,ReplyCode,ReplyText)
Mode and number of proofs:
channel_close(+compound,+integer,+atom) - one_or_error

---

exchange_declare/3

Declares an exchange on the server.

Compilation flags:
static

Template:
exchange_declare(Channel,Exchange,Options)
Mode and number of proofs:
exchange_declare(+compound,+atom,+list) - one_or_error

Remarks:

- Option type(Type): Exchange type: direct, fanout, topic, headers. Default is direct.
- Option durable(Boolean): Survive server restart. Default is false.
- Option auto_delete(Boolean): Delete when unused. Default is false.
- Option internal(Boolean): Internal exchange. Default is false.
- Option arguments(Arguments): Additional arguments as key-value pairs.

---

exchange_delete/3

Deletes an exchange.

Compilation flags:
static

Template:
exchange_delete(Channel,Exchange,Options)
Mode and number of proofs:
exchange_delete(+compound,+atom,+list) - one_or_error

Remarks:

- Option if_unused(Boolean): Only delete if unused. Default is false.

---

exchange_bind/4

Binds an exchange to another exchange.

Compilation flags:
    static

Template:
    exchange_bind(Channel,Destination,Source,Options)
Mode and number of proofs:
    exchange_bind(+compound,+atom,+atom,+list) - one_or_error

Remarks:

- Option routing_key(Key): Routing key for binding. Default is empty.
- Option arguments(Arguments): Additional arguments.

---

exchange_unbind/4

Unbinds an exchange from another exchange.

Compilation flags:
    static

Template:
    exchange_unbind(Channel,Destination,Source,Options)
Mode and number of proofs:
    exchange_unbind(+compound,+atom,+atom,+list) - one_or_error

---

queue_declare/3

Declares a queue on the server. If Queue is a variable, the server generates a unique name.

Compilation flags:
    static

Template:
    queue_declare(Channel,Queue,Options)
Mode and number of proofs:
    queue_declare(+compound,?atom,+list) - one_or_error

Remarks:

- Option durable(Boolean): Survive server restart. Default is false.
- Option exclusive(Boolean): Exclusive to this connection. Default is false.
- Option auto_delete(Boolean): Delete when unused. Default is false.
- Option arguments(Arguments): Additional arguments (e.g., message TTL, dead letter exchange).

---

queue_delete/3

Deletes a queue.

Compilation flags:
    static

Template:
    queue_delete(Channel,Queue,Options)
Mode and number of proofs:
    queue_delete(+compound,+atom,+list) - one_or_error

Remarks:

- Option if_unused(Boolean): Only delete if unused. Default is false.
- Option if_empty(Boolean): Only delete if empty. Default is false.

---

queue_bind/4

Binds a queue to an exchange.

Compilation flags:
    static

Template:
    queue_bind(Channel,Queue,Exchange,Options)
Mode and number of proofs:
    queue_bind(+compound,+atom,+atom,+list) - one_or_error

Remarks:

- Option routing_key(Key): Routing key for binding. Default is empty.
- Option arguments(Arguments): Additional arguments.

---

queue_unbind/4

Unbinds a queue from an exchange.

Compilation flags:
    static

Template:
    queue_unbind(Channel,Queue,Exchange,Options)
Mode and number of proofs:
    queue_unbind(+compound,+atom,+atom,+list) - one_or_error

---

queue_purge/2

Purges all messages from a queue.

Compilation flags:
    static

Template:

queue_purge(Channel,Queue)
Mode and number of proofs:
queue_purge(+compound,+atom) - one_or_error

---

basic_publish/4

Publishes a message to an exchange.

Compilation flags:
static

Template:
basic_publish(Channel,Exchange,Body,Options)
Mode and number of proofs:
basic_publish(+compound,+atom,+term,+list) - one_or_error

Remarks:

- Option routing_key(Key): Routing key for message. Default is empty.
- Option mandatory(Boolean): Return if not routable. Default is false.
- Option immediate(Boolean): Return if not deliverable. Default is false (deprecated in RabbitMQ).
- Option content_type(Type): MIME content type.
- Option content_encoding(Enc): Content encoding.
- Option correlation_id(Id): Correlation identifier.
- Option reply_to(Queue): Reply queue name.
- Option expiration(Ms): Message TTL in milliseconds.
- Option message_id(Id): Application message identifier.
- Option timestamp(Ts): Message timestamp.
- Option type(Type): Message type name.
- Option user_id(Id): Creating user ID.
- Option app_id(Id): Creating application ID.
- Option delivery_mode(Mode): 1 for non-persistent, 2 for persistent.
- Option priority(P): Message priority (0-9).
- Option headers(H): Application headers as key-value pairs.

---

basic_consume/3

Starts consuming messages from a queue.

Compilation flags:
    static

Template:
    basic_consume(Channel,Queue,Options)
Mode and number of proofs:
    basic_consume(+compound,+atom,+list) - one_or_error

Remarks:

- Option consumer_tag(Tag): Consumer identifier. Server generates if not provided.
- Option no_local(Boolean): Do not receive own messages. Default is false.
- Option no_ack(Boolean): No acknowledgment required. Default is false.
- Option exclusive(Boolean): Exclusive consumer. Default is false.
- Option arguments(Arguments): Additional arguments.

---

basic_cancel/3

Cancels a consumer.

Compilation flags:
    static

Template:
    basic_cancel(Channel,ConsumerTag,Options)
Mode and number of proofs:
    basic_cancel(+compound,+atom,+list) - one_or_error

---

basic_get/3

Synchronously gets a message from a queue.

Compilation flags:
    static

Template:
    basic_get(Channel,Queue,Options)
Mode and number of proofs:
    basic_get(+compound,+atom,+list) - one_or_error

Remarks:

- Option no_ack(Boolean): No acknowledgment required. Default is false.

---

basic_ack/3

Acknowledges a message.

Compilation flags:
    static

Template:
    basic_ack(Channel,DeliveryTag,Options)
Mode and number of proofs:
    basic_ack(+compound,+integer,+list) - one_or_error

Remarks:

- Option multiple(Boolean): Acknowledge all up to this tag. Default is false.

---

basic_nack/3

Negatively acknowledges a message (RabbitMQ extension).

Compilation flags:
    static

Template:
    basic_nack(Channel,DeliveryTag,Options)
Mode and number of proofs:
    basic_nack(+compound,+integer,+list) - one_or_error

Remarks:

- Option multiple(Boolean): Reject all up to this tag. Default is false.
- Option requeue(Boolean): Requeue the message. Default is true.

---

basic_reject/3

Rejects a message.

Compilation flags:
    static

Template:
    basic_reject(Channel,DeliveryTag,Options)
Mode and number of proofs:
    basic_reject(+compound,+integer,+list) - one_or_error

Remarks:

- Option requeue(Boolean): Requeue the message. Default is true.

---

basic_qos/2

Sets quality of service (prefetch) settings.

Compilation flags:
    static

Template:
    basic_qos(Channel,Options)
Mode and number of proofs:
    basic_qos(+compound,+list) - one_or_error

Remarks:

- Option prefetch_size(Size): Prefetch window size in bytes. Default is 0 (no limit).
- Option prefetch_count(Count): Prefetch window in messages. Default is 0 (no limit).
- Option global(Boolean): Apply to entire connection. Default is false.

---

basic_recover/2

Asks the server to redeliver unacknowledged messages.

Compilation flags:
    static

Template:
    basic_recover(Channel,Options)
Mode and number of proofs:
    basic_recover(+compound,+list) - one_or_error

Remarks:

- Option requeue(Boolean): Requeue messages. Default is false.

---

receive/3

Receives a message or method from the server. Blocks until data is available or timeout.

Compilation flags:
    static

Template:
    receive(Channel,Message,Options)
Mode and number of proofs:
    receive(+compound,-compound,+list) - zero_or_one_or_error

Remarks:

- Option timeout(Milliseconds): Timeout in milliseconds. 0 for non-blocking, -1 for infinite. Default is -1.

---

tx_select/1

Enables transaction mode on a channel.

Compilation flags:
    static

Template:
    tx_select(Channel)
Mode and number of proofs:
    tx_select(+compound) - one_or_error

---

tx_commit/1

Commits the current transaction.

Compilation flags:
    static

Template:
    tx_commit(Channel)

Mode and number of proofs:
tx_commit(+compound) - one_or_error

---

tx_rollback/1

Rolls back the current transaction.

Compilation flags:
static

Template:
tx_rollback(Channel)
Mode and number of proofs:
tx_rollback(+compound) - one_or_error

---

confirm_select/1

Enables publisher confirms on a channel (RabbitMQ extension).

Compilation flags:
static

Template:
confirm_select(Channel)
Mode and number of proofs:
confirm_select(+compound) - one_or_error

---

send_heartbeat/1

Sends a heartbeat frame to the server.

Compilation flags:
static

Template:
send_heartbeat(Connection)
Mode and number of proofs:
send_heartbeat(+compound) - one_or_error

---

message_body/2

Extracts the body from a message.

Compilation flags:
static

Template:
message_body(Message,Body)
Mode and number of proofs:
message_body(+compound,-term) - one

---

message_properties/2

Extracts the properties from a message as a list.

Compilation flags:
static

Template:
message_properties(Message,Properties)
Mode and number of proofs:
message_properties(+compound,-list) - one

---

message_property/3

Extracts a specific property from a message. Fails if not present.

Compilation flags:
    static

Template:
    message_property(Message,PropertyName,Value)
Mode and number of proofs:
    message_property(+compound,+atom,-term) - zero_or_one

---

message_delivery_tag/2

Extracts the delivery tag from a message.

Compilation flags:
    static

Template:
    message_delivery_tag(Message,DeliveryTag)
Mode and number of proofs:
    message_delivery_tag(+compound,-integer) - one

---

message_exchange/2

Extracts the exchange name from a message.

Compilation flags:
    static

Template:
    message_exchange(Message,Exchange)
Mode and number of proofs:
    message_exchange(+compound,-atom) - one

---

message_routing_key/2

Extracts the routing key from a message.

Compilation flags:
    static

Template:
    message_routing_key(Message,RoutingKey)
Mode and number of proofs:
    message_routing_key(+compound,-atom) - one

---

encode_frame/2

Encodes an AMQP frame to a list of bytes.

Compilation flags:
    static

Template:
    encode_frame(Frame,Bytes)
Mode and number of proofs:
    encode_frame(+compound,-list) - one

---

decode_frame/2

Decodes a list of bytes to an AMQP frame.

Compilation flags:
    static

Template:
    decode_frame(Bytes,Frame)
Mode and number of proofs:
    decode_frame(+list,-compound) - one_or_error

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

decode_field_value/3

Compilation flags:
static

Operators

(none)

category

1.3.2 amqp_pool

AMQP connection pool category. Import this category into an object to create a named connection pool with automatic connection management.

Availability:
logtalk_load(amqp(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-19

Compilation flags:
static

Uses:
amqp
list
os

Remarks:
(none)

Inherited public predicates:

(none)

- Public predicates
  - initialize/1
  - destroy/0
  - acquire/1
  - release/1
  - with_connection/1
  - stats/1
  - resize/2
- Protected predicates
- Private predicates
  - pool_config/5
  - available/1
  - in_use/2
- Operators

Public predicates

initialize/1

Initializes the connection pool with the given configuration options. Must be called before using other pool predicates.

Compilation flags:
static

Template:
initialize(Options)
Mode and number of proofs:
initialize(+list) - one_or_error

Remarks:

- Option host(Host): AMQP server hostname. Default is localhost.
- Option port(Port): AMQP server port. Default is 5672.
- Option min_size(N): Minimum number of connections to maintain. Default is 1.

- Option max_size(N): Maximum number of connections allowed. Default is 10.
- Option connection_options(Options): Options passed to amqp::connect/4. Default is [].

---

destroy/0

Destroys the pool, closing all connections and clearing state.

Compilation flags:
    static

Mode and number of proofs:
    destroy - one

---

acquire/1

Acquires a connection from the pool. Returns an available connection or creates a new one if the pool is not at maximum capacity.

Compilation flags:
    static

Template:
    acquire(Connection)
Mode and number of proofs:
    acquire(--compound) - one_or_error

Exceptions:
    Pool not initialized:
        pool_error(not_initialized)
    Pool exhausted (at max capacity):
        pool_error(exhausted)

---

release/1

Releases a connection back to the pool, making it available for reuse.

Compilation flags:
    static

Template:
    release(Connection)
Mode and number of proofs:
    release(+compound) - one

---

with_connection/1

Acquires a connection, calls Goal with the connection as argument, and releases the connection. The connection is released even if Goal fails or throws an exception.

Compilation flags:
    static

Template:
    with_connection(Goal)
Meta-predicate template:
    with_connection(1)
Mode and number of proofs:
    with_connection(+callable) - zero_or_more

---

stats/1

Returns pool statistics as a compound term stats(Available, InUse, Total, MinSize, MaxSize).

Compilation flags:
    static

Template:
    stats(Stats)
Mode and number of proofs:

stats(-compound) - one

---

resize/2

Resizes the pool by setting new minimum and maximum sizes.

Compilation flags:
static

Template:
resize(MinSize,MaxSize)
Mode and number of proofs:
resize(+integer,+integer) - one_or_error

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

pool_config/5

Stores the pool configuration parameters.

Compilation flags:
dynamic

Template:
pool_config(Host,Port,MinSize,MaxSize,ConnectionOptions)
Mode and number of proofs:
pool_config(?atom,?integer,?integer,?integer,?list) - zero_or_one

---

available/1

Tracks connections that are available for use.

Compilation flags:
    dynamic

Template:
    available(Connection)
Mode and number of proofs:
    available(?compound) - zero_or_more

---

in_use/2

Tracks connections currently in use along with their acquisition timestamp.

Compilation flags:
    dynamic

Template:
    in_use(Connection,AcquireTimestamp)
Mode and number of proofs:
    in_use(?compound,?compound) - zero_or_more

Operators

(none)

1.4 anomaly_detection_protocols

protocol

1.4.1 anomaly_dataset_protocol

Protocol for datasets used with anomaly-detection algorithms.

Availability:
logtalk_load(anomaly_detection_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-19

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - attribute_values/2
  - class/1
  - class_values/1
  - example/3
- Protected predicates
- Private predicates
- Operators

Public predicates

attribute_values/2

Enumerates by backtracking the attributes and their possible values. For discrete attributes, Values is a list of possible values. For continuous attributes, Values is the atom continuous.

Compilation flags:
    static

Template:
    attribute_values(Attribute,Values)
Mode and number of proofs:
    attribute_values(?atom,-list(atom)) - zero_or_more
    attribute_values(?atom,-atom) - zero_or_more

---

class/1

Returns the name of the target label attribute.

Compilation flags:
    static

Template:
    class(Class)
Mode and number of proofs:
    class(-atom) - one

---

class_values/1

Returns the list of possible labels for the target attribute. Current anomaly benchmarks conventionally use [normal, anomaly].

Compilation flags:
    static

Template:

class_values(Values)
Mode and number of proofs:
class_values(-list(atom)) - one

---

example/3

Enumerates by backtracking the examples in the dataset. Each example has an Id, a Class label, and a list of Attribute-Value pairs.

Compilation flags:
static

Template:
example(Id,Class,AttributeValues)
Mode and number of proofs:
example(-integer,-atom,-list(pair)) - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

category

1.4.2 anomaly_detector_common

Shared predicates for anomaly detector learning defaults, threshold-based prediction, baseline training-data selection, dataset helpers, and export.

Availability:
logtalk_load(anomaly_detection_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-01

Compilation flags:
static

Implements:
public anomaly_detector_protocol
Extends:
public options
Uses:
format
list
type

Remarks:
(none)

Inherited public predicates:
anomaly_detector_options/2 check_anomaly_detector/1 check_option/1 check_options/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 predict/4 print_anomaly_detector/1 score/3 score_all/3 valid_anomaly_detector/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
  - anomaly_detector_diagnostics_data/2
  - anomaly_detector_export_template/2
  - anomaly_detector_term_template/2
  - dataset_attributes/2
  - check_examples_non_empty/2
  - baseline_training_examples/3
  - valid_baseline_class_values/1
  - extract_scores/2
  - print_anomaly_detector_template/1
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

anomaly_detector_diagnostics_data/2

Hook predicate that importing anomaly detector implementations may override in order to expose diagnostics and metadata for learned detectors.

Compilation flags:
    static

Template:
    anomaly_detector_diagnostics_data(AnomalyDetector,Diagnostics)
Mode and number of proofs:
    anomaly_detector_diagnostics_data(+compound,-list(compound)) - one

---

anomaly_detector_export_template/2

Hook predicate that importing anomaly detector implementations must define in order to expose the exported detector predicate template for a given functor.

Compilation flags:
    static

Template:
    anomaly_detector_export_template(Functor,Template)
Mode and number of proofs:
    anomaly_detector_export_template(+atom,-callable) - one

---

anomaly_detector_term_template/2

Hook predicate that importing anomaly detector implementations must define in order to expose the learned detector term template used by pretty-printing helpers.

Compilation flags:
    static

Template:
    anomaly_detector_term_template(AnomalyDetector,Template)
Mode and number of proofs:
    anomaly_detector_term_template(+compound,-compound) - one

---

dataset_attributes/2

Collects the dataset attribute declarations as Attribute-Values pairs.

Compilation flags:
    static

Template:
    dataset_attributes(Dataset,Attributes)
Mode and number of proofs:
    dataset_attributes(+object_identifier,-list(pair)) - one

---

check_examples_non_empty/2

Checks that a dataset contains at least one example.

Compilation flags:
    static

Template:
    check_examples_non_empty(Dataset,Examples)
Mode and number of proofs:
    check_examples_non_empty(+object_identifier,+list) - one

---

baseline_training_examples/3

Collects training examples selected by the baseline_class_values/1 and baseline_selection_policy/1 options. The reject policy throws an error when non-baseline examples are present. The filter policy removes non-baseline examples and throws an error when no baseline examples remain.

Compilation flags:
    static

Template:
    baseline_training_examples(Dataset,Examples,Options)
Mode and number of proofs:
    baseline_training_examples(+object_identifier,-list,+list(compound)) - one

---

valid_baseline_class_values/1

Checks that a baseline_class_values/1 option value is a non-empty list of distinct atoms.

Compilation flags:
    static

Template:
    valid_baseline_class_values(BaselineClassValues)
Mode and number of proofs:
    valid_baseline_class_values(+list(atom)) - one

---

extract_scores/2

Transforms Score-Id-Class tuples into the public Id-Class-Score representation.

Compilation flags:
    static

Template:

extract_scores(Pairs,Scores)
Mode and number of proofs:
extract_scores(+list,-list) - one

---

print_anomaly_detector_template/1

Prints the learned anomaly detector term template in a human-readable form.

Compilation flags:
static

Template:
print_anomaly_detector_template(AnomalyDetector)
Mode and number of proofs:
print_anomaly_detector_template(+compound) - one

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.4.3 anomaly_detector_protocol

Protocol for machine learning anomaly detectors.

Availability:
logtalk_load(anomaly_detection_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:

static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - learn/2
  - learn/3
  - check_anomaly_detector/1
  - valid_anomaly_detector/1
  - diagnostics/2
  - diagnostic/2
  - anomaly_detector_options/2
  - predict/3
  - predict/4
  - score/3
  - score_all/3
  - export_to_clauses/4
  - export_to_file/4
  - print_anomaly_detector/1
- Protected predicates
- Private predicates
- Operators

Public predicates

learn/2

Learns an anomaly detector from the given dataset object.

Compilation flags:
    static

Template:
    learn(Dataset,AnomalyDetector)
Mode and number of proofs:
    learn(+object_identifier,-compound) - one

---

learn/3

Learns an anomaly detector from the given dataset object using the specified options.

Compilation flags:
    static

Template:
    learn(Dataset,AnomalyDetector,Options)
Mode and number of proofs:
    learn(+object_identifier,-compound,+list(compound)) - one

---

check_anomaly_detector/1

Checks that a learned anomaly detector term is structurally valid for the receiving implementation. Throws an exception when the term is not a valid anomaly detector representation.

Compilation flags:
    static

Template:
    check_anomaly_detector(AnomalyDetector)
Mode and number of proofs:

check_anomaly_detector(@compound) - one_or_error

Exceptions:
AnomalyDetector is a variable:
instantiation_error
AnomalyDetector is neither a variable nor a valid anomaly detector:
domain_error(anomaly_detector,AnomalyDetector)

---

valid_anomaly_detector/1

True when a learned anomaly detector term is structurally valid for the receiving implementation. Succeeds iff check_anomaly_detector/1 succeeds without throwing an exception.

Compilation flags:
static

Template:
valid_anomaly_detector(AnomalyDetector)
Mode and number of proofs:
valid_anomaly_detector(@compound) - zero_or_one

---

diagnostics/2

Returns diagnostics and metadata associated with a learned anomaly detector in a representation-independent way.

Compilation flags:
static

Template:
diagnostics(AnomalyDetector,Diagnostics)
Mode and number of proofs:
diagnostics(+compound,-list(compound)) - one

---

diagnostic/2

Tests or enumerates individual diagnostics metadata terms for a learned anomaly detector.

Compilation flags:
    static

Template:
    diagnostic(AnomalyDetector,Diagnostic)
Mode and number of proofs:
    diagnostic(+compound,?compound) - zero_or_more

---

anomaly_detector_options/2

Returns the effective training options recorded in a learned anomaly detector.

Compilation flags:
    static

Template:
    anomaly_detector_options(AnomalyDetector,Options)
Mode and number of proofs:
    anomaly_detector_options(+compound,-list(compound)) - one

---

predict/3

Predicts whether a new instance is normal or anomaly using the learned detector. The instance is a list of Attribute-Value pairs.

Compilation flags:
    static

Template:
    predict(AnomalyDetector,Instance,Prediction)
Mode and number of proofs:
    predict(+compound,+list,-atom) - one

---

predict/4

Predicts whether a new instance is normal or anomaly using the learned detector and the specified options. The instance is a list of Attribute-Value pairs.

Compilation flags:
    static

Template:
    predict(AnomalyDetector,Instance,Prediction,Options)
Mode and number of proofs:
    predict(+compound,+list,-atom,+list(compound)) - one

---

score/3

Computes an anomaly score in the interval [0.0, 1.0] for a new instance. Larger scores indicate more anomalous instances.

Compilation flags:
    static

Template:
    score(AnomalyDetector,Instance,Score)
Mode and number of proofs:
    score(+compound,+list,-float) - one

---

score_all/3

Computes anomaly scores for all examples in a dataset and returns a list of Id-Class-Score triples sorted by descending score.

Compilation flags:
    static

Template:

score_all(Dataset,AnomalyDetector,Scores)
Mode and number of proofs:
score_all(+object_identifier,+compound,-list) - one

---

export_to_clauses/4

Converts an anomaly detector into a list of predicate clauses. Functor is the functor for the generated predicate clauses. When exporting a serialized detector term, a noun such as detector or model is usually clearer than a verb such as detect.

Compilation flags:
static

Template:
export_to_clauses(Dataset,AnomalyDetector,Functor,Clauses)
Mode and number of proofs:
export_to_clauses(+object_identifier,+compound,+callable,-list(clause)) - one

---

export_to_file/4

Exports an anomaly detector to a file. Functor is the functor for the generated predicate clauses. When exporting a serialized detector term, a noun such as detector or model is usually clearer than a verb such as detect.

Compilation flags:
static

Template:
export_to_file(Dataset,AnomalyDetector,Functor,File)
Mode and number of proofs:
export_to_file(+object_identifier,+compound,+callable,+atom) - one

---

print_anomaly_detector/1

Prints an anomaly detector to the current output stream in a human-readable format.

Compilation flags:
    static

Template:
    print_anomaly_detector(AnomalyDetector)
Mode and number of proofs:
    print_anomaly_detector(+compound) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

anomaly_dataset_protocol, anomaly_detector_common, isolation_forest_anomaly_detector, knn_distance_anomaly_detector, lof_anomaly_detector

1.5 application

category

1.5.1 application_common

Application metadata and provenance predicates.

Availability:
logtalk_load(application(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-03-25

Compilation flags:
static

Implements:
public application_protocol
Uses:
os

Remarks:
(none)

Inherited public predicates:
built_date/1 creators/1 description/1 distribution/1 external_reference/2 git_object_identifier/1 homepage/1 license/1 loader_file/1 name/1 originator/1 package/1 property/1 release_date/1 repository/1 repository_branch/1 repository_commit/1 repository_commit_abbreviated/1 repository_commit_author/1 repository_commit_date/1 repository_commit_message/1 software_heritage_identifier/1 supplier/1 valid_until_date/1 version/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.5.2 application_protocol

Application metadata predicates, including optional source provenance facts.

Availability:
logtalk_load(application(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-03-25

Compilation flags:
static

Dependencies:
(none)

Remarks:

- Explicit metadata: All predicates describe metadata declared by the application. The protocol does not require or imply reflection over the current and transient state of some application repository.
- Release metadata: Predicates such as name/1, version/1, description/1, license/1, distribution/1, package/1, release_date/1, and valid_until_date/1 are intended for release-oriented metadata.

- Source provenance metadata: Predicates such as repository/1, repository_branch/1, repository_commit/1, repository_commit_date/1, repository_commit_author/1, repository_commit_message/1, git_object_identifier/1, and software_heritage_identifier/1 are optional explicit provenance facts about the source used for the application and may or may not correspond to the release artifact identity.

Inherited public predicates:
(none)

- Public predicates
  - name/1
  - version/1
  - description/1
  - license/1
  - homepage/1
  - distribution/1
  - package/1
  - loader_file/1
  - creators/1
  - supplier/1
  - originator/1
  - built_date/1
  - release_date/1
  - valid_until_date/1
  - external_reference/2
  - repository/1
  - repository_branch/1
  - repository_commit/1
  - repository_commit_abbreviated/1
  - repository_commit_date/1
  - repository_commit_author/1
  - repository_commit_message/1
  - git_object_identifier/1
  - software_heritage_identifier/1
  - property/1
- Protected predicates

- Private predicates
- Operators

Public predicates

name/1

Application name.

Compilation flags:
    static

Template:
    name(Name)
Mode and number of proofs:
    name(?atom) - zero_or_one

---

version/1

Application version.

Compilation flags:
    static

Template:
    version(Version)
Mode and number of proofs:
    version(?atom) - zero_or_one

---

description/1

Application short description.

Compilation flags:
    static

Template:
    description(Description)
Mode and number of proofs:
    description(?atom) - zero_or_one

---

license/1

Application license.

Compilation flags:
    static

Template:
    license(License)
Mode and number of proofs:
    license(?atom) - zero_or_one

---

homepage/1

Application homepage URL.

Compilation flags:
    static

Template:
    homepage(URL)
Mode and number of proofs:
    homepage(?atom) - zero_or_one

---

distribution/1

Application distribution or download location.

Compilation flags:
    static

Template:
    distribution(URL)
Mode and number of proofs:
    distribution(?atom) - zero_or_one

---

package/1

Application package identifier as a PURL.

Compilation flags:
    static

Template:
    package(PURL)
Mode and number of proofs:
    package(?atom) - zero_or_one

---

loader_file/1

Application main loader file absolute path.

Compilation flags:
    static

Template:
    loader_file(File)
Mode and number of proofs:
    loader_file(?atom) - zero_or_one

---

creators/1

Application creators, authors, or other credited producers of the application or its release metadata.

Compilation flags:
    static

Template:
    creators(Creators)
Mode and number of proofs:
    creators(?list(atom)) - zero_or_one

---

supplier/1

Application supplier.

Compilation flags:
    static

Template:
    supplier(Supplier)
Mode and number of proofs:
    supplier(?atom) - zero_or_one

---

originator/1

Original source of the application software when distinct from its creators.

Compilation flags:
    static

Template:
    originator(Originator)
Mode and number of proofs:
    originator(?atom) - zero_or_one

---

built_date/1

Application build date.

Compilation flags:
    static

Template:
    built_date(Date)
Mode and number of proofs:
    built_date(?atom) - zero_or_one

---

release_date/1

Application release date.

Compilation flags:
    static

Template:
    release_date(Date)
Mode and number of proofs:
    release_date(?atom) - zero_or_one

---

valid_until_date/1

Application validity limit date.

Compilation flags:
    static

Template:
    valid_until_date(Date)
Mode and number of proofs:
    valid_until_date(?atom) - zero_or_one

---

external_reference/2

Application explicit external references using the same vocabulary as the corresponding first-class metadata predicates. The second argument can be a URL or a non-URL identifier depending on the reference type.

Compilation flags:
    static

Template:
    external_reference(Type,Locator)
Mode and number of proofs:
    external_reference(?atom,?atom) - zero_or_more

---

repository/1

Application source provenance repository metadata.

Compilation flags:
    static

Template:
    repository(URL)
Mode and number of proofs:
    repository(?atom) - zero_or_one

---

repository_branch/1

Application source provenance git branch metadata.

Compilation flags:
    static

Template:
    repository_branch(Branch)
Mode and number of proofs:
    repository_branch(?atom) - zero_or_one

---

repository_commit/1

Application source provenance git commit metadata.

Compilation flags:
    static

Template:
    repository_commit(Hash)
Mode and number of proofs:
    repository_commit(?atom) - zero_or_one

---

repository_commit_abbreviated/1

Application abbreviated source provenance git commit metadata.

Compilation flags:
    static

Template:
    repository_commit_abbreviated(Hash)
Mode and number of proofs:
    repository_commit_abbreviated(?atom) - zero_or_one

---

repository_commit_date/1

Application source provenance git commit date metadata.

Compilation flags:
    static

Template:
    repository_commit_date(Date)
Mode and number of proofs:

repository_commit_date(?atom) - zero_or_one

---

repository_commit_author/1

Application source provenance git commit author metadata.

Compilation flags:
static

Template:
repository_commit_author(Author)
Mode and number of proofs:
repository_commit_author(?atom) - zero_or_one

---

repository_commit_message/1

Application source provenance git commit message metadata.

Compilation flags:
static

Template:
repository_commit_message(Message)
Mode and number of proofs:
repository_commit_message(?atom) - zero_or_one

---

git_object_identifier/1

Application source provenance git object identifier metadata as a gitoid.

Compilation flags:
static

Template:

git_object_identifier(GITOID)
Mode and number of proofs:
git_object_identifier(?atom) - zero_or_one

---

software_heritage_identifier/1

Application source provenance Software Heritage identifier metadata as an SWHID.

Compilation flags:
static

Template:
software_heritage_identifier(SWHID)
Mode and number of proofs:
software_heritage_identifier(?atom) - zero_or_one

---

property/1

Enumerates declared application metadata and optional source provenance as individual property terms.

Compilation flags:
static

Template:
property(Property)
Mode and number of proofs:
property(?compound) - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.6 apriori_pattern_miner

object

1.6.1 apriori_pattern_miner

Apriori frequent itemset miner for transaction datasets using deterministic level-wise candidate generation and anti-monotone pruning.

Availability:
logtalk_load(apriori_pattern_miner(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public frequent_pattern_mining_common
Uses:
avltree
combinations
fnv1a_32
format
list

Remarks:
(none)

Inherited public predicates:

check_option/1 check_options/1 check_pattern_miner/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 mine/2 mine/3 option/2 option/3 pattern_miner_options/2 print_pattern_miner/1 valid_option/1 valid_options/1 valid_pattern_miner/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

pattern_miner_protocol, transaction_dataset_protocol, combinations

1.7 arbitrary

category

1.7.1 arbitrary

Adds predicates for generating and shrinking random values for selected types to the library type object. User extensible.

Availability:
logtalk_load(arbitrary(loader))

Author: Paulo Moura
Version: 2:38:0
Date: 2026-02-28

Compilation flags:
static

Complements:
type
Uses:
fast_random
integer
list
type

Remarks:

- Logtalk specific types: entity, object, protocol, category, entity_identifier, object_identifier, protocol_identifier, category_identifier, event, predicate.
- Prolog module related types (when the backend compiler supports modules): module, module_identifier, qualified_callable.
- Prolog base types: term, var, nonvar, atomic, atom, number, integer, float, compound, callable, ground.
- Atom derived types: non_quoted_atom, non_empty_atom, non_empty_atom(CharSet), boolean, character, in_character, char, operator_specifier, hex_char.
- Atom derived parametric types: atom(CharSet), atom(CharSet,Length), non_empty_atom(CharSet), character(CharSet), in_character(CharSet), char(CharSet).
- Number derived types: positive_number, negative_number, non_positive_number, non_negative_number.
- Float derived types: positive_float, negative_float, non_positive_float, non_negative_float, probability.
- Integer derived types: positive_integer, negative_integer, non_positive_integer, non_negative_integer, byte, in_byte, character_code, in_character_code, code, operator_priority, hex_code.
- Integer derived parametric types: character_code(CharSet), in_character_code(CharSet), code(CharSet).

- List types (compound derived types): list, non_empty_list, partial_list, list_or_partial_list, list(Type), list(Type,Length), list(Type,Min,Max), list(Type,Length,Min,Max), non_empty_list(Type), codes, chars.
- Difference list types (compound derived types): difference_list, difference_list(Type).
- List and difference list types length: The types that do not take a fixed length generate lists with a length in the [0,MaxSize] interval ([1,MaxSize] for non-empty list types).
- Predicate and non-terminal indicator types arity: These types generate indicators with an arity in the [0,MaxSize] interval.
- Other compound derived types: compound(Name,Types), predicate_indicator, non_terminal_indicator, predicate_or_non_terminal_indicator, clause, grammar_rule, pair, pair(KeyType,ValueType).
- Other types: text, text(CharSet), Object::Closure, between(Type,Lower,Upper), property(Type, LambdaExpression), one_of(Type,Set), var_or(Type), ground(Type), types(Types), types_frequency(Pairs), transform(Type,Closure), constrain(Type,Closure).
- Types text and text(CharSet) notes: Generate random text represented using either atoms, character lists, or character code lists.
- Type Object::Closure notes: Allows calling public object predicates as generators and shrinkers. The Closure closure is extended with either a single argument, the generated arbitrary value, or with two arguments, when shrinking a value.
- Type compound(Name,Types) notes: Generate a random compound term with the given name with a random argument for each type.
- Type types_frequency(Pairs) notes: Generate a random term for one of the types in a list of Type-Frequency pairs. The type is randomly selected taking into account the types frequency.
- Type transform(Type,Closure) notes: Generate a random term by transforming the term generated for the given type using the given closure.
- Type constrain(Type,Closure) notes: Generate a random term for the given type that satisfy the given closure.
- Registering new types: Add clauses for the arbitrary/1-2 multifile predicates and optionally for the shrinker/1 and shrink/3 multifile predicates. The clauses must have a bound first argument to avoid introducing spurious choice-points.
- Shrinking values: The shrink/3 should either succeed or fail but never throw an exception.
- Character sets: ascii_identifier, ascii_printable, ascii_full, hexadecimal, byte, unicode_bmp, unicode_full.
- Default character sets: The default character set when using a parameterizable type that takes a character set parameter depends on the type.
- Default character sets: Entity, predicate, and non-terminal identifier types plus compound and callable types default to an ascii_identifier functor. Character and character code types default to ascii_full. Other types default to ascii_printable.
- Caveats: The type argument (and any type parameterization) to the predicates is not type-checked (or checked for consistency) for performance reasons.
- Unicode limitations: Currently, correct character/code generation is only ensured for SWI-Prolog and XVM as other backends do not provide support for querying a Unicode code point category.

Inherited public predicates:

(none)

- Public predicates
  - arbitrary/1
  - arbitrary/2
  - shrinker/1
  - shrink/3
  - shrink_sequence/3
  - edge_case/2
  - get_seed/1
  - set_seed/1
  - max_size/1
- Protected predicates
- Private predicates
- Operators

Public predicates

arbitrary/1

Table of defined types for which an arbitrary value can be generated. A new type can be registered by defining a clause for this predicate and adding a clause for the arbitrary/2 multifile predicate.

Compilation flags:
static, multifile

Template:
arbitrary(Type)
Mode and number of proofs:
arbitrary(?callable) - zero_or_more

arbitrary/2

Generates an arbitrary term of the specified type. Fails if the type is not supported. A new generator can be defined by adding a clause for this predicate and registering it via the arbitrary/1 predicate.

Compilation flags:
    static, multifile

Template:
    arbitrary(Type,Term)
Meta-predicate template:
    arbitrary(::,*)
Mode and number of proofs:
    arbitrary(@callable,-term) - zero_or_one

---

shrinker/1

Table of defined types for which a shrinker is provided. A new shrinker can be registered by defining a clause for this predicate and adding a definition for the shrink/3 multifile predicate.

Compilation flags:
    static, multifile

Template:
    shrinker(Type)
Mode and number of proofs:
    shrinker(?callable) - zero_or_more

---

shrink/3

Shrinks a value to a smaller value if possible. Must generate a finite number of solutions. Fails if the type is not supported. A new shrinker can be defined by adding a clause for this predicate and registering it via the shrinker/1 predicate.

Compilation flags:
    static, multifile

Template:

shrink(Type,Large,Small)
Mode and number of proofs:
shrink(@callable,@term,-term) - zero_or_more

---

shrink_sequence/3

Shrinks a value repeatedly until shrinking is no longer possible returning the sequence of values (ordered from larger to smaller value). Fails if the type is not supported.

Compilation flags:
static

Template:
shrink_sequence(Type,Value,Sequence)
Mode and number of proofs:
shrink_sequence(@callable,@term,-list(term)) - zero_or_one

---

edge_case/2

Table of type edge cases. Fails if the given type have no defined edge cases. New edge cases for existing or new types can be added by defining a clause for this multifile predicate.

Compilation flags:
static, multifile

Template:
edge_case(Type,Term)
Mode and number of proofs:
edge_case(?callable,?term) - zero_or_more

---

get_seed/1

Gets the current random generator seed. Seed should be regarded as an opaque ground term.

Compilation flags:
    static

Template:
    get_seed(Seed)
Mode and number of proofs:
    get_seed(-ground) - one

---

set_seed/1

Sets the random generator seed to a given value returned by calling the get_seed/1 predicate.

Compilation flags:
    static

Template:
    set_seed(Seed)
Mode and number of proofs:
    set_seed(+ground) - one

---

max_size/1

User defined maximum size for types where its meaningful and implicit. When not defined, defaults to 42. When multiple definitions exist, the first valid one found is used.

Compilation flags:
    static, multifile

Template:
    max_size(Size)
Mode and number of proofs:
    max_size(?positive_integer) - zero_or_one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

type

1.8 arrangements

object

1.8.1 arrangements

Implementation of arrangements operations with repetition over lists.

Availability:
logtalk_load(arrangements(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-12

Compilation flags:
static, context_switching_calls

Implements:
public arrangements_protocol
Uses:
fast_random(Algorithm)
list

Remarks:
(none)

Inherited public predicates:
arrangement/3 arrangement/4 arrangement_index/4 arrangement_index/5 arrangements/3 arrangements/4 count_arrangements/3 count_distinct_arrangements/3 distinct_arrangement/3 distinct_arrangement/4 distinct_arrangement_index/4 distinct_arrangements/3 distinct_arrangements/4 next_arrangement/3 nth_arrangement/4 nth_arrangement/5 nth_distinct_arrangement/4 previous_arrangement/3 random_arrangement/3 random_distinct_arrangement/3 sample_arrangements/4 sample_distinct_arrangements/4

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.8.2 arrangements_protocol

Protocol for arrangements operations over lists.

Availability:
logtalk_load(arrangements(loader))

Author: Paulo Moura

Version: 1:0:0
Date: 2026-05-12

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - arrangements/3
  - arrangement/3
  - arrangements/4
  - arrangement/4
  - distinct_arrangements/3
  - distinct_arrangement/3
  - distinct_arrangements/4
  - distinct_arrangement/4
  - nth_arrangement/4
  - nth_arrangement/5
  - arrangement_index/4
  - arrangement_index/5
  - count_arrangements/3
  - count_distinct_arrangements/3
  - nth_distinct_arrangement/4
  - distinct_arrangement_index/4
  - random_arrangement/3
  - sample_arrangements/4
  - random_distinct_arrangement/3
  - sample_distinct_arrangements/4

  - next_arrangement/3
  - previous_arrangement/3
- Protected predicates
- Private predicates
- Operators

Public predicates

arrangements/3

Generates all K-arrangements (ordered K-element selections with replacement) of a list using default order.

Compilation flags:
    static

Template:
    arrangements(K,List,Arrangements)
Mode and number of proofs:
    arrangements(+integer,+list,-list) - one

---

arrangement/3

True iff the third argument is a K-arrangement with replacement of a list using default order.

Compilation flags:
    static

Template:
    arrangement(K,List,Arrangement)
Mode and number of proofs:
    arrangement(+integer,+list,-list) - one_or_more

---

arrangements/4

Generates all K-arrangements with the given order: default or lexicographic.

Compilation flags:
    static

Template:
    arrangements(K,List,Order,Arrangements)
Mode and number of proofs:
    arrangements(+integer,+list,+atom,-list) - one

---

arrangement/4

True iff the fourth argument is a K-arrangement with replacement with the given order: default or lexicographic.

Compilation flags:
    static

Template:
    arrangement(K,List,Order,Arrangement)
Mode and number of proofs:
    arrangement(+integer,+list,+atom,-list) - one_or_more

---

distinct_arrangements/3

Generates all distinct K-arrangements (deduplicating repeated values in the input list) using default order.

Compilation flags:
    static

Template:
    distinct_arrangements(K,List,Arrangements)
Mode and number of proofs:
    distinct_arrangements(+integer,+list,-list) - one

---

distinct_arrangement/3

True iff the third argument is a distinct K-arrangement with replacement of a list using default order.

Compilation flags:
    static

Template:
    distinct_arrangement(K,List,Arrangement)
Mode and number of proofs:
    distinct_arrangement(+integer,+list,-list) - one_or_more

---

distinct_arrangements/4

Generates all distinct K-arrangements with the given order: default or lexicographic.

Compilation flags:
    static

Template:
    distinct_arrangements(K,List,Order,Arrangements)
Mode and number of proofs:
    distinct_arrangements(+integer,+list,+atom,-list) - one

---

distinct_arrangement/4

True iff the fourth argument is a distinct K-arrangement with replacement with the given order: default or lexicographic.

Compilation flags:
    static

Template:
    distinct_arrangement(K,List,Order,Arrangement)

Mode and number of proofs:
distinct_arrangement(+integer,+list,+atom,-list) - one_or_more

---

nth_arrangement/4

Returns the K-arrangement at a given zero-based index using default order.

Compilation flags:
static

Template:
nth_arrangement(K,List,Index,Arrangement)
Mode and number of proofs:
nth_arrangement(+integer,+list,+integer,-list) - zero_or_one

---

nth_arrangement/5

Returns the K-arrangement at a given zero-based index in the given order: default or lexicographic.

Compilation flags:
static

Template:
nth_arrangement(K,List,Order,Index,Arrangement)
Mode and number of proofs:
nth_arrangement(+integer,+list,+atom,+integer,-list) - zero_or_one

---

arrangement_index/4

Returns the zero-based index of a K-arrangement using default order.

Compilation flags:
static

Template:
arrangement_index(K,List,Arrangement,Index)
Mode and number of proofs:
arrangement_index(+integer,+list,+list,-integer) - zero_or_one

---

arrangement_index/5

Returns the zero-based index of a K-arrangement in the given order: default or lexicographic.

Compilation flags:
static

Template:
arrangement_index(K,List,Order,Arrangement,Index)
Mode and number of proofs:
arrangement_index(+integer,+list,+atom,+list,-integer) - zero_or_one

---

count_arrangements/3

Counts the number of K-arrangements with replacement of a list.

Compilation flags:
static

Template:
count_arrangements(K,List,Count)
Mode and number of proofs:
count_arrangements(+integer,+list,-integer) - one

---

count_distinct_arrangements/3

Counts the number of distinct K-arrangements with replacement of a list (deduplicating repeated values in the input list).

Compilation flags:
    static

Template:
    count_distinct_arrangements(K,List,Count)
Mode and number of proofs:
    count_distinct_arrangements(+integer,+list,-integer) - one

---

nth_distinct_arrangement/4

Returns the distinct K-arrangement at a given zero-based index in default generation order.

Compilation flags:
    static

Template:
    nth_distinct_arrangement(K,List,Index,Arrangement)
Mode and number of proofs:
    nth_distinct_arrangement(+integer,+list,+integer,-list) - zero_or_one

---

distinct_arrangement_index/4

Returns the zero-based index of a distinct K-arrangement in default generation order.

Compilation flags:
    static

Template:
    distinct_arrangement_index(K,List,Arrangement,Index)
Mode and number of proofs:
    distinct_arrangement_index(+integer,+list,+list,-integer) - zero_or_one

---

random_arrangement/3

Returns a random K-arrangement with replacement of a list.

Compilation flags:
    static

Template:
    random_arrangement(K,List,Arrangement)
Mode and number of proofs:
    random_arrangement(+integer,+list,-list) - zero_or_one

---

sample_arrangements/4

Returns SampleCount random K-arrangements with replacement of a list, sampled with replacement.

Compilation flags:
    static

Template:
    sample_arrangements(K,List,SampleCount,Samples)
Mode and number of proofs:
    sample_arrangements(+integer,+list,+integer,-list) - zero_or_one

---

random_distinct_arrangement/3

Returns a random distinct K-arrangement with replacement of a list (deduplicating repeated values in the input list).

Compilation flags:
    static

Template:
    random_distinct_arrangement(K,List,Arrangement)

Mode and number of proofs:
    random_distinct_arrangement(+integer,+list,-list) - zero_or_one

---

sample_distinct_arrangements/4

Returns SampleCount random distinct K-arrangements with replacement of a list, sampled with replacement after deduplicating repeated values in the input list.

Compilation flags:
    static

Template:
    sample_distinct_arrangements(K,List,SampleCount,Samples)
Mode and number of proofs:
    sample_distinct_arrangements(+integer,+list,+integer,-list) - zero_or_one

---

next_arrangement/3

Returns the next distinct arrangement value in lexicographic order induced by the first argument.

Compilation flags:
    static

Template:
    next_arrangement(List,Arrangement,Next)
Mode and number of proofs:
    next_arrangement(+list,+list,-list) - zero_or_one

---

previous_arrangement/3

Returns the previous distinct arrangement value in lexicographic order induced by the first argument.

Compilation flags:
    static

Template:
    previous_arrangement(List,Arrangement,Previous)
Mode and number of proofs:
    previous_arrangement(+list,+list,-list) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.9 assertions

object

1.9.1 assertions

Proxy object for simplifying the use of the assertion meta-predicates.

Availability:
    logtalk_load(assertions(loader))

Author: Paulo Moura
Version: 2:0:0
Date: 2014-04-03

Compilation flags:
static, context_switching_calls

Extends:
public assertions(_)

Remarks:
(none)

Inherited public predicates:
assertion/1 assertion/2 goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.9.2 assertions(Mode)

A simple assertions framework. Can be used as a hook object for either suppressing assertions (production mode) or expanding them with file context information (debug mode).

Availability:
logtalk_load(assertions(loader))

Author: Paulo Moura
Version: 2:2:2
Date: 2022-07-04

Compilation flags:
static, context_switching_calls

Implements:
public expanding
Uses:
logtalk

Remarks:
(none)

Inherited public predicates:
goal_expansion/2 term_expansion/2

- Public predicates
  - assertion/1
  - assertion/2
- Protected predicates
- Private predicates
- Operators

Public predicates

assertion/1

Checks that an assertion is true. Uses the structured message printing mechanism for printing the results using a silent message for assertion success and a error message for assertion failure.

Compilation flags:
    static

Template:
    assertion(Goal)
Meta-predicate template:
    assertion(0)
Mode and number of proofs:
    assertion(@callable) - one

---

assertion/2

Checks that an assertion is true. Uses the structured message printing mechanism for printing the results using a silent message for assertion success and a error message for assertion failure. The context argument can be used to e.g. pass location data.

Compilation flags:
    static

Template:
    assertion(Context,Goal)
Meta-predicate template:
    assertion(*,0)
Mode and number of proofs:
    assertion(@term,@callable) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.9.3 assertions_messages

Assertions framework default message translations.

Availability:
logtalk_load(assertions(loader))

Author: Paulo Moura
Version: 2:2:0
Date: 2018-02-20

Compilation flags:
static

Provides:
logtalk::message_prefix_stream/4
logtalk::message_tokens//2

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
- Protected predicates

- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.10 assignvars

object

1.10.1 assignvars

Assignable variables (supporting backtracable assignment of non-variable terms).

Availability:
    logtalk_load(assignvars(loader))

Author: Nobukuni Kino and Paulo Moura
Version: 1:7:0
Date: 2018-07-11

Compilation flags:
    static, context_switching_calls

Implements:
    public assignvarsp

Remarks:
(none)

Inherited public predicates:
(<=)/2 (=>)/2 assignable/1 assignable/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.10.2 assignvarsp

Assignable variables (supporting backtracable assignment of non-variable terms) protocol.

Availability:
logtalk_load(assignvars(loader))

Author: Nobukuni Kino and Paulo Moura
Version: 1:0:1
Date: 2019-06-10

Compilation flags:

static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - assignable/1
  - assignable/2
  - (<=)/2
  - (=>)/2
- Protected predicates
- Private predicates
- Operators
  - op(100,xfx,<=)
  - op(100,xfx,=>)

Public predicates

assignable/1

Makes Variable an assignable variable. Initial state will be empty.

Compilation flags:
static

Template:
assignable(Variable)
Mode and number of proofs:
assignable(--assignvar) - one

Exceptions:
Variable is not a variable:

type_error(variable,Variable)

---

assignable/2

Makes Variable an assignable variable and sets its initial state to Value.

Compilation flags:
static

Template:
assignable(Variable,Value)
Mode and number of proofs:
assignable(--assignvar,@nonvar) - one

Exceptions:
Variable is not a variable:
type_error(variable,Variable)
Value is not instantiated:
instantiation_error

(<=)/2

Sets the state of the assignable variable Variable to Value (initializing the variable if needed).

Compilation flags:
static

Template:
Variable<=Value
Mode and number of proofs:
(?assignvar)<=(@nonvar) - one

Exceptions:
Value is not instantiated:
instantiation_error

(=>)/2

Unifies Value with the current state of the assignable variable Variable.

Compilation flags:
    static

Template:
    Variable=>Value
Mode and number of proofs:
    +assignvar=> ?nonvar - zero_or_one

Exceptions:
    Variable is not instantiated:
        instantiation_error

Protected predicates

(none)

Private predicates

(none)

Operators

op(100,xfx,<=)

Scope:
    public

op(100,xfx,=>)

Scope:
    public

See also

assignvars

1.11 avro

object

1.11.1 avro

Apache Avro binary format parser and generator.

Availability:
logtalk_load(avro(loader))

Author: Paulo Moura
Version: 1:2:0
Date: 2026-05-09

Compilation flags:
static, context_switching_calls

Uses:
ieee_754(Precision,ByteOrder,NaNRepresentation)
json(ObjectRepresentation,PairRepresentation,StringRepresentation)
list
reader

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - parse/2
  - parse/3
  - generate/3
  - generate/4
- Protected predicates
- Private predicates
- Operators

Public predicates

parse/2

Parses Avro binary data from the given source (bytes(List), stream(Stream), or file(Path)) returning a Schema-Data pair. When the schema is not present in the file, Schema is unified with false.

Compilation flags:
    static

Template:
    parse(Source,Schema-Data)
Mode and number of proofs:
    parse(++compound,--pair) - one_or_error

---

parse/3

Parses Avro binary data from the given source using the provided schema, returning the decoded data.

Compilation flags:
    static

Template:
    parse(Source,Schema,Data)
Mode and number of proofs:
    parse(++compound,++term,--term) - one_or_error

---

generate/3

Generates Avro binary data to the given sink (bytes(List), stream(Stream), or file(Path)) from the given schema and data. The schema is not included in the output.

Compilation flags:
    static

Template:
    generate(Sink,Schema,Data)

Mode and number of proofs:
    generate(++compound,++term,++term) - one_or_error

---

generate/4

Generates Avro binary data to the given sink from the given schema and data. When IncludeSchema is true, generates an Avro Object Container File with the schema embedded.

Compilation flags:
    static

Template:
    generate(Sink,IncludeSchema,Schema,Data)
Mode and number of proofs:
    generate(++compound,++boolean,++term,++term) - one_or_error

---

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.12 base32

object

1.12.1 base32

Base32 encoder and decoder (RFC 4648).

Availability:
logtalk_load(base32(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2025-01-23

Compilation flags:
static, context_switching_calls

Uses:
reader

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - parse/2
  - generate/2
- Protected predicates
- Private predicates
- Operators

Public predicates

parse/2

Parses Base32 data from the given source (atom(Atom), chars(List), codes(List), stream(Stream), or file(Path)) into a list of bytes.

Compilation flags:
static

Template:
parse(Source,Bytes)
Mode and number of proofs:
parse(++compound,--list(byte)) - one_or_error

---

generate/2

Generates Base32 in the representation specified in the first argument (atom(Atom), chars(List), codes(List), stream(Stream), or file(Path)) for the list of bytes in the second argument.

Compilation flags:
static

Template:
generate(Sink,Bytes)
Mode and number of proofs:
generate(+compound,+list(byte)) - one_or_error

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.13 base58

object

1.13.1 base58

Base58 encoder and decoder (Bitcoin alphabet variant).

Availability:
logtalk_load(base58(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-04-17

Compilation flags:
static, context_switching_calls

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - parse/2
  - generate/2
- Protected predicates
- Private predicates
- Operators

Public predicates

parse/2

Parses Base58 data from the given source (atom(Atom), chars(List), or codes(List)) into a list of bytes.

Compilation flags:
static

Template:
parse(Source,Bytes)
Mode and number of proofs:
parse(++compound,--list(byte)) - one_or_error

---

generate/2

Generates Base58 in the representation specified in the first argument (atom(Atom), chars(List), or codes(List)) for the list of bytes in the second argument.

Compilation flags:
static

Template:
generate(Sink,Bytes)
Mode and number of proofs:
generate(+compound,+list(byte)) - one_or_error

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.14 base64

object

1.14.1 base64

Base64 parser and generator.

Availability:
logtalk_load(base64(loader))

Author: Paulo Moura
Version: 0:10:0
Date: 2021-03-22

Compilation flags:
static, context_switching_calls

Uses:
reader

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - parse/2
  - generate/2
- Protected predicates
- Private predicates
- Operators

Public predicates

parse/2

Parses the Base64 data from the given source (atom(Atom), chars(List), codes(List), stream(Stream), or file(Path) into a list of bytes.

Compilation flags:
static

Template:
parse(Source,Bytes)
Mode and number of proofs:
parse(++compound,--list(byte)) - one_or_error

---

generate/2

Generates Base64 in the representation specified in the first argument (atom(Atom), chars(List), codes(List), stream(Stream), or file(Path) for the list of bytes in the second argument.

Compilation flags:
static

Template:
generate(Sink,Bytes)
Mode and number of proofs:
generate(+compound,+list(byte)) - one_or_error

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.14.2 base64url

Base64URL parser and generator.

Availability:
logtalk_load(base64(loader))

Author: Paulo Moura
Version: 0:9:0
Date: 2021-03-10

Compilation flags:
static, context_switching_calls

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - parse/2
  - generate/2
- Protected predicates
- Private predicates
- Operators

Public predicates

parse/2

Parses the Base64URL data from the given source (atom(Atom), chars(List), or codes(List) into a URL (using the same format as the source).

Compilation flags:
static

Template:
parse(Source,URL)
Mode and number of proofs:
parse(++compound,--types([atom,chars,codes])) - one_or_error

generate/2

Generates Base64URL data in the representation specified in the first argument (atom(Atom), chars(List), or codes(List) for the given URL (given in the same format as the sink).

Compilation flags:
static

Template:
generate(Sink,URL)
Mode and number of proofs:
generate(+compound,+types([atom,chars,codes])) - one_or_error

---

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.15 base85

object

1.15.1 base85

Base85 encoder and decoder (Ascii85/btoa variant).

Availability:
logtalk_load(base85(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2025-01-23

Compilation flags:
static, context_switching_calls

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - parse/2
  - generate/2
- Protected predicates
- Private predicates
- Operators

Public predicates

parse/2

Parses Base85 (Ascii85) data from the given source (atom(Atom), chars(List), or codes(List)) into a list of bytes.

Compilation flags:
static

Template:
parse(Source,Bytes)
Mode and number of proofs:
parse(++compound,--list(byte)) - one_or_error

generate/2

Generates Base85 (Ascii85) in the representation specified in the first argument (atom(Atom), chars(List), or codes(List)) for the list of bytes in the second argument.

Compilation flags:
static

Template:
generate(Sink,Bytes)
Mode and number of proofs:
generate(+compound,+list(byte)) - one_or_error

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.16 bayesian_ridge_regression

object

1.16.1 bayesian_ridge_regression

Bayesian ridge regression regressor supporting continuous and mixed-feature datasets using evidence maximization and posterior uncertainty over coefficients. Learns from a dataset object implementing the regression_dataset_protocol protocol and returns a regressor term that can be used for prediction, predictive-distribution queries, and export as predicate clauses.

Availability:
logtalk_load(bayesian_ridge_regression(loader))

Author: Paulo Moura

Version: 1:0:0
Date: 2026-05-07

Compilation flags:
static, context_switching_calls

Imports:
public regressor_common
Uses:
format
linear_algebra
list
numberlist
population
type

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_regressor/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 print_regressor/1 regressor_options/2 valid_option/1 valid_options/1 valid_regressor/1

- Public predicates
  - predict_distribution/3
  - weight_variances/2
- Protected predicates
- Private predicates
- Operators

Public predicates

predict_distribution/3

Predicts the posterior predictive Gaussian distribution for a new instance using the learned regressor. The returned term has the shape gaussian(Mean, Variance) where Variance includes the learned observation noise variance and coefficient posterior uncertainty only; the intercept is not treated as a probabilistic parameter.

Compilation flags:
    static

Template:
    predict_distribution(Regressor,Instance,Distribution)
Mode and number of proofs:
    predict_distribution(+compound,+list,-compound) - one

---

weight_variances/2

Returns the posterior marginal variances of the encoded feature coefficients in encoder order. Encoded features dropped from fitting because they had zero variance are reported with posterior variance zero.

Compilation flags:
    static

Template:
    weight_variances(Regressor,Variances)
Mode and number of proofs:
    weight_variances(+compound,-list(float)) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

linear_regression, ridge_regression, lasso_regression, elastic_net_regression, gaussian_process_regression, knn_regression, regression_tree, random_forest_regression, gradient_boosting_regression

1.17 borda_ranker

object

1.17.1 borda_ranker

Borda grouped-ranking ranker. Learns one deterministic score per item from a dataset object implementing the ranking_dataset_protocol protocol by summing per-group Borda points and returns a self-describing ranker term with diagnostics that can be used for ranking and export.

Availability:
logtalk_load(borda_ranker(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public ranking_dataset_common
public score_ranker_model_common
Uses:
avltree

Remarks:
(none)

Inherited public predicates:

check_option/1 check_options/1 check_ranker/1 default_option/1 default_options/1
diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 grouped_dataset_groups/2
grouped_dataset_items/2 grouped_dataset_items/3 grouped_dataset_summary/2 learn/2
learn/3 option/2 option/3 pairwise_dataset_connected_components/2
pairwise_dataset_declared_items/2 pairwise_dataset_items/2 pairwise_dataset_matchups/2
pairwise_dataset_preferences/2 pairwise_dataset_summary/2 pairwise_dataset_win_totals/2
pairwise_measurement_dataset_connected_components/2
pairwise_measurement_dataset_declared_items/2 pairwise_measurement_dataset_items/2
pairwise_measurement_dataset_measurements/2 pairwise_measurement_dataset_summary/2
print_ranker/1 rank/3 ranker_options/2 scores/2
temporal_pairwise_dataset_connected_components/2
temporal_pairwise_dataset_declared_items/2 temporal_pairwise_dataset_games/2
temporal_pairwise_dataset_games/3 temporal_pairwise_dataset_items/2
temporal_pairwise_dataset_periods/2 temporal_pairwise_dataset_summary/2 valid_option/1
valid_options/1 valid_ranker/1 validate_grouped_dataset/1 validate_grouped_dataset/2
validate_pairwise_dataset/1 validate_pairwise_dataset/2
validate_pairwise_measurement_dataset/1 validate_pairwise_measurement_dataset/2
validate_temporal_pairwise_dataset/1 validate_temporal_pairwise_dataset/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

ranking_dataset_protocol, ranker_protocol, copeland_ranker

1.18 bradley_terry_ranker

object

1.18.1 bradley_terry_ranker

Bradley-Terry pairwise preference ranker. Learns one positive strength parameter per item from a dataset object implementing the pairwise_ranking_dataset_protocol protocol when the directed win graph admits a finite Bradley-Terry maximum-likelihood estimate, and returns a self-describing ranker term with diagnostics that can be used for ranking and export.

Availability:
logtalk_load(bradley_terry_ranker(loader))

Author: Paulo Moura
Version: 2:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public ranking_dataset_common
public pairwise_strength_ranker_common
public score_ranker_common
Uses:
avltree
format
list

Remarks:
(none)

Inherited public predicates:

check_option/1 check_options/1 check_ranker/1 default_option/1 default_options/1
diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 grouped_dataset_groups/2
grouped_dataset_items/2 grouped_dataset_items/3 grouped_dataset_summary/2 learn/2
learn/3 option/2 option/3 pairwise_dataset_connected_components/2
pairwise_dataset_declared_items/2 pairwise_dataset_items/2 pairwise_dataset_matchups/2
pairwise_dataset_preferences/2 pairwise_dataset_summary/2 pairwise_dataset_win_totals/2
pairwise_measurement_dataset_connected_components/2
pairwise_measurement_dataset_declared_items/2 pairwise_measurement_dataset_items/2
pairwise_measurement_dataset_measurements/2 pairwise_measurement_dataset_summary/2
print_ranker/1 rank/3 ranker_options/2 scores/2
temporal_pairwise_dataset_connected_components/2
temporal_pairwise_dataset_declared_items/2 temporal_pairwise_dataset_games/2
temporal_pairwise_dataset_games/3 temporal_pairwise_dataset_items/2
temporal_pairwise_dataset_periods/2 temporal_pairwise_dataset_summary/2 valid_option/1
valid_options/1 valid_ranker/1 validate_grouped_dataset/1 validate_grouped_dataset/2
validate_pairwise_dataset/1 validate_pairwise_dataset/2
validate_pairwise_measurement_dataset/1 validate_pairwise_measurement_dataset/2
validate_temporal_pairwise_dataset/1 validate_temporal_pairwise_dataset/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

pairwise_ranking_dataset_protocol, ranking_dataset_protocol, ranker_protocol, regularized_bradley_terry_ranker

1.19 c45_classifier

object

1.19.1 c45_classifier

C4.5 decision tree learning algorithm. Builds a decision tree from a dataset object implementing the dataset_protocol protocol and provides predicates for exporting the learned tree as a list of predicate clauses or to a file. Supports both discrete and continuous attributes, handles missing values, and supports tree pruning.

Availability:
logtalk_load(c45_classifier(loader))

Author: Paulo Moura
Version: 2:0:0
Date: 2026-05-11

Compilation flags:
static, context_switching_calls

Imports:
public classifier_common
Uses:
format
list
numberlist
pairs
type

Remarks:
(none)

Inherited public predicates:
check_classifier/1 check_option/1 check_options/1 classifier_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 print_classifier/1 valid_classifier/1 valid_option/1 valid_options/1

- Public predicates
  - prune/5

  - prune/3
- Protected predicates
- Private predicates
- Operators

Public predicates

prune/5

Prunes a decision tree using pessimistic error pruning (PEP). This post-pruning method estimates error rates using the upper confidence bound of the binomial distribution with the given confidence factor and replaces subtrees with leaf nodes when doing so would not increase the estimated error. Pruning helps reduce overfitting and can improve generalization to unseen data.

Compilation flags:
    static

Template:
    prune(Dataset,Tree,ConfidenceFactor,MinInstances,PrunedTree)
Mode and number of proofs:
    prune(+object_identifier,+tree,+float,+positive_integer,-tree) - one

Remarks:

- Confidence factor: The confidence factor controls the aggressiveness of pruning. It must be in the range (0.0, 1.0). Lower values result in more aggressive pruning (smaller, simpler trees), while higher values result in less pruning (larger, more complex trees). The default value is 0.25.
- Minimum instances per leaf: The minimum number of instances required at a leaf node. When a node has fewer instances than this value, the node may be pruned. It must be a positive integer. The default value is 2.
- Statistical basis: The pruning uses the upper confidence bound of the binomial distribution to estimate the true error rate.

---

prune/3

Prunes a decision tree using pessimistic error pruning (PEP) with default parameter values. Calls prune/5 with ConfidenceFactor = 0.25 and MinInstances = 2.

Compilation flags:
    static

Template:
    prune(Dataset,Tree,PrunedTree)
Mode and number of proofs:
    prune(+object_identifier,+tree,-tree) - one

Remarks:

- Default parameters: Uses the standard C4.5 default values: confidence factor of 0.25 (the confidence level for computing the upper bound of the error estimate) and minimum instances per leaf of 2.

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

dataset_protocol, isolation_forest_anomaly_detector, knn_classifier, naive_bayes_classifier, nearest_centroid_classifier, random_forest_classifier, adaptive_boosting_classifier

1.20 cartesian_products

object

1.20.1 cartesian_products

Implementation of Cartesian-product operations over lists.

Availability:
logtalk_load(cartesian_products(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-12

Compilation flags:
static, context_switching_calls

Implements:
public cartesian_products_protocol
Uses:
fast_random(Algorithm)
list

Remarks:
(none)

Inherited public predicates:
cartesian_product/2 cartesian_product/3 cartesian_tuple/2 cartesian_tuple/3
cartesian_tuple_index/3 cartesian_tuple_index/4 count_cartesian_product/2
count_distinct_cartesian_product/2 distinct_cartesian_product/2 distinct_cartesian_product/3
distinct_cartesian_tuple/2 distinct_cartesian_tuple/3 distinct_cartesian_tuple_index/3
next_cartesian_tuple/3 nth_cartesian_tuple/3 nth_cartesian_tuple/4
nth_distinct_cartesian_tuple/3 previous_cartesian_tuple/3 random_cartesian_tuple/2
random_distinct_cartesian_tuple/2 sample_cartesian_tuples/3
sample_distinct_cartesian_tuples/3

- Public predicates
- Protected predicates
- Private predicates

- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.20.2 cartesian_products_protocol

Protocol for Cartesian-product operations over lists.

Availability:
logtalk_load(cartesian_products(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-12

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:

(none)

- Public predicates
  - cartesian_product/2
  - cartesian_tuple/2
  - cartesian_product/3
  - cartesian_tuple/3
  - distinct_cartesian_product/2
  - distinct_cartesian_tuple/2
  - distinct_cartesian_product/3
  - distinct_cartesian_tuple/3
  - count_cartesian_product/2
  - count_distinct_cartesian_product/2
  - nth_cartesian_tuple/3
  - nth_cartesian_tuple/4
  - cartesian_tuple_index/3
  - cartesian_tuple_index/4
  - nth_distinct_cartesian_tuple/3
  - distinct_cartesian_tuple_index/3
  - random_cartesian_tuple/2
  - sample_cartesian_tuples/3
  - random_distinct_cartesian_tuple/2
  - sample_distinct_cartesian_tuples/3
  - next_cartesian_tuple/3
  - previous_cartesian_tuple/3
- Protected predicates
- Private predicates
- Operators

Public predicates

cartesian_product/2

Generates the Cartesian product of a list of lists using default order.

Compilation flags:
    static

Template:
    cartesian_product(Lists,Product)
Mode and number of proofs:
    cartesian_product(+list(list),-list) - one

---

cartesian_tuple/2

True iff the second argument is a tuple obtained by selecting one element from each list in the first argument using default order.

Compilation flags:
    static

Template:
    cartesian_tuple(Lists,Tuple)
Mode and number of proofs:
    cartesian_tuple(+list(list),-list) - one_or_more

---

cartesian_product/3

Generates the Cartesian product with the given order: default or lexicographic.

Compilation flags:
    static

Template:
    cartesian_product(Lists,Order,Product)
Mode and number of proofs:

cartesian_product(+list(list),+atom,-list) - one

---

cartesian_tuple/3

True iff the third argument is a Cartesian-product tuple with the given order: default or lexicographic.

Compilation flags:
static

Template:
cartesian_tuple(Lists,Order,Tuple)
Mode and number of proofs:
cartesian_tuple(+list(list),+atom,-list) - one_or_more

---

distinct_cartesian_product/2

Generates the Cartesian product after deduplicating repeated values in each factor list using default order.

Compilation flags:
static

Template:
distinct_cartesian_product(Lists,Product)
Mode and number of proofs:
distinct_cartesian_product(+list(list),-list) - one

---

distinct_cartesian_tuple/2

True iff the second argument is a tuple in the distinct Cartesian product of the factor lists using default order.

Compilation flags:
static

Template:
distinct_cartesian_tuple(Lists,Tuple)
Mode and number of proofs:
distinct_cartesian_tuple(+list(list),-list) - one_or_more

---

distinct_cartesian_product/3

Generates the distinct Cartesian product with the given order: default or lexicographic.

Compilation flags:
static

Template:
distinct_cartesian_product(Lists,Order,Product)
Mode and number of proofs:
distinct_cartesian_product(+list(list),+atom,-list) - one

---

distinct_cartesian_tuple/3

True iff the third argument is a tuple in the distinct Cartesian product with the given order: default or lexicographic.

Compilation flags:
static

Template:
distinct_cartesian_tuple(Lists,Order,Tuple)
Mode and number of proofs:
distinct_cartesian_tuple(+list(list),+atom,-list) - one_or_more

---

count_cartesian_product/2

Counts the number of tuples in the Cartesian product of a list of lists.

Compilation flags:
    static

Template:
    count_cartesian_product(Lists,Count)
Mode and number of proofs:
    count_cartesian_product(+list(list),-integer) - one

---

count_distinct_cartesian_product/2

Counts the number of tuples in the distinct Cartesian product after deduplicating repeated values in each factor list.

Compilation flags:
    static

Template:
    count_distinct_cartesian_product(Lists,Count)
Mode and number of proofs:
    count_distinct_cartesian_product(+list(list),-integer) - one

---

nth_cartesian_tuple/3

Returns the Cartesian-product tuple at a given zero-based index using default order.

Compilation flags:
    static

Template:
    nth_cartesian_tuple(Lists,Index,Tuple)
Mode and number of proofs:
    nth_cartesian_tuple(+list(list),+integer,-list) - zero_or_one

---

nth_cartesian_tuple/4

Returns the Cartesian-product tuple at a given zero-based index in the given order: default or lexicographic.

Compilation flags:
    static

Template:
    nth_cartesian_tuple(Lists,Order,Index,Tuple)
Mode and number of proofs:
    nth_cartesian_tuple(+list(list),+atom,+integer,-list) - zero_or_one

---

cartesian_tuple_index/3

Returns the zero-based index of a Cartesian-product tuple using default order.

Compilation flags:
    static

Template:
    cartesian_tuple_index(Lists,Tuple,Index)
Mode and number of proofs:
    cartesian_tuple_index(+list(list),+list,-integer) - zero_or_one

---

cartesian_tuple_index/4

Returns the zero-based index of a Cartesian-product tuple in the given order: default or lexicographic.

Compilation flags:
    static

Template:
    cartesian_tuple_index(Lists,Order,Tuple,Index)
Mode and number of proofs:

cartesian_tuple_index(+list(list),+atom,+list,-integer) - zero_or_one

---

nth_distinct_cartesian_tuple/3

Returns the distinct Cartesian-product tuple at a given zero-based index in default generation order.

Compilation flags:
static

Template:
nth_distinct_cartesian_tuple(Lists,Index,Tuple)
Mode and number of proofs:
nth_distinct_cartesian_tuple(+list(list),+integer,-list) - zero_or_one

---

distinct_cartesian_tuple_index/3

Returns the zero-based index of a distinct Cartesian-product tuple in default generation order.

Compilation flags:
static

Template:
distinct_cartesian_tuple_index(Lists,Tuple,Index)
Mode and number of proofs:
distinct_cartesian_tuple_index(+list(list),+list,-integer) - zero_or_one

---

random_cartesian_tuple/2

Returns a random Cartesian-product tuple.

Compilation flags:
static

Template:

random_cartesian_tuple(Lists,Tuple)
Mode and number of proofs:
random_cartesian_tuple(+list(list),-list) - zero_or_one

---

sample_cartesian_tuples/3

Returns SampleCount random Cartesian-product tuples, sampled with replacement.

Compilation flags:
static

Template:
sample_cartesian_tuples(Lists,SampleCount,Samples)
Mode and number of proofs:
sample_cartesian_tuples(+list(list),+integer,-list) - zero_or_one

---

random_distinct_cartesian_tuple/2

Returns a random tuple from the distinct Cartesian product after deduplicating repeated values in each factor list.

Compilation flags:
static

Template:
random_distinct_cartesian_tuple(Lists,Tuple)
Mode and number of proofs:
random_distinct_cartesian_tuple(+list(list),-list) - zero_or_one

---

sample_distinct_cartesian_tuples/3

Returns SampleCount random tuples from the distinct Cartesian product, sampled with replacement.

Compilation flags:
    static

Template:
    sample_distinct_cartesian_tuples(Lists,SampleCount,Samples)
Mode and number of proofs:
    sample_distinct_cartesian_tuples(+list(list),+integer,-list) - zero_or_one

---

next_cartesian_tuple/3

Returns the next distinct Cartesian-product tuple in lexicographic order induced by the factor lists.

Compilation flags:
    static

Template:
    next_cartesian_tuple(Lists,Tuple,Next)
Mode and number of proofs:
    next_cartesian_tuple(+list(list),+list,-list) - zero_or_one

---

previous_cartesian_tuple/3

Returns the previous distinct Cartesian-product tuple in lexicographic order induced by the factor lists.

Compilation flags:
    static

Template:
    previous_cartesian_tuple(Lists,Tuple,Previous)
Mode and number of proofs:
    previous_cartesian_tuple(+list(list),+list,-list) - zero_or_one

---

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.21 cbor

object

1.21.1 cbor

Concise Binary Object Representation (CBOR) format exporter and importer. Uses atoms to represent decoded CBOR strings.

Availability:
logtalk_load(cbor(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2021-03-04

Compilation flags:
static, context_switching_calls

Extends:
public cbor(atom)

Remarks:
(none)

Inherited public predicates:
generate/2 parse/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.21.2 cbor(StringRepresentation)

- StringRepresentation - Text representation to be used when decoding CBOR strings. Possible values are atom (default), chars, and codes.

Concise Binary Object Representation (CBOR) format exporter and importer.

Availability:
logtalk_load(cbor(loader))

Author: Paulo Moura
Version: 0:12:0
Date: 2026-05-09

Compilation flags:
static, context_switching_calls

Uses:
ieee_754(Precision,ByteOrder,NaNRepresentation)

list

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - parse/2
  - generate/2
- Protected predicates
- Private predicates
- Operators

Public predicates

parse/2

Parses a list of bytes in the CBOR format returning the corresponding term representation. Throws an error when parsing is not possible (usually due to an invalid byte sequence).

Compilation flags:
static

Template:
parse(Bytes,Term)
Mode and number of proofs:
parse(@list(byte),-ground) - one_or_error

generate/2

Generates a list of bytes in the CBOR format representing the given term. Throws an error when generating is not possible (usually due to a term that have no CBOR corresponding representation).

Compilation flags:
    static

Template:
    generate(Term,Bytes)
Mode and number of proofs:
    generate(@ground,-list(byte)) - one_or_error

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.22 ccsds_frames

object

1.22.1 ccsds_aos_frames(FrameLength,InsertZoneLength,HasOCF,HasFECF)

- FrameLength - Fixed AOS transfer frame length in octets.
- InsertZoneLength - Length in octets of the optional insert zone when present.
- HasOCF - Boolean indicating if frames include the operational control field.
- HasFECF - Boolean indicating if frames include the frame error control field.

CCSDS advanced orbiting systems transfer frame parser and generator.

Availability:
    logtalk_load(ccsds_frames(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-09

Compilation flags:
static, context_switching_calls

Implements:
public ccsds_frame_protocol
Uses:
list
reader
type

Remarks:
(none)

Inherited public predicates:
data_field/2 fecf/2 frame_type/2 generate/2 generate/3 ocf/2 parse/2 spacecraft_id/2 valid/1 version/2 virtual_channel_id/2

- Public predicates
  - virtual_channel_frame_count/2
  - signaling_field/2
  - insert_zone/2
  - update_fecf/2
  - verify_fecf/1
- Protected predicates
- Private predicates
- Operators

Public predicates

virtual_channel_frame_count/2

Extracts the virtual channel frame count.

Compilation flags:
    static

Template:
    virtual_channel_frame_count(Frame,Count)
Mode and number of proofs:
    virtual_channel_frame_count(+compound,-integer) - one

---

signaling_field/2

Extracts the signaling field term.

Compilation flags:
    static

Template:
    signaling_field(Frame,SignalingField)
Mode and number of proofs:
    signaling_field(+compound,-compound) - one

---

insert_zone/2

Extracts the optional insert zone. Returns none when absent.

Compilation flags:
    static

Template:
    insert_zone(Frame,InsertZone)
Mode and number of proofs:
    insert_zone(+compound,-compound) - one

---

update_fecf/2

Computes the correct AOS transfer frame FECF for the selected object configuration and returns the corresponding updated frame term.

Compilation flags:
    static

Template:
    update_fecf(Frame,UpdatedFrame)
Mode and number of proofs:
    update_fecf(+compound,-compound) - one_or_error

---

verify_fecf/1

True if the AOS transfer frame term contains the correct FECF for the selected object configuration.

Compilation flags:
    static

Template:
    verify_fecf(Frame)
Mode and number of proofs:
    verify_fecf(@compound) - zero_or_one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.22.2 ccsds_frame_protocol

CCSDS transfer frame parser and generator protocol.

Availability:
logtalk_load(ccsds_frames(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-08

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - parse/2
  - generate/2
  - generate/3
  - valid/1
  - update_fecf/2

  - verify_fecf/1
  - frame_type/2
  - version/2
  - spacecraft_id/2
  - virtual_channel_id/2
  - data_field/2
  - ocf/2
  - fecf/2
- Protected predicates
- Private predicates
- Operators

Public predicates

parse/2

Parses CCSDS transfer frame(s) from a source. Supported source terms are file(File), stream(Stream), and bytes(Bytes).

Compilation flags:
    static

Template:
    parse(Source,Frames)
Mode and number of proofs:
    parse(+compound,-list(compound)) - one_or_error

---

generate/2

Generates CCSDS transfer frame bytes to a sink. Supported sink terms are file(File), stream(Stream), and bytes(Bytes).

Compilation flags:
    static

Template:

generate(Sink,Frames)
Mode and number of proofs:
generate(+compound,+list(compound)) - one_or_error

---

generate/3

Generates a list of bytes from a single CCSDS transfer frame term with an open tail. When the selected object includes an FECF, the generated bytes use a freshly computed FECF derived from the remaining frame fields.

Compilation flags:
static

Template:
generate(Frame,Bytes,Tail)
Mode and number of proofs:
generate(+compound,-list(byte),--variable) - one_or_error

---

valid/1

True if the argument is a valid transfer frame term for the selected object and, when applicable, carries the correct FECF for the remaining frame fields.

Compilation flags:
static

Template:
valid(Frame)
Mode and number of proofs:
valid(@compound) - zero_or_one

---

update_fecf/2

Computes the correct frame error control field for the selected object configuration and returns the corresponding updated frame term.

Compilation flags:
    static

Template:
    update_fecf(Frame,UpdatedFrame)
Mode and number of proofs:
    update_fecf(+compound,-compound) - one_or_error

---

verify_fecf/1

True if the frame term contains the correct FECF for the selected object configuration.

Compilation flags:
    static

Template:
    verify_fecf(Frame)
Mode and number of proofs:
    verify_fecf(@compound) - zero_or_one

---

frame_type/2

Returns the transfer frame type for a frame term.

Compilation flags:
    static

Template:
    frame_type(Frame,Type)
Mode and number of proofs:
    frame_type(+compound,-atom) - one

---

version/2

Extracts the transfer frame version number.

Compilation flags:
    static

Template:
    version(Frame,Version)
Mode and number of proofs:
    version(+compound,-integer) - one

---

spacecraft_id/2

Extracts the spacecraft identifier.

Compilation flags:
    static

Template:
    spacecraft_id(Frame,SpacecraftId)
Mode and number of proofs:
    spacecraft_id(+compound,-integer) - one

---

virtual_channel_id/2

Extracts the virtual channel identifier.

Compilation flags:
    static

Template:
    virtual_channel_id(Frame,VirtualChannelId)
Mode and number of proofs:

virtual_channel_id(+compound,-integer) - one

---

data_field/2

Extracts the transfer frame data field as a list of bytes.

Compilation flags:
static

Template:
data_field(Frame,DataField)
Mode and number of proofs:
data_field(+compound,-list(byte)) - one

---

ocf/2

Extracts the operational control field. Returns none when absent.

Compilation flags:
static

Template:
ocf(Frame,OCF)
Mode and number of proofs:
ocf(+compound,-compound) - one

---

fecf/2

Extracts the frame error control field. Returns none when absent.

Compilation flags:
static

Template:

fecf(Frame,FECF)
Mode and number of proofs:
fecf(+compound,-compound) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

object

1.22.3 ccsds_frames

Helpers for introspecting CCSDS transfer frame terms.

Availability:
logtalk_load(ccsds_frames(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-09

Compilation flags:
static, context_switching_calls

Uses:
ccsds_packets(SecondaryHeaderLength)
list
type

Remarks:
(none)

Inherited public predicates:

(none)

- Public predicates
  - valid/1
  - frame_type/2
  - version/2
  - spacecraft_id/2
  - virtual_channel_id/2
  - data_field/2
  - ocf/2
  - fecf/2
  - update_fecf/2
  - verify_fecf/1
  - extract_packets/3
  - insert_packets/4
- Protected predicates
- Private predicates
- Operators

Public predicates

valid/1

True if the argument is a known CCSDS transfer frame term.

Compilation flags:
static

Template:
valid(Frame)
Mode and number of proofs:
valid(@compound) - zero_or_one

frame_type/2

Returns the CCSDS transfer frame type for a known frame term.

Compilation flags:
    static

Template:
    frame_type(Frame,Type)
Mode and number of proofs:
    frame_type(+compound,-atom) - one

---

version/2

Extracts the transfer frame version number from a known frame term.

Compilation flags:
    static

Template:
    version(Frame,Version)
Mode and number of proofs:
    version(+compound,-integer) - one

---

spacecraft_id/2

Extracts the spacecraft identifier from a known frame term.

Compilation flags:
    static

Template:
    spacecraft_id(Frame,SpacecraftId)
Mode and number of proofs:
    spacecraft_id(+compound,-integer) - one

---

virtual_channel_id/2

Extracts the virtual channel identifier from a known frame term.

Compilation flags:
    static

Template:
    virtual_channel_id(Frame,VirtualChannelId)
Mode and number of proofs:
    virtual_channel_id(+compound,-integer) - one

---

data_field/2

Extracts the raw data field bytes from a known frame term.

Compilation flags:
    static

Template:
    data_field(Frame,DataField)
Mode and number of proofs:
    data_field(+compound,-list(byte)) - one

---

ocf/2

Extracts the operational control field. Returns none when absent.

Compilation flags:
    static

Template:
    ocf(Frame,OCF)
Mode and number of proofs:
    ocf(+compound,-compound) - one

---

fecf/2

Extracts the frame error control field. Returns none when absent.

Compilation flags:
    static

Template:
    fecf(Frame,FECF)
Mode and number of proofs:
    fecf(+compound,-compound) - one

---

update_fecf/2

Refreshes a present frame error control field from the remaining frame fields while leaving frames without an FECF unchanged.

Compilation flags:
    static

Template:
    update_fecf(Frame,UpdatedFrame)
Mode and number of proofs:
    update_fecf(+compound,-compound) - one_or_error

---

verify_fecf/1

True if a present frame error control field matches the remaining frame fields. Frames without an FECF succeed.

Compilation flags:
    static

Template:
    verify_fecf(Frame)
Mode and number of proofs:
    verify_fecf(@compound) - zero_or_one

---

extract_packets/3

Parses the raw frame data field as CCSDS space packet bytes using the given packet secondary header length.

Compilation flags:
    static

Template:
    extract_packets(Frame,SecondaryHeaderLength,Packets)
Mode and number of proofs:
    extract_packets(+compound,+integer,-list(compound)) - one_or_error

---

insert_packets/4

Generates CCSDS packet bytes and replaces the raw data field of a frame term, preserving the remaining header fields unchanged.

Compilation flags:
    static

Template:
    insert_packets(Packets,SecondaryHeaderLength,Frame,UpdatedFrame)
Mode and number of proofs:
    insert_packets(+list(compound),+integer,+compound,-compound) - one_or_error

---

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.22.4 ccsds_frames_types

Type definitions for CCSDS transfer frames.

Availability:
logtalk_load(ccsds_frames(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-08

Compilation flags:
static

Provides:
type::type/1
type::check/2
Uses:
ccsds_aos_frames(FrameLength,InsertZoneLength,HasOCF,HasFECF)
ccsds_tc_frames(FrameLength,SegmentHeaderLength,HasFECF)
ccsds_tm_frames(FrameLength,SecondaryHeaderLength,HasFECF)
type

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
- Protected predicates

- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.22.5 ccsds_tc_frames(FrameLength,SegmentHeaderLength,HasFECF)

- FrameLength - Fixed telecommand transfer frame length in octets.
- SegmentHeaderLength - Length in octets of the optional segment header when present.
- HasFECF - Boolean indicating if frames include the frame error control field.

CCSDS telecommand transfer frame parser and generator.

Availability:
    logtalk_load(ccsds_frames(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-09

Compilation flags:
    static, context_switching_calls

Implements:
    public ccsds_frame_protocol
Uses:

list
reader
type

Remarks:
(none)

Inherited public predicates:
data_field/2 fecf/2 frame_type/2 generate/2 generate/3 ocf/2 parse/2 spacecraft_id/2 valid/1 version/2 virtual_channel_id/2

- Public predicates
  - bypass_flag/2
  - control_command_flag/2
  - sequence_number/2
  - segment_header/2
  - update_fecf/2
  - verify_fecf/1
- Protected predicates
- Private predicates
- Operators

Public predicates

bypass_flag/2

Extracts the bypass flag value.

Compilation flags:
static

Template:
bypass_flag(Frame,Flag)
Mode and number of proofs:
bypass_flag(+compound,-integer) - one

control_command_flag/2

Extracts the control command flag value.

Compilation flags:
    static

Template:
    control_command_flag(Frame,Flag)
Mode and number of proofs:
    control_command_flag(+compound,-integer) - one

---

sequence_number/2

Extracts the frame sequence number.

Compilation flags:
    static

Template:
    sequence_number(Frame,SequenceNumber)
Mode and number of proofs:
    sequence_number(+compound,-integer) - one

---

segment_header/2

Extracts the optional segment header. Returns none when absent.

Compilation flags:
    static

Template:
    segment_header(Frame,SegmentHeader)
Mode and number of proofs:
    segment_header(+compound,-compound) - one

---

update_fecf/2

Computes the correct telecommand transfer frame FECF for the selected object configuration and returns the corresponding updated frame term.

Compilation flags:
    static

Template:
    update_fecf(Frame,UpdatedFrame)
Mode and number of proofs:
    update_fecf(+compound,-compound) - one_or_error

---

verify_fecf/1

True if the telecommand transfer frame term contains the correct FECF for the selected object configuration.

Compilation flags:
    static

Template:
    verify_fecf(Frame)
Mode and number of proofs:
    verify_fecf(@compound) - zero_or_one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.22.6 ccsds_tm_frames(FrameLength,SecondaryHeaderLength,HasFECF)

- FrameLength - Fixed telemetry transfer frame length in octets.
- SecondaryHeaderLength - Length in octets of the telemetry transfer frame secondary header when present.
- HasFECF - Boolean indicating if frames include the frame error control field.

CCSDS telemetry transfer frame parser and generator.

Availability:
logtalk_load(ccsds_frames(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-09

Compilation flags:
static, context_switching_calls

Implements:
public ccsds_frame_protocol
Uses:
list
reader
type

Remarks:
(none)

Inherited public predicates:
data_field/2 fecf/2 frame_type/2 generate/2 generate/3 ocf/2 parse/2 spacecraft_id/2 valid/1 version/2 virtual_channel_id/2

- Public predicates
  - ocf_flag/2

    - master_channel_frame_count/2
    - virtual_channel_frame_count/2
    - secondary_header_flag/2
    - synchronization_flag/2
    - packet_order_flag/2
    - segment_length_identifier/2
    - first_header_pointer/2
    - secondary_header/2
    - update_fecf/2
    - verify_fecf/1
- Protected predicates
- Private predicates
- Operators

Public predicates

ocf_flag/2

Extracts the operational control field flag. Returns absent or present.

Compilation flags:
    static

Template:
    ocf_flag(Frame,Flag)
Mode and number of proofs:
    ocf_flag(+compound,-atom) - one

---

master_channel_frame_count/2

Extracts the master channel frame count.

Compilation flags:
    static

Template:

master_channel_frame_count(Frame,Count)
Mode and number of proofs:
master_channel_frame_count(+compound,-integer) - one

---

virtual_channel_frame_count/2

Extracts the virtual channel frame count.

Compilation flags:
static

Template:
virtual_channel_frame_count(Frame,Count)
Mode and number of proofs:
virtual_channel_frame_count(+compound,-integer) - one

---

secondary_header_flag/2

Extracts the telemetry transfer frame secondary header flag. Returns absent or present.

Compilation flags:
static

Template:
secondary_header_flag(Frame,Flag)
Mode and number of proofs:
secondary_header_flag(+compound,-atom) - one

---

synchronization_flag/2

Extracts the synchronization flag value.

Compilation flags:
    static

Template:
    synchronization_flag(Frame,Flag)
Mode and number of proofs:
    synchronization_flag(+compound,-integer) - one

---

packet_order_flag/2

Extracts the packet order flag value.

Compilation flags:
    static

Template:
    packet_order_flag(Frame,Flag)
Mode and number of proofs:
    packet_order_flag(+compound,-integer) - one

---

segment_length_identifier/2

Extracts the segment length identifier value.

Compilation flags:
    static

Template:
    segment_length_identifier(Frame,Identifier)
Mode and number of proofs:
    segment_length_identifier(+compound,-integer) - one

---

first_header_pointer/2

Extracts the first header pointer value.

Compilation flags:
    static

Template:
    first_header_pointer(Frame,Pointer)
Mode and number of proofs:
    first_header_pointer(+compound,-integer) - one

---

secondary_header/2

Extracts the telemetry transfer frame secondary header. Returns none when absent.

Compilation flags:
    static

Template:
    secondary_header(Frame,SecondaryHeader)
Mode and number of proofs:
    secondary_header(+compound,-compound) - one

---

update_fecf/2

Computes the correct telemetry transfer frame FECF for the selected object configuration and returns the corresponding updated frame term.

Compilation flags:
    static

Template:
    update_fecf(Frame,UpdatedFrame)
Mode and number of proofs:
    update_fecf(+compound,-compound) - one_or_error

verify_fecf/1

True if the telemetry transfer frame term contains the correct FECF for the selected object configuration.

Compilation flags:
    static

Template:
    verify_fecf(Frame)
Mode and number of proofs:
    verify_fecf(@compound) - zero_or_one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.23 ccsds_link_profiles

object

1.23.1 ccsds_link_profiles

Ergonomic wrappers around the CCSDS TM, TC, and AOS frame objects using explicit profile terms.

Availability:
    logtalk_load(ccsds_link_profiles(loader))

Author: Paulo Moura
Version: 0:1:0

Date: 2026-05-09

Compilation flags:
static, context_switching_calls

Uses:
ccsds_packet_services

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - valid_profile/1
  - parse_frame/3
  - parse_frames/3
  - generate_frame/3
  - generate_frames/3
  - valid_reassembly_state/1
  - initial_reassembly_state/1
  - pending_fragments/2
  - valid_discontinuity_policy/1
  - extract_packets/4
  - insert_packets/5
  - reassemble_packets/6
  - reassemble_packets/7
  - reassemble_packets/8
  - reassemble_frames/6
  - reassemble_frames/7
  - reassemble_frames/8
- Protected predicates
- Private predicates
- Operators

Public predicates

valid_profile/1

True if the argument is a supported CCSDS link profile term.

Compilation flags:
    static

Template:
    valid_profile(Profile)
Mode and number of proofs:
    valid_profile(@compound) - zero_or_one

---

parse_frame/3

Parses exactly one CCSDS transfer frame from a source using a link profile term. Supported source terms are file(File), stream(Stream), and bytes(Bytes).

Compilation flags:
    static

Template:
    parse_frame(Source,Profile,Frame)
Mode and number of proofs:
    parse_frame(+compound,+compound,-compound) - one_or_error

Exceptions:
    Source is a variable:
        instantiation_error
    Profile is a variable:
        instantiation_error
    Profile is neither a variable nor a supported link profile term:
        domain_error(ccsds_link_profile,Profile)
    Source does not contain exactly one frame for the selected profile:
        domain_error(ccsds_single_frame_source,Source)

---

parse_frames/3

Parses zero or more CCSDS transfer frames from a source using a link profile term. Supported source terms are file(File), stream(Stream), and bytes(Bytes).

Compilation flags:
static

Template:
parse_frames(Source,Profile,Frames)
Mode and number of proofs:
parse_frames(+compound,+compound,-list(compound)) - one_or_error

Exceptions:
Source is a variable:
instantiation_error
Profile is a variable:
instantiation_error
Profile is neither a variable nor a supported link profile term:
domain_error(ccsds_link_profile,Profile)

---

generate_frame/3

Generates exactly one CCSDS transfer frame to a sink using a link profile term. Supported sink terms are file(File), stream(Stream), and bytes(Bytes).

Compilation flags:
static

Template:
generate_frame(Sink,Profile,Frame)
Mode and number of proofs:
generate_frame(+compound,+compound,+compound) - one_or_error

Exceptions:
Sink is a variable:
instantiation_error
Profile is a variable:
instantiation_error
Frame is a variable:

instantiation_error
Profile is neither a variable nor a supported link profile term:
domain_error(ccsds_link_profile,Profile)
Frame is neither a variable nor a valid frame term for the selected profile:
domain_error(ccsds_frame_term,Frame)

---

generate_frames/3

Generates zero or more CCSDS transfer frames to a sink using a link profile term. Supported sink terms are file(File), stream(Stream), and bytes(Bytes).

Compilation flags:
static

Template:
generate_frames(Sink,Profile,Frames)
Mode and number of proofs:
generate_frames(+compound,+compound,+list(compound)) - one_or_error

Exceptions:
Sink is a variable:
instantiation_error
Profile is a variable:
instantiation_error
Frames is a variable:
instantiation_error
Profile is neither a variable nor a supported link profile term:
domain_error(ccsds_link_profile,Profile)
Frames is neither a variable nor a list of valid frame terms for the selected profile:
domain_error(ccsds_frame_terms,Frames)

---

valid_reassembly_state/1

True if the argument is a valid TM/AOS channel packet reassembly state term for the profile-level packet reassembly predicates.

Compilation flags:
    static

Template:
    valid_reassembly_state(State)
Mode and number of proofs:
    valid_reassembly_state(@compound) - zero_or_one

---

initial_reassembly_state/1

Returns the initial TM/AOS channel packet reassembly state for the profile-level packet reassembly predicates.

Compilation flags:
    static

Template:
    initial_reassembly_state(State)
Mode and number of proofs:
    initial_reassembly_state(-compound) - one

---

pending_fragments/2

Returns the non-empty pending packet fragments currently buffered per TM or AOS virtual channel.

Compilation flags:
    static

Template:
    pending_fragments(State,PendingFragments)
Mode and number of proofs:
    pending_fragments(+compound,-list(compound)) - one_or_error

---

valid_discontinuity_policy/1

True if the argument is a valid packet reassembly discontinuity recovery policy atom. Valid values are throw, drop, and resynchronize.

Compilation flags:
    static

Template:
    valid_discontinuity_policy(Policy)
Mode and number of proofs:
    valid_discontinuity_policy(@atom) - zero_or_one

---

extract_packets/4

Extracts the TM first-header-pointer packet zone or the AOS M_PDU packet zone from a TM or AOS frame using the selected link profile and packet secondary header length.

Compilation flags:
    static

Template:
    extract_packets(Profile,SecondaryHeaderLength,Frame,PacketZone)
Mode and number of proofs:
    extract_packets(+compound,+integer,+compound,-compound) - one_or_error

Exceptions:
    Profile is a variable:
        instantiation_error
    SecondaryHeaderLength is a variable:
        instantiation_error
    Frame is a variable:
        instantiation_error
    Profile is neither a variable nor a supported link profile term:
        domain_error(ccsds_link_profile,Profile)
    Profile is a telecommand profile and therefore does not support packet extraction:
        domain_error(ccsds_packet_link_profile,Profile)
    Frame is neither a variable nor a valid frame term for the selected profile:

domain_error(ccsds_frame_term,Frame)

---

insert_packets/5

Regenerates the TM first-header-pointer packet zone or the AOS M_PDU packet zone for a TM or AOS frame using the selected link profile and packet secondary header length.

Compilation flags:
static

Template:
insert_packets(Profile,SecondaryHeaderLength,PacketZone,Frame,UpdatedFrame)
Mode and number of proofs:
insert_packets(+compound,+integer,+compound,+compound,-compound) - one_or_error

Exceptions:
Profile is a variable:
instantiation_error
SecondaryHeaderLength is a variable:
instantiation_error
PacketZone is a variable:
instantiation_error
Frame is a variable:
instantiation_error
Profile is neither a variable nor a supported link profile term:
domain_error(ccsds_link_profile,Profile)
Profile is a telecommand profile and therefore does not support packet insertion:
domain_error(ccsds_packet_link_profile,Profile)
Frame is neither a variable nor a valid frame term for the selected profile:
domain_error(ccsds_frame_term,Frame)

---

reassemble_packets/6

Reassembles complete packets from a single TM or AOS transfer frame using the default throw discontinuity recovery policy.

Compilation flags:
static

Template:
reassemble_packets(Profile,SecondaryHeaderLength,Frame,State,Packets,UpdatedState)
Mode and number of proofs:
reassemble_packets(+compound,+integer,+compound,+compound,-list(compound),-compound) - one_or_error

---

reassemble_packets/7

Reassembles complete packets from a single TM or AOS transfer frame using the selected discontinuity recovery policy.

Compilation flags:
static

Template:
reassemble_packets(Profile,SecondaryHeaderLength,Frame,Policy,State,Packets,UpdatedState)
Mode and number of proofs:
reassemble_packets(+compound,+integer,+compound,+atom,+compound,-list(compound),-compound) - one_or_error

---

reassemble_packets/8

Reassembles complete packets from a single TM or AOS transfer frame using the selected discontinuity recovery policy and returns any recovery events.

Compilation flags:
static

Template:

reassemble_packets(Profile,SecondaryHeaderLength,Frame,Policy,State,Packets,UpdatedState,
Events)

Mode and number of proofs:

reassemble_packets(+compound,+integer,+compound,+atom,+compound,-list(compound),
-compound,-list(compound)) - one_or_error

---

reassemble_frames/6

Reassembles complete packets across a sequence of TM or AOS transfer frames using the default throw discontinuity recovery policy.

Compilation flags:

static

Template:

reassemble_frames(Profile,SecondaryHeaderLength,Frames,State,Packets,UpdatedState)

Mode and number of proofs:

reassemble_frames(+compound,+integer,+list(compound),+compound,-list(compound),-compound)
- one_or_error

---

reassemble_frames/7

Reassembles complete packets across a sequence of TM or AOS transfer frames using the selected discontinuity recovery policy.

Compilation flags:

static

Template:

reassemble_frames(Profile,SecondaryHeaderLength,Frames,Policy,State,Packets,UpdatedState)

Mode and number of proofs:

reassemble_frames(+compound,+integer,+list(compound),+atom,+compound,-list(compound),
-compound) - one_or_error

---

reassemble_frames/8

Reassembles complete packets across a sequence of TM or AOS transfer frames using the selected discontinuity recovery policy and returns any recovery events.

Compilation flags:
static

Template:
reassemble_frames(Profile,SecondaryHeaderLength,Frames,Policy,State,Packets,UpdatedState,Events)

Mode and number of proofs:
reassemble_frames(+compound,+integer,+list(compound),+atom,+compound,-list(compound),-compound,-list(compound)) - one_or_error

---

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.24 ccsds_packet_services

object

1.24.1 ccsds_packet_services

Helpers for CCSDS packet service packet-zone splitting and cross-frame TM/AOS packet reassembly.

Availability:
logtalk_load(ccsds_packet_services(loader))

Author: Paulo Moura
Version: 0:1:0

Date: 2026-05-08

Compilation flags:
static, context_switching_calls

Uses:
ccsds_packets(SecondaryHeaderLength)
list
type

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - valid_reassembly_state/1
  - initial_reassembly_state/1
  - pending_data/2
  - valid_channel_reassembly_state/1
  - initial_channel_reassembly_state/1
  - pending_fragments/2
  - valid_discontinuity_policy/1
  - valid/1
  - prefix_data/2
  - packets/2
  - suffix_data/2
  - split_packet_zone/4
  - join_packet_zone/4
  - extract_tm_packets/3
  - insert_tm_packets/4
  - split_aos_packet_zone/3
  - join_aos_packet_zone/3
  - reassemble_packet_zone/5
  - extract_aos_packets/3

  - insert_aos_packets/4
  - reassemble_tm_packets/5
  - reassemble_tm_packets/6
  - reassemble_tm_packets/7
  - reassemble_tm_frames/5
  - reassemble_tm_frames/6
  - reassemble_tm_frames/7
  - reassemble_aos_packets/5
  - reassemble_aos_packets/6
  - reassemble_aos_packets/7
  - reassemble_aos_frames/5
  - reassemble_aos_frames/6
  - reassemble_aos_frames/7
- Protected predicates
- Private predicates
- Operators

Public predicates

valid_reassembly_state/1

True if the argument is a valid packet reassembly state term.

Compilation flags:
    static

Template:
    valid_reassembly_state(State)
Mode and number of proofs:
    valid_reassembly_state(@compound) - zero_or_one

initial_reassembly_state/1

Returns the initial cross-frame packet reassembly state.

Compilation flags:
    static

Template:
    initial_reassembly_state(State)
Mode and number of proofs:
    initial_reassembly_state(-compound) - one

---

pending_data/2

Extracts the currently buffered trailing packet fragment bytes from a packet reassembly state.

Compilation flags:
    static

Template:
    pending_data(State,Bytes)
Mode and number of proofs:
    pending_data(+compound,-list(byte)) - one

---

valid_channel_reassembly_state/1

True if the argument is a valid frame-channel packet reassembly state term keyed by frame type, spacecraft identifier, and virtual channel identifier.

Compilation flags:
    static

Template:
    valid_channel_reassembly_state(State)
Mode and number of proofs:
    valid_channel_reassembly_state(@compound) - zero_or_one

---

initial_channel_reassembly_state/1

Returns the initial frame-channel packet reassembly state.

Compilation flags:
    static

Template:
    initial_channel_reassembly_state(State)
Mode and number of proofs:
    initial_channel_reassembly_state(-compound) - one

---

pending_fragments/2

Extracts the non-empty pending packet fragments buffered per frame type, spacecraft identifier, and virtual channel identifier.

Compilation flags:
    static

Template:
    pending_fragments(State,PendingFragments)
Mode and number of proofs:
    pending_fragments(+compound,-list(compound)) - one

---

valid_discontinuity_policy/1

True if the argument is a valid discontinuity recovery policy atom. Valid values are throw, drop, and resynchronize.

Compilation flags:
    static

Template:

valid_discontinuity_policy(Policy)
Mode and number of proofs:
valid_discontinuity_policy(@atom) - zero_or_one

---

valid/1

True if the argument is a valid packet-zone term.

Compilation flags:
static

Template:
valid(PacketZone)
Mode and number of proofs:
valid(@compound) - zero_or_one

---

prefix_data/2

Extracts the bytes that precede the first complete packet in a packet zone.

Compilation flags:
static

Template:
prefix_data(PacketZone,PrefixData)
Mode and number of proofs:
prefix_data(+compound,-list(byte)) - one

---

packets/2

Extracts the list of complete packets in a packet zone.

Compilation flags:
    static

Template:
    packets(PacketZone,Packets)
Mode and number of proofs:
    packets(+compound,-list(compound)) - one

---

suffix_data/2

Extracts the bytes that trail the last complete packet in a packet zone.

Compilation flags:
    static

Template:
    suffix_data(PacketZone,SuffixData)
Mode and number of proofs:
    suffix_data(+compound,-list(byte)) - one

---

split_packet_zone/4

Splits a packet-zone byte sequence using the first header pointer and parses any complete packets starting at that offset. A pointer value of 2047 denotes that no packet starts in the zone.

Compilation flags:
    static

Template:
    split_packet_zone(Bytes,FirstHeaderPointer,SecondaryHeaderLength,PacketZone)
Mode and number of proofs:
    split_packet_zone(+list(byte),+integer,+integer,-compound) - one_or_error

---

join_packet_zone/4

Encodes a packet-zone term back into bytes and returns the corresponding first header pointer. When the packet list is empty, the pointer is set to 2047.

Compilation flags:
    static

Template:
    join_packet_zone(PacketZone,SecondaryHeaderLength,Bytes,FirstHeaderPointer)
Mode and number of proofs:
    join_packet_zone(+compound,+integer,-list(byte),-integer) - one_or_error

---

extract_tm_packets/3

Extracts a TM transfer frame packet zone as a packet-zone term using the frame first header pointer and the given packet secondary header length.

Compilation flags:
    static

Template:
    extract_tm_packets(Frame,SecondaryHeaderLength,PacketZone)
Mode and number of proofs:
    extract_tm_packets(+compound,+integer,-compound) - one_or_error

---

insert_tm_packets/4

Updates a TM transfer frame data field and first header pointer from a packet-zone term while preserving the remaining frame fields unchanged.

Compilation flags:
    static

Template:

insert_tm_packets(PacketZone,SecondaryHeaderLength,Frame,UpdatedFrame)
Mode and number of proofs:
insert_tm_packets(+compound,+integer,+compound,-compound) - one_or_error

---

split_aos_packet_zone/3

Splits an AOS M_PDU packet-service data field into a packet-zone term. The first two octets are interpreted as the M_PDU first header pointer field, where 2046 denotes idle data only and 2047 denotes that no packet starts in the packet zone.

Compilation flags:
static

Template:
split_aos_packet_zone(Bytes,SecondaryHeaderLength,PacketZone)
Mode and number of proofs:
split_aos_packet_zone(+list(byte),+integer,-compound) - one_or_error

---

join_aos_packet_zone/3

Encodes a packet-zone term as an AOS M_PDU packet-service data field, including the two-octet first header pointer field.

Compilation flags:
static

Template:
join_aos_packet_zone(PacketZone,SecondaryHeaderLength,Bytes)
Mode and number of proofs:
join_aos_packet_zone(+compound,+integer,-list(byte)) - one_or_error

---

reassemble_packet_zone/5

Reassembles complete packets from a packet-zone term and a prior packet reassembly state, returning any complete packets plus the updated trailing fragment state.

Compilation flags:
static

Template:
reassemble_packet_zone(PacketZone,SecondaryHeaderLength,State,Packets,UpdatedState)
Mode and number of proofs:
reassemble_packet_zone(+compound,+integer,+compound,-list(compound),-compound) - one_or_error

---

extract_aos_packets/3

Extracts an AOS transfer frame packet zone as a packet-zone term using the two-octet M_PDU header at the beginning of the frame data field.

Compilation flags:
static

Template:
extract_aos_packets(Frame,SecondaryHeaderLength,PacketZone)
Mode and number of proofs:
extract_aos_packets(+compound,+integer,-compound) - one_or_error

---

insert_aos_packets/4

Updates an AOS transfer frame data field from a packet-zone term by regenerating the M_PDU first header pointer and packet-zone bytes while preserving the remaining frame fields unchanged.

Compilation flags:
static

Template:
insert_aos_packets(PacketZone,SecondaryHeaderLength,Frame,UpdatedFrame)

Mode and number of proofs:
insert_aos_packets(+compound,+integer,+compound,-compound) - one_or_error

---

reassemble_tm_packets/5

Extracts the packet zone from a TM transfer frame, using the default throw discontinuity recovery policy, and returns the complete packets plus the updated channel reassembly state.

Compilation flags:
static

Template:
reassemble_tm_packets(Frame,SecondaryHeaderLength,State,Packets,UpdatedState)
Mode and number of proofs:
reassemble_tm_packets(+compound,+integer,+compound,-list(compound),-compound) - one_or_error

---

reassemble_tm_packets/6

Extracts the packet zone from a TM transfer frame, applies the selected discontinuity recovery policy when frame-count continuity is broken, and returns the complete packets plus the updated channel reassembly state.

Compilation flags:
static

Template:
reassemble_tm_packets(Frame,SecondaryHeaderLength,Policy,State,Packets,UpdatedState)
Mode and number of proofs:
reassemble_tm_packets(+compound,+integer,+atom,+compound,-list(compound),-compound) - one_or_error

---

reassemble_tm_packets/7

Extracts the packet zone from a TM transfer frame, applies the selected discontinuity recovery policy when frame-count continuity is broken, and returns the complete packets, updated channel reassembly state, and explicit recovery events.

Compilation flags:
static

Template:
reassemble_tm_packets(Frame,SecondaryHeaderLength,Policy,State,Packets,UpdatedState,Events)
Mode and number of proofs:
reassemble_tm_packets(+compound,+integer,+atom,+compound,-list(compound),-compound,
-list(compound)) - one_or_error

---

reassemble_tm_frames/5

Reassembles complete packets across a sequence of TM transfer frames using the default throw discontinuity recovery policy.

Compilation flags:
static

Template:
reassemble_tm_frames(Frames,SecondaryHeaderLength,State,Packets,UpdatedState)
Mode and number of proofs:
reassemble_tm_frames(+list(compound),+integer,+compound,-list(compound),-compound) -
one_or_error

---

reassemble_tm_frames/6

Reassembles complete packets across a sequence of TM transfer frames using the selected discontinuity recovery policy.

Compilation flags:
static

Template:

reassemble_tm_frames(Frames,SecondaryHeaderLength,Policy,State,Packets,UpdatedState)

Mode and number of proofs:

reassemble_tm_frames(+list(compound),+integer,+atom,+compound,-list(compound),-compound) - one_or_error

---

reassemble_tm_frames/7

Reassembles complete packets across a sequence of TM transfer frames using the selected discontinuity recovery policy and returns any recovery events in frame order.

Compilation flags:

static

Template:

reassemble_tm_frames(Frames,SecondaryHeaderLength,Policy,State,Packets,UpdatedState,Events)

Mode and number of proofs:

reassemble_tm_frames(+list(compound),+integer,+atom,+compound,-list(compound),-compound,-list(compound)) - one_or_error

---

reassemble_aos_packets/5

Extracts the packet zone from an AOS transfer frame, using the default throw discontinuity recovery policy, and returns the complete packets plus the updated channel reassembly state.

Compilation flags:

static

Template:

reassemble_aos_packets(Frame,SecondaryHeaderLength,State,Packets,UpdatedState)

Mode and number of proofs:

reassemble_aos_packets(+compound,+integer,+compound,-list(compound),-compound) - one_or_error

---

reassemble_aos_packets/6

Extracts the packet zone from an AOS transfer frame, applies the selected discontinuity recovery policy when frame-count continuity is broken, and returns the complete packets plus the updated channel reassembly state.

Compilation flags:
static

Template:
reassemble_aos_packets(Frame,SecondaryHeaderLength,Policy,State,Packets,UpdatedState)
Mode and number of proofs:
reassemble_aos_packets(+compound,+integer,+atom,+compound,-list(compound),-compound) - one_or_error

---

reassemble_aos_packets/7

Extracts the packet zone from an AOS transfer frame, applies the selected discontinuity recovery policy when frame-count continuity is broken, and returns the complete packets, updated channel reassembly state, and explicit recovery events.

Compilation flags:
static

Template:
reassemble_aos_packets(Frame,SecondaryHeaderLength,Policy,State,Packets,UpdatedState,Events)
Mode and number of proofs:
reassemble_aos_packets(+compound,+integer,+atom,+compound,-list(compound),-compound, -list(compound)) - one_or_error

---

reassemble_aos_frames/5

Reassembles complete packets across a sequence of AOS transfer frames using the default throw discontinuity recovery policy.

Compilation flags:
static

Template:
reassemble_aos_frames(Frames,SecondaryHeaderLength,State,Packets,UpdatedState)
Mode and number of proofs:
reassemble_aos_frames(+list(compound),+integer,+compound,-list(compound),-compound) - one_or_error

---

reassemble_aos_frames/6

Reassembles complete packets across a sequence of AOS transfer frames using the selected discontinuity recovery policy.

Compilation flags:
static

Template:
reassemble_aos_frames(Frames,SecondaryHeaderLength,Policy,State,Packets,UpdatedState)
Mode and number of proofs:
reassemble_aos_frames(+list(compound),+integer,+atom,+compound,-list(compound),-compound) - one_or_error

---

reassemble_aos_frames/7

Reassembles complete packets across a sequence of AOS transfer frames using the selected discontinuity recovery policy and returns any recovery events in frame order.

Compilation flags:
static

Template:
reassemble_aos_frames(Frames,SecondaryHeaderLength,Policy,State,Packets,UpdatedState,Events)
Mode and number of proofs:
reassemble_aos_frames(+list(compound),+integer,+atom,+compound,-list(compound),-compound,-list(compound)) - one_or_error

---

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.25 ccsds_packetization

object

1.25.1 ccsds_packetization

Helpers for packetizing CCSDS space packets into TM and AOS service-data regions with cross-frame carryover and idle fill generation.

Availability:
logtalk_load(ccsds_packetization(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-09

Compilation flags:
static, context_switching_calls

Uses:
ccsds_packets(SecondaryHeaderLength)
list
type

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - initial_state/1
  - pending_packets/2
  - packetize_tm_packets/6
  - packetize_tm_packets/7
  - packetize_aos_packets/6
  - packetize_aos_packets/7
  - packetize_tm_frames/7
  - packetize_tm_frames/8
  - packetize_aos_frames/7
  - packetize_aos_frames/8
  - generate_idle_packet/4
- Protected predicates
- Private predicates
- Operators

Public predicates

initial_state/1

Returns the initial packetizer state.

Compilation flags:
    static

Template:
    initial_state(State)
Mode and number of proofs:
    initial_state(-compound) - one

pending_packets/2

Extracts the non-empty queued packets and pending trailing packet bytes buffered per frame type, spacecraft identifier, and virtual channel identifier.

Compilation flags:
    static

Template:
    pending_packets(State,PendingPackets)
Mode and number of proofs:
    pending_packets(+compound,-list(compound)) - one_or_error

---

packetize_tm_packets/6

Packetizes packets into the packet service region of a TM transfer frame, preserving any still-pending trailing packet bytes and queuing packets that do not fully fit.

Compilation flags:
    static

Template:
    packetize_tm_packets(Frame,SecondaryHeaderLength,State,Packets,UpdatedFrame,UpdatedState)
Mode and number of proofs:
    packetize_tm_packets(+compound,+integer,+compound,+list(compound),-compound,-compound) - one_or_error

---

packetize_tm_packets/7

Packetizes packets into the packet service region of a TM transfer frame and also returns explicit packetization events.

Compilation flags:
    static

Template:
    packetize_tm_packets(Frame,SecondaryHeaderLength,State,Packets,UpdatedFrame,UpdatedState,Events)

Mode and number of proofs:
packetize_tm_packets(+compound,+integer,+compound,+list(compound),-compound,-compound,-list(compound)) - one_or_error

---

packetize_aos_packets/6

Packetizes packets into the packet service region of an AOS transfer frame, preserving any still-pending trailing packet bytes and queuing packets that do not fully fit.

Compilation flags:
static

Template:
packetize_aos_packets(Frame,SecondaryHeaderLength,State,Packets,UpdatedFrame,UpdatedState)
Mode and number of proofs:
packetize_aos_packets(+compound,+integer,+compound,+list(compound),-compound,-compound) - one_or_error

---

packetize_aos_packets/7

Packetizes packets into the packet service region of an AOS transfer frame and also returns explicit packetization events.

Compilation flags:
static

Template:
packetize_aos_packets(Frame,SecondaryHeaderLength,State,Packets,UpdatedFrame,UpdatedState,Events)
Mode and number of proofs:
packetize_aos_packets(+compound,+integer,+compound,+list(compound),-compound,-compound,-list(compound)) - one_or_error

---

packetize_tm_frames/7

Packetizes packets across a sequence of TM transfer frames. Packets that remain queued for the first frame channel are also returned.

Compilation flags:
static

Template:
packetize_tm_frames(Frames,SecondaryHeaderLength,State,Packets,UpdatedFrames,
RemainingPackets,UpdatedState)

Mode and number of proofs:
packetize_tm_frames(+list(compound),+integer,+compound,+list(compound),-list(compound),
-list(compound),-compound) - one_or_error

---

packetize_tm_frames/8

Packetizes packets across a sequence of TM transfer frames and also returns explicit packetization events in frame order.

Compilation flags:
static

Template:
packetize_tm_frames(Frames,SecondaryHeaderLength,State,Packets,UpdatedFrames,
RemainingPackets,UpdatedState,Events)

Mode and number of proofs:
packetize_tm_frames(+list(compound),+integer,+compound,+list(compound),-list(compound),
-list(compound),-compound,-list(compound)) - one_or_error

---

packetize_aos_frames/7

Packetizes packets across a sequence of AOS transfer frames. Packets that remain queued for the first frame channel are also returned.

Compilation flags:
static

Template:
packetize_aos_frames(Frames,SecondaryHeaderLength,State,Packets,UpdatedFrames,
RemainingPackets,UpdatedState)
Mode and number of proofs:
packetize_aos_frames(+list(compound),+integer,+compound,+list(compound),-list(compound),
-list(compound),-compound) - one_or_error

---

packetize_aos_frames/8

Packetizes packets across a sequence of AOS transfer frames and also returns explicit packetization events in frame order.

Compilation flags:
static

Template:
packetize_aos_frames(Frames,SecondaryHeaderLength,State,Packets,UpdatedFrames,
RemainingPackets,UpdatedState,Events)
Mode and number of proofs:
packetize_aos_frames(+list(compound),+integer,+compound,+list(compound),-list(compound),
-list(compound),-compound,-list(compound)) - one_or_error

---

generate_idle_packet/4

Generates a telemetry idle packet using APID 2047, the given packet secondary header length, sequence count, and user-data length.

Compilation flags:
static

Template:
generate_idle_packet(SecondaryHeaderLength,SequenceCount,UserDataLength,Packet)
Mode and number of proofs:
generate_idle_packet(+integer,+integer,+integer,-compound) - one_or_error

---

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.26 ccsds_packets

object

1.26.1 ccsds_packets

CCSDS Space Packet parser with no secondary header parsing. For secondary header support, use ccsds_packets(Length) where Length is the secondary header size in bytes.

Availability:
logtalk_load(ccsds_packets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2025-12-05

Compilation flags:
static, context_switching_calls

Extends:
public ccsds_packets(0)

Remarks:
(none)

Inherited public predicates:
apid/2 data_length/2 generate/2 generate/3 parse/2 secondary_header/2 secondary_header_flag/2 secondary_header_time/3 sequence_count/2 sequence_flags/2 type/2 user_data/2 version/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.26.2 ccsds_packets(SecondaryHeaderLength)

- SecondaryHeaderLength - Length in bytes of the secondary header when present (0 for no secondary header parsing, or a positive integer).

CCSDS Space Packet parser following the CCSDS 133.0-B-2 standard. Parses binary packet data including optional secondary headers.

Availability:
logtalk_load(ccsds_packets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-09

Compilation flags:
static, context_switching_calls

Uses:

ccsds_time_fields
list
reader

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - parse/2
  - generate/2
  - generate/3
  - version/2
  - type/2
  - secondary_header_flag/2
  - apid/2
  - sequence_flags/2
  - sequence_count/2
  - data_length/2
  - user_data/2
  - secondary_header/2
  - secondary_header_time/3
- Protected predicates
- Private predicates
- Operators

Public predicates

parse/2

Parses CCSDS packet(s) from a source into a list of packet terms. The source can be file(File), stream(Stream), or bytes(Bytes).

Compilation flags:
static

Template:
parse(Source,Packets)
Mode and number of proofs:
parse(+compound,-list(compound)) - one_or_error

Exceptions:
Source is a variable:
instantiation_error
Source is neither a variable nor a valid source:
domain_error(ccsds_source,Source)
Source is a valid source but the data cannot be parsed as a CCSDS packet:
domain_error(ccsds_byte_sequence,Bytes)

---

generate/2

Generates CCSDS packet bytes to a sink from a list of packet terms. The sink can be file(File), stream(Stream), or bytes(Bytes). For file(File) and stream(Stream), writes to the binary file or stream. For bytes(Bytes), unifies Bytes with the generated byte list.

Compilation flags:
static

Template:
generate(Sink,Packets)
Mode and number of proofs:
generate(+compound,+list(compound)) - one_or_error

Exceptions:
Sink is a variable:
instantiation_error
Sink is neither a variable nor a valid sink:
domain_error(ccsds_sink,Sink)
Packets is a partial list or a list with an element Packet which is a variable:
instantiation_error
An element Packet of the list Packets is neither a variable nor a valid CCSDS packet term:
domain_error(ccsds_packet_term,Packet)

---

generate/3

Generates a list of bytes from a CCSDS packet term with an open tail. Mainly used when generating arbitrary CCSDS packets.

Compilation flags:
    static

Template:
    generate(Packet,Bytes,Tail)
Mode and number of proofs:
    generate(+compound,-list(byte),--variable) - one_or_error

Exceptions:
    Packet is a variable:
        instantiation_error
    Packet is neither a variable nor a valid CCSDS packet term:
        domain_error(ccsds_packet_term,Packet)

---

version/2

Extracts the version number from a packet (always 0 for CCSDS Space Packets).

Compilation flags:
    static

Template:
    version(Packet,Version)
Mode and number of proofs:
    version(+compound,-integer) - one

---

type/2

Extracts the packet type from a packet. Returns telemetry or telecommand.

Compilation flags:
    static

Template:
    type(Packet,Type)
Mode and number of proofs:
    type(+compound,-atom) - one

---

secondary_header_flag/2

Extracts the secondary header flag. Returns absent or present.

Compilation flags:
    static

Template:
    secondary_header_flag(Packet,Flag)
Mode and number of proofs:
    secondary_header_flag(+compound,-atom) - one

---

apid/2

Extracts the Application Process Identifier (APID) from a packet.

Compilation flags:
    static

Template:
    apid(Packet,APID)
Mode and number of proofs:
    apid(+compound,-integer) - one

---

sequence_flags/2

Extracts the sequence flags. Returns continuation, first, last, or standalone.

Compilation flags:
    static

Template:
    sequence_flags(Packet,Flags)
Mode and number of proofs:
    sequence_flags(+compound,-atom) - one

---

sequence_count/2

Extracts the packet sequence count (0-16383).

Compilation flags:
    static

Template:
    sequence_count(Packet,Count)
Mode and number of proofs:
    sequence_count(+compound,-integer) - one

---

data_length/2

Extracts the packet data length field value.

Compilation flags:
    static

Template:
    data_length(Packet,Length)
Mode and number of proofs:
    data_length(+compound,-integer) - one

---

user_data/2

Extracts the user data field as a list of bytes.

Compilation flags:
    static

Template:
    user_data(Packet,Data)
Mode and number of proofs:
    user_data(+compound,-list(byte)) - one

---

secondary_header/2

Extracts the secondary header. Returns none if not present, or secondary_header(Bytes) with the raw bytes.

Compilation flags:
    static

Template:
    secondary_header(Packet,SecondaryHeader)
Mode and number of proofs:
    secondary_header(+compound,-compound) - one

---

secondary_header_time/3

Parses time from a packet secondary header. When Descriptor is a variable, the secondary header is parsed as a self-describing CCSDS time field. When Descriptor is instantiated, it is used to parse the raw T-field bytes.

Compilation flags:
    static

Template:
    secondary_header_time(Packet,Descriptor,TimeCode)
Mode and number of proofs:
    secondary_header_time(+compound,?compound,-compound) - zero_or_one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.26.3 ccsds_packets_types

Type definitions and arbitrary generators for CCSDS packets.

Availability:
logtalk_load(ccsds_packets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-08

Compilation flags:
static

Provides:
type::type/1
type::check/2
arbitrary::arbitrary/1
arbitrary::arbitrary/2
Uses:
ccsds_packets(SecondaryHeaderLength)
type

Remarks:
(none)

Inherited public predicates:

(none)

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.27 ccsds_tc_services

object

1.27.1 ccsds_tc_services

Helpers for CCSDS telecommand segment extraction, insertion, and cross-frame TC service reassembly.

Availability:
logtalk_load(ccsds_tc_services(loader))

Author: Paulo Moura
Version: 0:1:0
Date: 2026-05-08

Compilation flags:
static, context_switching_calls

Uses:
    list
    type

Remarks:
    (none)

Inherited public predicates:
    (none)

- Public predicates
  - valid_segment/1
  - sequence_flags/2
  - map_id/2
  - valid_map_dispatch/1
  - dispatch_map_id/2
  - dispatch_service_units/2
  - segment_header_suffix/2
  - valid_reassembled_segment/1
  - segment_header_suffixes/2
  - segment_data/2
  - valid_reassembly_state/1
  - initial_reassembly_state/1
  - pending_fragments/2
  - valid_discontinuity_policy/1
  - extract_tc_segment/2
  - insert_tc_segment/3
  - reassemble_tc_frame/4
  - reassemble_tc_frame/5
  - reassemble_tc_frame/6
  - reassemble_tc_frames/4
  - reassemble_tc_frames/5
  - reassemble_tc_frames/6
  - reassemble_tc_frame_with_provenance/5
  - reassemble_tc_frame_with_provenance/6

  - reassemble_tc_frames_with_provenance/5
  - reassemble_tc_frames_with_provenance/6
  - dispatch_service_units_by_map/2
  - dispatch_service_units_by_map/3
- Protected predicates
- Private predicates
- Operators

Public predicates

valid_segment/1

True if the argument is a valid CCSDS telecommand segment term.

Compilation flags:
    static

Template:
    valid_segment(Segment)
Mode and number of proofs:
    valid_segment(@compound) - zero_or_one

---

sequence_flags/2

Extracts the telecommand segment sequence flags.

Compilation flags:
    static

Template:
    sequence_flags(Segment,SequenceFlags)
Mode and number of proofs:
    sequence_flags(+compound,-atom) - one

---

map_id/2

Extracts the telecommand service-unit MAP identifier from either a telecommand segment term or a provenance-aware reassembled telecommand service-unit term.

Compilation flags:
    static

Template:
    map_id(ServiceUnit,MapId)
Mode and number of proofs:
    map_id(+compound,-integer) - one

---

valid_map_dispatch/1

True if the argument is a valid MAP-aware dispatch term grouping complete telecommand service units for a single MAP identifier.

Compilation flags:
    static

Template:
    valid_map_dispatch(Dispatch)
Mode and number of proofs:
    valid_map_dispatch(@compound) - zero_or_one

---

dispatch_map_id/2

Extracts the MAP identifier from a MAP-aware dispatch term.

Compilation flags:
    static

Template:
    dispatch_map_id(Dispatch,MapId)
Mode and number of proofs:
    dispatch_map_id(+compound,-integer) - one

---

dispatch_service_units/2

Extracts the complete telecommand service units from a MAP-aware dispatch term.

Compilation flags:
    static

Template:
    dispatch_service_units(Dispatch,ServiceUnits)
Mode and number of proofs:
    dispatch_service_units(+compound,-list(compound)) - one

---

segment_header_suffix/2

Extracts the mission-specific segment-header suffix bytes. Returns the empty list when the segment has no mission-specific suffix.

Compilation flags:
    static

Template:
    segment_header_suffix(Segment,HeaderSuffix)
Mode and number of proofs:
    segment_header_suffix(+compound,-list(byte)) - one

---

valid_reassembled_segment/1

True if the argument is a valid provenance-aware reassembled telecommand service-unit term.

Compilation flags:
    static

Template:
    valid_reassembled_segment(ReassembledSegment)

Mode and number of proofs:
valid_reassembled_segment(@compound) - zero_or_one

---

segment_header_suffixes/2

Extracts the ordered list of per-frame mission-specific segment-header suffixes that contributed to a provenance-aware reassembled telecommand service-unit term.

Compilation flags:
static

Template:
segment_header_suffixes(ReassembledSegment,HeaderSuffixes)
Mode and number of proofs:
segment_header_suffixes(+compound,-list(list(byte))) - one

---

segment_data/2

Extracts the telecommand segment data bytes.

Compilation flags:
static

Template:
segment_data(Segment,Data)
Mode and number of proofs:
segment_data(+compound,-list(byte)) - one

---

valid_reassembly_state/1

True if the argument is a valid telecommand segment reassembly state term.

Compilation flags:
    static

Template:
    valid_reassembly_state(State)
Mode and number of proofs:
    valid_reassembly_state(@compound) - zero_or_one

---

initial_reassembly_state/1

Returns the initial telecommand segment reassembly state.

Compilation flags:
    static

Template:
    initial_reassembly_state(State)
Mode and number of proofs:
    initial_reassembly_state(-compound) - one

---

pending_fragments/2

Extracts the pending segmented telecommand data fragments keyed by spacecraft, virtual channel, and MAP identifier.

Compilation flags:
    static

Template:
    pending_fragments(State,PendingFragments)
Mode and number of proofs:
    pending_fragments(+compound,-list(compound)) - one

---

valid_discontinuity_policy/1

True if the argument is a valid discontinuity recovery policy atom. Valid values are throw, drop, and resynchronize.

Compilation flags:
    static

Template:
    valid_discontinuity_policy(Policy)
Mode and number of proofs:
    valid_discontinuity_policy(@atom) - zero_or_one

---

extract_tc_segment/2

Extracts a telecommand segment from a telecommand transfer frame using the representation tc_segment(SequenceFlags, MapId, HeaderSuffix, Data). When mission-specific segment headers use multiple octets, only the first octet is interpreted for the standard MAP and sequence semantics and the remaining octets are returned as HeaderSuffix.

Compilation flags:
    static

Template:
    extract_tc_segment(Frame,Segment)
Mode and number of proofs:
    extract_tc_segment(+compound,-compound) - one_or_error

---

insert_tc_segment/3

Updates a telecommand transfer frame from a telecommand segment term by encoding the standard MAP and sequence semantics in the first segment-header octet and preserving any remaining mission-specific segment-header octets.

Compilation flags:

static

Template:
insert_tc_segment(Segment,Frame,UpdatedFrame)
Mode and number of proofs:
insert_tc_segment(+compound,+compound,-compound) - one_or_error

---

reassemble_tc_frame/4

Reassembles complete telecommand service units from a single telecommand transfer frame using the default throw discontinuity recovery policy.

Compilation flags:
static

Template:
reassemble_tc_frame(Frame,State,Segments,UpdatedState)
Mode and number of proofs:
reassemble_tc_frame(+compound,+compound,-list(compound),-compound) - one_or_error

---

reassemble_tc_frame/5

Reassembles complete telecommand service units from a single telecommand transfer frame using the selected discontinuity recovery policy.

Compilation flags:
static

Template:
reassemble_tc_frame(Frame,Policy,State,Segments,UpdatedState)
Mode and number of proofs:
reassemble_tc_frame(+compound,+atom,+compound,-list(compound),-compound) - one_or_error

---

reassemble_tc_frame/6

Reassembles complete telecommand service units from a single telecommand transfer frame using the selected discontinuity recovery policy and returns any recovery events.

Compilation flags:
static

Template:
reassemble_tc_frame(Frame,Policy,State,Segments,UpdatedState,Events)
Mode and number of proofs:
reassemble_tc_frame(+compound,+atom,+compound,-list(compound),-compound,-list(compound)) - one_or_error

---

reassemble_tc_frames/4

Reassembles complete telecommand service units across a sequence of telecommand transfer frames using the default throw discontinuity recovery policy.

Compilation flags:
static

Template:
reassemble_tc_frames(Frames,State,Segments,UpdatedState)
Mode and number of proofs:
reassemble_tc_frames(+list(compound),+compound,-list(compound),-compound) - one_or_error

---

reassemble_tc_frames/5

Reassembles complete telecommand service units across a sequence of telecommand transfer frames using the selected discontinuity recovery policy.

Compilation flags:
static

Template:
reassemble_tc_frames(Frames,Policy,State,Segments,UpdatedState)

Mode and number of proofs:
reassemble_tc_frames(+list(compound),+atom,+compound,-list(compound),-compound) - one_or_error

---

reassemble_tc_frames/6

Reassembles complete telecommand service units across a sequence of telecommand transfer frames using the selected discontinuity recovery policy and returns any recovery events in frame order.

Compilation flags:
static

Template:
reassemble_tc_frames(Frames,Policy,State,Segments,UpdatedState,Events)
Mode and number of proofs:
reassemble_tc_frames(+list(compound),+atom,+compound,-list(compound),-compound, -list(compound)) - one_or_error

---

reassemble_tc_frame_with_provenance/5

Reassembles complete telecommand service units from a single telecommand transfer frame using the default throw discontinuity recovery policy and returns provenance-aware reassembled service-unit terms.

Compilation flags:
static

Template:
reassemble_tc_frame_with_provenance(Frame,State,ReassembledSegments,UpdatedState,Events)
Mode and number of proofs:
reassemble_tc_frame_with_provenance(+compound,+compound,-list(compound),-compound, -list(compound)) - one_or_error

---

reassemble_tc_frame_with_provenance/6

Reassembles complete telecommand service units from a single telecommand transfer frame using the selected discontinuity recovery policy and returns provenance-aware reassembled service-unit terms plus any recovery events.

Compilation flags:
    static

Template:
    reassemble_tc_frame_with_provenance(Frame,Policy,State,ReassembledSegments,UpdatedState,Events)
Mode and number of proofs:
    reassemble_tc_frame_with_provenance(+compound,+atom,+compound,-list(compound),-compound,-list(compound)) - one_or_error

---

reassemble_tc_frames_with_provenance/5

Reassembles complete telecommand service units across a sequence of telecommand transfer frames using the default throw discontinuity recovery policy and returns provenance-aware reassembled service-unit terms.

Compilation flags:
    static

Template:
    reassemble_tc_frames_with_provenance(Frames,State,ReassembledSegments,UpdatedState,Events)
Mode and number of proofs:
    reassemble_tc_frames_with_provenance(+list(compound),+compound,-list(compound),-compound,-list(compound)) - one_or_error

---

reassemble_tc_frames_with_provenance/6

Reassembles complete telecommand service units across a sequence of telecommand transfer frames using the selected discontinuity recovery policy and returns provenance-aware reassembled service-unit terms plus any recovery events in frame order.

Compilation flags:
    static

Template:
reassemble_tc_frames_with_provenance(Frames,Policy,State,ReassembledSegments,UpdatedState,Events)
Mode and number of proofs:
reassemble_tc_frames_with_provenance(+list(compound),+atom,+compound,-list(compound),-compound,-list(compound)) - one_or_error

---

dispatch_service_units_by_map/2

Groups complete telecommand service-unit terms by MAP identifier, preserving the original service-unit order within each MAP bucket and the first-seen order of MAP buckets. Accepts both telecommand segment terms and provenance-aware reassembled telecommand service-unit terms.

Compilation flags:
static

Template:
dispatch_service_units_by_map(ServiceUnits,Dispatches)
Mode and number of proofs:
dispatch_service_units_by_map(+list(compound),-list(compound)) - one_or_error

---

dispatch_service_units_by_map/3

Extracts the complete telecommand service units for a specific MAP identifier from a list of complete telecommand service-unit terms.

Compilation flags:
static

Template:
dispatch_service_units_by_map(ServiceUnits,MapId,DispatchedServiceUnits)
Mode and number of proofs:
dispatch_service_units_by_map(+list(compound),+integer,-list(compound)) - one_or_error

---

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.28 ccsds_time_codes

object

1.28.1 ccsds_ccs(CalendarVariant,FractionOctets)

- CalendarVariant - Calendar representation. Supported values are calendar and day_of_year.
- FractionOctets - Number of BCD fraction octets. Supported values are non-negative integers.

CCSDS calendar segmented time code parser and generator.

Availability:
logtalk_load(ccsds_time_codes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-08

Compilation flags:
static, context_switching_calls

Implements:
public ccsds_time_code_protocol
Uses:
reader
type

Remarks:
(none)

Inherited public predicates:
epoch/1 format/1 from_unix_seconds/2 generate/2 generate/3 parse/2 unix_seconds/2 valid/1

- Public predicates
  - year/2
  - day_of_year/2
  - month/2
  - day/2
  - time_of_day/4
- Protected predicates
- Private predicates
- Operators

Public predicates

year/2

Extracts the year value from a CCS time term.

Compilation flags:
static

Template:
year(TimeCode,Year)
Mode and number of proofs:
year(+compound,-integer) - one

---

day_of_year/2

Extracts the day of year from an ordinal CCS time term.

Compilation flags:
static

Template:
day_of_year(TimeCode,DayOfYear)

Mode and number of proofs:
day_of_year(+compound,-integer) - zero_or_one

---

month/2

Extracts the month from a calendar CCS time term.

Compilation flags:
static

Template:
month(TimeCode,Month)
Mode and number of proofs:
month(+compound,-integer) - zero_or_one

---

day/2

Extracts the day of month from a calendar CCS time term.

Compilation flags:
static

Template:
day(TimeCode,Day)
Mode and number of proofs:
day(+compound,-integer) - zero_or_one

---

time_of_day/4

Extracts the hour, minute, and seconds-with-fraction from a CCS time term.

Compilation flags:
static

Template:
time_of_day(TimeCode,Hour,Minute,Seconds)
Mode and number of proofs:
time_of_day(+compound,-integer,-integer,-number) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.28.2 ccsds_cds(DaySegmentOctets,SubmillisecondOctets,Epoch)

- DaySegmentOctets - Number of day segment octets. Supported values are 2 and 3.
- SubmillisecondOctets - Number of submillisecond octets. Supported values are 0, 2, and 4.
- Epoch - Epoch used when converting to and from Unix seconds. Supported values are ccsds_epoch and unix_epoch.

CCSDS day segmented time code parser and generator.

Availability:
logtalk_load(ccsds_time_codes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-09

Compilation flags:
static, context_switching_calls

Implements:
public ccsds_time_code_protocol
Uses:
reader

type

Remarks:
(none)

Inherited public predicates:
epoch/1 format/1 from_unix_seconds/2 generate/2 generate/3 parse/2 unix_seconds/2 valid/1

- Public predicates
  - days/2
  - milliseconds/2
  - submilliseconds/2
- Protected predicates
- Private predicates
- Operators

Public predicates

days/2

Extracts the day segment value from a CDS time term.

Compilation flags:
static

Template:
days(TimeCode,Days)
Mode and number of proofs:
days(+compound,-integer) - one

milliseconds/2

Extracts the millisecond-of-day value from a CDS time term.

Compilation flags:
    static

Template:
    milliseconds(TimeCode,Milliseconds)
Mode and number of proofs:
    milliseconds(+compound,-integer) - one

---

submilliseconds/2

Extracts the optional submillisecond value from a CDS time term. Two-octet objects use microseconds; four-octet objects use picoseconds within the millisecond.

Compilation flags:
    static

Template:
    submilliseconds(TimeCode,Submilliseconds)
Mode and number of proofs:
    submilliseconds(+compound,-integer) - zero_or_one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.28.3 ccsds_cuc(CoarseOctets,FineOctets,Epoch)

- CoarseOctets - Number of coarse time octets.
- FineOctets - Number of fine time octets.
- Epoch - Epoch used when converting to and from Unix seconds. Supported values are ccsds_epoch and unix_epoch.

CCSDS unsegmented time code parser and generator.

Availability:
logtalk_load(ccsds_time_codes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-08

Compilation flags:
static, context_switching_calls

Implements:
public ccsds_time_code_protocol
Uses:
list
reader

Remarks:
(none)

Inherited public predicates:
epoch/1 format/1 from_unix_seconds/2 generate/2 generate/3 parse/2 unix_seconds/2 valid/1

- Public predicates
  - coarse/2
  - fine/2
- Protected predicates

- Private predicates
- Operators

Public predicates

coarse/2

Extracts the coarse field from a CUC time term.

Compilation flags:
    static

Template:
    coarse(TimeCode,Coarse)
Mode and number of proofs:
    coarse(+compound,-integer) - one

---

fine/2

Extracts the fine field from a CUC time term.

Compilation flags:
    static

Template:
    fine(TimeCode,Fine)
Mode and number of proofs:
    fine(+compound,-integer) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.28.4 ccsds_time_code_protocol

CCSDS time code parser and generator protocol.

Availability:
logtalk_load(ccsds_time_codes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-08

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - parse/2
  - generate/2

  - generate/3
  - valid/1
  - format/1
  - epoch/1
  - unix_seconds/2
  - from_unix_seconds/2
- Protected predicates
- Private predicates
- Operators

Public predicates

parse/2

Parses a CCSDS time code from a source term. Supported source terms are file(File), stream(Stream), and bytes(Bytes).

Compilation flags:
    static

Template:
    parse(Source,TimeCode)
Mode and number of proofs:
    parse(+compound,-compound) - one_or_error

Exceptions:
    Source is a variable:
        instantiation_error
    Source is neither a variable nor a valid source:
        domain_error(ccsds_time_code_source,Source)
    Source contents are not a valid encoding for the selected time code object:
        domain_error(ccsds_time_code_byte_sequence,Bytes)

generate/2

Generates a CCSDS time code to a sink term. Supported sink terms are file(File), stream(Stream), and bytes(Bytes).

Compilation flags:
    static

Template:
    generate(Sink,TimeCode)
Mode and number of proofs:
    generate(+compound,+compound) - one_or_error

Exceptions:
    Sink is a variable:
        instantiation_error
    Sink is neither a variable nor a valid sink:
        domain_error(ccsds_time_code_sink,Sink)
    TimeCode is neither a variable nor a valid time code term for the selected object:
        domain_error(ccsds_time_code_term,TimeCode)

---

generate/3

Generates a list of bytes from a CCSDS time code term with an open tail. Mainly used when generating arbitrary values.

Compilation flags:
    static

Template:
    generate(TimeCode,Bytes,Tail)
Mode and number of proofs:
    generate(+compound,-list(byte),--variable) - one_or_error

Exceptions:
    TimeCode is a variable:
        instantiation_error
    TimeCode is neither a variable nor a valid time code term for the selected object:
        domain_error(ccsds_time_code_term,TimeCode)

---

valid/1

True if the argument is a valid time code term for the selected object.

Compilation flags:
    static

Template:
    valid(TimeCode)
Mode and number of proofs:
    valid(@compound) - zero_or_one

---

format/1

Returns the CCSDS time code format handled by the object.

Compilation flags:
    static

Template:
    format(Format)
Mode and number of proofs:
    format(-atom) - one

---

epoch/1

Returns the epoch used by the object.

Compilation flags:
    static

Template:
    epoch(Epoch)
Mode and number of proofs:

epoch(-atom) - one

---

unix_seconds/2

Converts a time code term into a Unix timestamp in seconds.

Compilation flags:
static

Template:
unix_seconds(TimeCode,Seconds)
Mode and number of proofs:
unix_seconds(+compound,-number) - one_or_error

Exceptions:
TimeCode is a variable:
instantiation_error
TimeCode is neither a variable nor a valid time code term for the selected object:
domain_error(ccsds_time_code_term,TimeCode)

---

from_unix_seconds/2

Converts a Unix timestamp in seconds into a time code term for the selected object.

Compilation flags:
static

Template:
from_unix_seconds(Seconds,TimeCode)
Mode and number of proofs:
from_unix_seconds(+number,-compound) - one_or_error

Exceptions:
Seconds is a variable:
instantiation_error
Seconds cannot be represented using the selected object:
domain_error(ccsds_time_code_unix_seconds,Seconds)

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

object

1.28.5 ccsds_time_codes

Helpers for introspecting CCSDS time code terms.

Availability:
logtalk_load(ccsds_time_codes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-08

Compilation flags:
static, context_switching_calls

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - valid/1
  - format/2
- Protected predicates
- Private predicates
- Operators

Public predicates

valid/1

True if the argument is a known CCSDS time code term.

Compilation flags:
    static

Template:
    valid(TimeCode)
Mode and number of proofs:
    valid(@compound) - zero_or_one

---

format/2

Returns the CCSDS time code format for a known time code term.

Compilation flags:
    static

Template:
    format(TimeCode,Format)
Mode and number of proofs:
    format(+compound,-atom) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

category

1.28.6 ccsds_time_codes_types

Type definitions and arbitrary generators for CCSDS time codes.

Availability:
logtalk_load(ccsds_time_codes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-08

Compilation flags:
static

Provides:
type::type/1
type::check/2
arbitrary::arbitrary/1
arbitrary::arbitrary/2
Uses:
ccsds_ccs(CalendarVariant,FractionOctets)
ccsds_cds(DaySegmentOctets,SubmillisecondOctets,Epoch)
ccsds_cuc(CoarseOctets,FineOctets,Epoch)
type

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.29 ccsds_time_fields

object

1.29.1 ccsds_time_fields

Helpers for parsing and generating self-describing CCSDS time fields using descriptor terms and the existing CCSDS time-code objects.

Availability:
logtalk_load(ccsds_time_fields(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-09

Compilation flags:
static, context_switching_calls

Uses:
reader
type

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - valid_descriptor/1
  - parse/3
  - generate/3
  - format/2
  - epoch/2
- Protected predicates
- Private predicates
- Operators

Public predicates

valid_descriptor/1

True if the argument is a supported CCSDS time-field descriptor term.

Compilation flags:
static

Template:
valid_descriptor(Descriptor)
Mode and number of proofs:
valid_descriptor(@compound) - zero_or_one

parse/3

Parses a self-describing CCSDS time field from a source term into a descriptor term and a time-code term. Supported source terms are file(File), stream(Stream), and bytes(Bytes).

Compilation flags:
static

Template:
parse(Source,Descriptor,TimeCode)
Mode and number of proofs:
parse(+compound,?compound,-compound) - one_or_error

Exceptions:
Source is a variable:
instantiation_error
Source is neither a variable nor a valid source:
domain_error(ccsds_time_field_source,Source)
Source contents are not a supported CCSDS time field:
domain_error(ccsds_time_field_byte_sequence,Bytes)

---

generate/3

Generates a self-describing CCSDS time field to a sink term from a descriptor term and a time-code term. Supported sink terms are file(File), stream(Stream), and bytes(Bytes).

Compilation flags:
static

Template:
generate(Sink,Descriptor,TimeCode)
Mode and number of proofs:
generate(+compound,+compound,+compound) - one_or_error

Exceptions:
Sink is a variable:
instantiation_error
Descriptor is a variable:
instantiation_error
TimeCode is a variable:

instantiation_error
Sink is neither a variable nor a valid sink:
domain_error(ccsds_time_field_sink,Sink)
Descriptor is neither a variable nor a supported descriptor term:
domain_error(ccsds_time_field_descriptor,Descriptor)
TimeCode is neither a variable nor a valid time code term for the selected descriptor:
domain_error(ccsds_time_code_term,TimeCode)

---

format/2

Returns the CCSDS time-code format selected by a descriptor term.

Compilation flags:
static

Template:
format(Descriptor,Format)
Mode and number of proofs:
format(+compound,-atom) - one_or_error

---

epoch/2

Returns the epoch selected by a descriptor term. Calendar segmented descriptors return none.

Compilation flags:
static

Template:
epoch(Descriptor,Epoch)
Mode and number of proofs:
epoch(+compound,-atom) - one_or_error

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.30 character_sets

category

1.30.1 character_set

Shared implementation support category for character set objects.

Availability:
logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
static

Implements:
public character_set_protocol

Remarks:
(none)

Inherited public predicates:
alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
  - valid_unicode_scalar/1
  - continuation_byte/1
  - high_surrogate/1
  - low_surrogate/1
  - word_bytes/4
  - dword_bytes/6
  - bytes_word/4
  - bytes_dword/6
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

valid_unicode_scalar/1

True if the argument is a valid Unicode scalar value.

Compilation flags:
    static

Template:
    valid_unicode_scalar(Code)
Mode and number of proofs:
    valid_unicode_scalar(+integer) - zero_or_one

continuation_byte/1

True if the argument is a valid UTF continuation byte.

Compilation flags:
    static

Template:
    continuation_byte(Byte)
Mode and number of proofs:
    continuation_byte(+integer) - zero_or_one

---

high_surrogate/1

True if the argument is a UTF-16 high surrogate code point.

Compilation flags:
    static

Template:
    high_surrogate(Code)
Mode and number of proofs:
    high_surrogate(+integer) - zero_or_one

---

low_surrogate/1

True if the argument is a UTF-16 low surrogate code point.

Compilation flags:
    static

Template:
    low_surrogate(Code)
Mode and number of proofs:
    low_surrogate(+integer) - zero_or_one

---

word_bytes/4

Converts a 16-bit word into two bytes using the given byte order.

Compilation flags:
    static

Template:
    word_bytes(Endian,Word,Byte1,Byte2)
Mode and number of proofs:
    word_bytes(+atom,+integer,-integer,-integer) - one

---

dword_bytes/6

Converts a 32-bit word into four bytes using the given byte order.

Compilation flags:
    static

Template:
    dword_bytes(Endian,Word,Byte1,Byte2,Byte3,Byte4)
Mode and number of proofs:
    dword_bytes(+atom,+integer,-integer,-integer,-integer,-integer) - one

---

bytes_word/4

Converts two bytes into a 16-bit word using the given byte order.

Compilation flags:
    static

Template:
    bytes_word(Endian,Byte1,Byte2,Word)
Mode and number of proofs:
    bytes_word(+atom,+integer,+integer,-integer) - one

---

bytes_dword/6

Converts four bytes into a 32-bit word using the given byte order.

Compilation flags:
    static

Template:
    bytes_dword(Endian,Byte1,Byte2,Byte3,Byte4,Word)
Mode and number of proofs:
    bytes_dword(+atom,+integer,+integer,+integer,+integer,-integer) - one

---

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.30.2 character_set_protocol

Character set protocol for converting between lists of character codes and lists of bytes.

Availability:
    logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
    static

Dependencies:
    (none)

Remarks:

- Object names: Concrete object names are derived from the preferred IANA MIME names by lowercasing them and replacing hyphens with underscores.
- IANA registry metadata: The metadata predicates reflect the IANA character set registry. When the registry does not define a distinct preferred MIME alias, the preferred_mime_name/1 and name/1 predicates return the same atom.
- Unicode encodings: Unicode character sets use Unicode scalar values and do not emit or consume a byte order mark (BOM).

Inherited public predicates:
(none)

- Public predicates
  - preferred_mime_name/1
  - name/1
  - alias/1
  - mibenum/1
  - codes_to_bytes/2
  - bytes_to_codes/2
- Protected predicates
- Private predicates
- Operators

Public predicates

preferred_mime_name/1

Preferred MIME name for the character set according to the IANA registry.

Compilation flags:
static

Template:
preferred_mime_name(Name)
Mode and number of proofs:
preferred_mime_name(?atom) - zero_or_one

name/1

Registered character set name according to the IANA registry.

Compilation flags:
    static

Template:
    name(Name)
Mode and number of proofs:
    name(?atom) - zero_or_one

---

alias/1

Alias for the character set according to the IANA registry.

Compilation flags:
    static

Template:
    alias(Alias)
Mode and number of proofs:
    alias(?atom) - zero_or_more

---

mibenum/1

MIBenum value for the character set according to the IANA registry.

Compilation flags:
    static

Template:
    mibenum(MIBenum)
Mode and number of proofs:
    mibenum(?integer) - zero_or_one

---

codes_to_bytes/2

Converts a list of character codes to the corresponding list of bytes in the character set when all codes are representable.

Compilation flags:
    static

Template:
    codes_to_bytes(Codes,Bytes)
Mode and number of proofs:
    codes_to_bytes(+list(integer),--list(byte)) - zero_or_one

---

bytes_to_codes/2

Converts a list of bytes in the character set to the corresponding list of character codes when the byte sequence is valid for that character set.

Compilation flags:
    static

Template:
    bytes_to_codes(Bytes,Codes)
Mode and number of proofs:
    bytes_to_codes(+list(byte),--list(integer)) - zero_or_one

---

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

object

1.30.3 iso_8859_1

ISO-8859-1 character set encoder and decoder.

Availability:
logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
static, context_switching_calls

Imports:
public single_byte_character_set(255)

Remarks:
(none)

Inherited public predicates:
alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.30.4 iso_8859_10

ISO-8859-10 character set encoder and decoder.

Availability:
logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
static, context_switching_calls

Imports:
public mapped_single_byte_character_set

Remarks:
(none)

Inherited public predicates:
alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.30.5 iso_8859_13

ISO-8859-13 character set encoder and decoder.

Availability:
logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
static, context_switching_calls

Imports:
public mapped_single_byte_character_set

Remarks:

(none)

Inherited public predicates:
alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.30.6 iso_8859_14

ISO-8859-14 character set encoder and decoder.

Availability:
logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
static, context_switching_calls

Imports:
public mapped_single_byte_character_set

Remarks:
(none)

Inherited public predicates:
alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.30.7 iso_8859_15

ISO-8859-15 character set encoder and decoder.

Availability:
logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0

Date: 2026-04-05

Compilation flags:
static, context_switching_calls

Imports:
public mapped_single_byte_character_set

Remarks:
(none)

Inherited public predicates:
alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.30.8 iso_8859_16

ISO-8859-16 character set encoder and decoder.

Availability:
logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
static, context_switching_calls

Imports:
public mapped_single_byte_character_set

Remarks:
(none)

Inherited public predicates:
alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.30.9 iso_8859_2

ISO-8859-2 character set encoder and decoder.

Availability:
logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
static, context_switching_calls

Imports:
public mapped_single_byte_character_set

Remarks:
(none)

Inherited public predicates:
alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
- Private predicates

- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.30.10 iso_8859_3

ISO-8859-3 character set encoder and decoder.

Availability:
logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
static, context_switching_calls

Imports:
public mapped_single_byte_character_set

Remarks:
(none)

Inherited public predicates:

alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.30.11 iso_8859_4

ISO-8859-4 character set encoder and decoder.

Availability:
logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
static, context_switching_calls

Imports:
public mapped_single_byte_character_set

Remarks:
(none)

Inherited public predicates:
alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.30.12 iso_8859_9

ISO-8859-9 character set encoder and decoder.

Availability:
logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:

static, context_switching_calls

Imports:
public mapped_single_byte_character_set

Remarks:
(none)

Inherited public predicates:
alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.30.13 mapped_single_byte_character_set

Single-byte character set implementation parameterized by sparse byte-to-code mapping facts and undefined bytes.

Availability:
logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
static

Extends:
public character_set

Remarks:
(none)

Inherited public predicates:
alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
  - mapping/2
  - undefined/2
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

mapping/2

Returns, by backtracking, declared byte-to-code mappings.

Compilation flags:
    static

Template:
    mapping(Byte,Code)
Mode and number of proofs:
    mapping(?integer,?integer) - zero_or_more

---

undefined/2

Returns, by backtracking, declared undefined byte values.

Compilation flags:
    static

Template:
    undefined(Byte,Code)
Mode and number of proofs:
    undefined(?integer,?integer) - zero_or_more

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.30.14 single_byte_character_set(MaxCode)

Single-byte character set implementation parameterized by the maximum valid code.

Availability:
logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
static

Extends:
public character_set

Remarks:
(none)

Inherited public predicates:
alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.30.15 us_ascii

US-ASCII character set encoder and decoder.

Availability:
    logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
    static, context_switching_calls

Imports:
    public single_byte_character_set(127)

Remarks:
    (none)

Inherited public predicates:
    alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.30.16 utf_16_character_set(Endian)

UTF-16 character set implementation parameterized by byte order.

Availability:
    logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
    static

Extends:
    public character_set

Remarks:

(none)

Inherited public predicates:
alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.30.17 utf_16be

UTF-16 big-endian character set encoder and decoder.

Availability:
logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
static, context_switching_calls

Imports:
public utf_16_character_set(big_endian)

Remarks:
(none)

Inherited public predicates:
alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.30.18 utf_16le

UTF-16 little-endian character set encoder and decoder.

Availability:
logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0

Date: 2026-04-05

Compilation flags:
static, context_switching_calls

Imports:
public utf_16_character_set(little_endian)

Remarks:
(none)

Inherited public predicates:
alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.30.19 utf_32_character_set(Endian)

UTF-32 character set implementation parameterized by byte order.

Availability:
logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
static

Extends:
public character_set

Remarks:
(none)

Inherited public predicates:
alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.30.20 utf_32be

UTF-32 big-endian character set encoder and decoder.

Availability:
logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
static, context_switching_calls

Imports:
public utf_32_character_set(big_endian)

Remarks:
(none)

Inherited public predicates:
alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
- Private predicates

- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.30.21 utf_32le

UTF-32 little-endian character set encoder and decoder.

Availability:
logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
static, context_switching_calls

Imports:
public utf_32_character_set(little_endian)

Remarks:
(none)

Inherited public predicates:

alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.30.22 utf_8

UTF-8 character set encoder and decoder.

Availability:
logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
static, context_switching_calls

Imports:
public character_set

Extends:
public utf_8_character_set

Remarks:
(none)

Inherited public predicates:
alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.30.23 utf_8_character_set

UTF-8 character set implementation.

Availability:
logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0

Date: 2026-04-05

Compilation flags:
static, context_switching_calls

Imports:
public character_set

Remarks:
(none)

Inherited public predicates:
alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.30.24 windows_1250

Windows-1250 character set encoder and decoder.

Availability:
logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
static, context_switching_calls

Imports:
public mapped_single_byte_character_set

Remarks:
(none)

Inherited public predicates:
alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.30.25 windows_1251

Windows-1251 character set encoder and decoder.

Availability:
logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
static, context_switching_calls

Imports:
public mapped_single_byte_character_set

Remarks:
(none)

Inherited public predicates:
alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
- Private predicates

- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.30.26 windows_1252

Windows-1252 character set encoder and decoder.

Availability:
logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
static, context_switching_calls

Imports:
public mapped_single_byte_character_set

Remarks:
(none)

Inherited public predicates:

alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.30.27 windows_1253

Windows-1253 character set encoder and decoder.

Availability:
logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
static, context_switching_calls

Imports:
public mapped_single_byte_character_set

Remarks:
(none)

Inherited public predicates:
alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.30.28 windows_1254

Windows-1254 character set encoder and decoder.

Availability:
logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:

static, context_switching_calls

Imports:
public mapped_single_byte_character_set

Remarks:
(none)

Inherited public predicates:
alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.30.29 windows_1257

Windows-1257 character set encoder and decoder.

Availability:
logtalk_load(character_sets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
static, context_switching_calls

Imports:
public mapped_single_byte_character_set

Remarks:
(none)

Inherited public predicates:
alias/1 bytes_to_codes/2 codes_to_bytes/2 mibenum/1 name/1 preferred_mime_name/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.31 classification_protocols

category

1.31.1 classifier_common

Shared predicates for classifier diagnostics, dataset validation, mixed-feature distance calculations, and export.

Availability:
logtalk_load(classification_protocols(loader))

Author: Paulo Moura
Version: 2:0:0
Date: 2026-05-11

Compilation flags:
static

Implements:
public classifier_protocol
Extends:
public options
Uses:
format
list
population
type

Remarks:
(none)

Inherited public predicates:

check_classifier/1 check_option/1 check_options/1 classifier_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 print_classifier/1 valid_classifier/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
  - classifier_diagnostics_data/2
  - classifier_export_template/4
  - classifier_term_template/2
  - print_classifier_template/1
  - valid_attribute_names/1
  - valid_class_values/1
  - valid_feature_types/2
  - valid_discrete_values/1
  - valid_linear_encoders/1
  - dataset_attributes/2
  - dataset_examples/2
  - check_examples_non_empty/2
  - check_examples/2
  - check_complete_examples/2
  - check_complete_examples_nonvar/2
  - build_linear_encoders/4
  - examples_to_linear_rows/3
  - encode_linear_instance/3
  - linear_encoders_feature_count/2
  - valid_classifier_metadata/2
  - valid_classifier_metadata/3
  - mixed_feature_distance/5
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

classifier_diagnostics_data/2

Hook predicate that importing classifier implementations must define in order to expose diagnostics metadata.

Compilation flags:
    static

Template:
    classifier_diagnostics_data(Classifier,Diagnostics)
Mode and number of proofs:
    classifier_diagnostics_data(+compound,-list(compound)) - one

---

classifier_export_template/4

Hook predicate that importing classifier implementations must define in order to expose the exported classifier template for a given functor.

Compilation flags:
    static

Template:
    classifier_export_template(Dataset,Classifier,Functor,Template)
Mode and number of proofs:
    classifier_export_template(+object_identifier,+compound,+atom,-callable) - one

---

classifier_term_template/2

Hook predicate that importing classifier implementations must define in order to expose the learned classifier term template used by pretty-printing helpers.

Compilation flags:
    static

Template:
    classifier_term_template(Classifier,Template)
Mode and number of proofs:
    classifier_term_template(+compound,-callable) - one

---

print_classifier_template/1

Pretty-printing helper predicate used by importing classifier implementations to show the learned classifier term template.

Compilation flags:
    static

Template:
    print_classifier_template(Classifier)
Mode and number of proofs:
    print_classifier_template(+compound) - one

---

valid_attribute_names/1

True when a list of attribute names is a proper list of distinct atoms.

Compilation flags:
    static

Template:
    valid_attribute_names(AttributeNames)
Mode and number of proofs:
    valid_attribute_names(+list(atom)) - zero_or_one

---

valid_class_values/1

True when a list of class values is a non-empty proper list of distinct atoms.

Compilation flags:
    static

Template:
    valid_class_values(ClassValues)
Mode and number of proofs:
    valid_class_values(+list(atom)) - zero_or_one

---

valid_feature_types/2

True when a list of feature type tags is non-empty and each tag belongs to the given allowed set.

Compilation flags:
    static

Template:
    valid_feature_types(FeatureTypes,AllowedTypes)
Mode and number of proofs:
    valid_feature_types(+list,+list) - zero_or_one

---

valid_discrete_values/1

True when a list of categorical values is non-empty, contains only nonvar terms, and has no duplicates.

Compilation flags:
    static

Template:
    valid_discrete_values(Values)
Mode and number of proofs:

valid_discrete_values(+list) - zero_or_one

---

valid_linear_encoders/1

True when a list of encoders only contains valid continuous/3 or categorical/2 encoder terms with distinct attributes.

Compilation flags:
static

Template:
valid_linear_encoders(Encoders)
Mode and number of proofs:
valid_linear_encoders(+list(compound)) - zero_or_one

---

dataset_attributes/2

Collects the declared dataset attributes and their value domains.

Compilation flags:
static

Template:
dataset_attributes(Dataset,Attributes)
Mode and number of proofs:
dataset_attributes(+object_identifier,-list(pair)) - one

---

dataset_examples/2

Collects the dataset training examples as Id-Class-AttributeValues terms.

Compilation flags:
static

Template:
dataset_examples(Dataset,Examples)
Mode and number of proofs:
dataset_examples(+object_identifier,-list(compound)) - one

---

check_examples_non_empty/2

Checks that a training example collection is not empty.

Compilation flags:
static

Template:
check_examples_non_empty(Dataset,Examples)
Mode and number of proofs:
check_examples_non_empty(+object_identifier,+list) - one

---

check_examples/2

Checks that a training dataset is non-empty, that all example classes belong to the declared class values, and that provided attribute bindings use declared attributes with values matching the declared domains. Missing attribute bindings are allowed.

Compilation flags:
static

Template:
check_examples(Dataset,Examples)
Mode and number of proofs:
check_examples(+object_identifier,+list) - one_or_error

---

check_complete_examples/2

Checks that a training dataset is non-empty, that all example classes belong to the declared class values, and that each example contains every declared attribute exactly once with values matching the declared domains. Missing values represented using variables are allowed.

Compilation flags:
    static

Template:
    check_complete_examples(Dataset,Examples)
Mode and number of proofs:
    check_complete_examples(+object_identifier,+list) - one_or_error

---

check_complete_examples_nonvar/2

Checks that a training dataset is non-empty, that all example classes belong to the declared class values, and that each example contains every declared attribute exactly once with non-variable values matching the declared domains.

Compilation flags:
    static

Template:
    check_complete_examples_nonvar(Dataset,Examples)
Mode and number of proofs:
    check_complete_examples_nonvar(+object_identifier,+list) - one_or_error

---

build_linear_encoders/4

Builds linear-model encoders for continuous and categorical attributes. Continuous encoders optionally standardize features when FeatureScaling is true.

Compilation flags:
    static

Template:

build_linear_encoders(Attributes,Examples,FeatureScaling,Encoders)
Mode and number of proofs:
build_linear_encoders(+list(pair),+list(compound),+boolean,-list(compound)) - one

---

examples_to_linear_rows/3

Encodes examples into Features-Class rows using the given linear encoders.

Compilation flags:
static

Template:
examples_to_linear_rows(Examples,Encoders,Rows)
Mode and number of proofs:
examples_to_linear_rows(+list(compound),+list(compound),-list(pair)) - one

---

encode_linear_instance/3

Encodes an instance into a numeric feature vector using the given linear encoders.

Compilation flags:
static

Template:
encode_linear_instance(Encoders,Instance,Features)
Mode and number of proofs:
encode_linear_instance(+list(compound),+list(pair),-list(float)) - one

---

linear_encoders_feature_count/2

Computes the number of numeric features produced by a set of linear encoders.

Compilation flags:
    static

Template:
    linear_encoders_feature_count(Encoders,Count)
Mode and number of proofs:
    linear_encoders_feature_count(+list(compound),-integer) - one

---

valid_classifier_metadata/2

True when diagnostics metadata contains the expected model term.

Compilation flags:
    static

Template:
    valid_classifier_metadata(Model,Diagnostics)
Mode and number of proofs:
    valid_classifier_metadata(+atom,+list(compound)) - zero_or_one

---

valid_classifier_metadata/3

True when diagnostics metadata contains the expected model term and records the given effective options.

Compilation flags:
    static

Template:
    valid_classifier_metadata(Model,Options,Diagnostics)
Mode and number of proofs:
    valid_classifier_metadata(+atom,+list(compound),+list(compound)) - zero_or_one

---

mixed_feature_distance/5

Computes a distance between two mixed-feature vectors using the given feature types and one of the supported metrics euclidean, manhattan, chebyshev, cosine, or minkowski(Order).

Compilation flags:
    static

Template:
    mixed_feature_distance(Metric,FeatureTypes,Values1,Values2,Distance)
Mode and number of proofs:
    mixed_feature_distance(+term,+list,+list,+list,-float) - one_or_error

---

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.31.2 classifier_protocol

Protocol for machine learning classifiers.

Availability:
    logtalk_load(classification_protocols(loader))

Author: Paulo Moura
Version: 2:0:0
Date: 2026-05-07

Compilation flags:
    static

Dependencies:
    (none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - learn/3
  - learn/2
  - predict/3
  - check_classifier/1
  - valid_classifier/1
  - diagnostics/2
  - diagnostic/2
  - classifier_options/2
  - export_to_clauses/4
  - export_to_file/4
  - print_classifier/1
- Protected predicates
- Private predicates
- Operators

Public predicates

learn/3

Learns a classifier from the given dataset object using the specified options.

Compilation flags:
static

Template:
learn(Dataset,Classifier,Options)
Mode and number of proofs:
learn(+object_identifier,-compound,+list(compound)) - one

learn/2

Learns a classifier from the given dataset object using default options.

Compilation flags:
    static

Template:
    learn(Dataset,Classifier)
Mode and number of proofs:
    learn(+object_identifier,-compound) - one

---

predict/3

Predicts the class label for a new instance using the learned classifier. The instance is a list of Attribute-Value pairs.

Compilation flags:
    static

Template:
    predict(Classifier,Instance,Class)
Mode and number of proofs:
    predict(+compound,+list,-atom) - one

---

check_classifier/1

Checks that a learned classifier term is structurally valid for the receiving implementation. Throws an exception when the term is not a valid classifier representation.

Compilation flags:
    static

Template:
    check_classifier(Classifier)
Mode and number of proofs:
    check_classifier(@compound) - one_or_error

Exceptions:
Classifier is a variable:
instantiation_error
Classifier is neither a variable nor a valid classifier:
domain_error(classifier,Classifier)

---

valid_classifier/1

True when a learned classifier term is structurally valid for the receiving implementation. Succeeds iff check_classifier/1 succeeds without throwing an exception.

Compilation flags:
static

Template:
valid_classifier(Classifier)
Mode and number of proofs:
valid_classifier(@compound) - zero_or_one

---

diagnostics/2

Returns diagnostics and metadata associated with a learned classifier in a representation-independent way.

Compilation flags:
static

Template:
diagnostics(Classifier,Diagnostics)
Mode and number of proofs:
diagnostics(+compound,-list(compound)) - one

---

diagnostic/2

Tests or enumerates individual diagnostics metadata terms for a learned classifier.

Compilation flags:
    static

Template:
    diagnostic(Classifier,Diagnostic)
Mode and number of proofs:
    diagnostic(+compound,?compound) - zero_or_more

---

classifier_options/2

Returns the effective training options recorded in a learned classifier diagnostics list.

Compilation flags:
    static

Template:
    classifier_options(Classifier,Options)
Mode and number of proofs:
    classifier_options(+compound,-list(compound)) - zero_or_one

---

export_to_clauses/4

Converts a classifier into a list of predicate clauses. Functor is the functor for the generated predicate clauses. When exporting a serialized classifier term, a noun such as classifier or model is usually clearer than a verb such as classify.

Compilation flags:
    static

Template:
    export_to_clauses(Dataset,Classifier,Functor,Clauses)
Mode and number of proofs:
    export_to_clauses(+object_identifier,+compound,+callable,-list(clause)) - one

export_to_file/4

Exports a classifier to a file. Functor is the functor for the generated predicate clauses. When exporting a serialized classifier term, a noun such as classifier or model is usually clearer than a verb such as classify.

Compilation flags:
    static

Template:
    export_to_file(Dataset,Classifier,Functor,File)
Mode and number of proofs:
    export_to_file(+object_identifier,+compound,+callable,+atom) - one

---

print_classifier/1

Prints a classifier to the current output stream in a human-readable format.

Compilation flags:
    static

Template:
    print_classifier(Classifier)
Mode and number of proofs:
    print_classifier(+compound) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

adaptive_boosting_classifier, c45_classifier, gradient_boosting_classifier, isolation_forest_anomaly_detector, kernel_svm_classifier, knn_classifier, lda_classifier, linear_svm_classifier, logistic_regression_classifier, naive_bayes_classifier, nearest_centroid_classifier, qda_classifier, random_forest_classifier, sgd_classifier

protocol

1.31.3 dataset_protocol

Protocol for datasets used with classifier algorithms.

Availability:
logtalk_load(classification_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-19

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - attribute_values/2
  - class/1
  - class_values/1
  - example/3
- Protected predicates
- Private predicates
- Operators

Public predicates

attribute_values/2

Enumerates by backtracking the attributes and their possible values. For discrete attributes, Values is a list of possible values. For continuous (numeric) attributes, Values is the atom continuous.

Compilation flags:
    static

Template:
    attribute_values(Attribute,Values)
Mode and number of proofs:
    attribute_values(?atom,-list(atom)) - zero_or_more
    attribute_values(?atom,-atom) - zero_or_more

---

class/1

Returns the name of the target class attribute.

Compilation flags:
    static

Template:
    class(Class)
Mode and number of proofs:
    class(-atom) - one

---

class_values/1

Returns the list of possible values for the target class attribute.

Compilation flags:
    static

Template:
    class_values(Values)
Mode and number of proofs:
    class_values(-list(atom)) - one

---

example/3

Enumerates by backtracking the examples in the dataset. Each example has an Id, a Class value, and a list of Attribute-Value pairs.

Compilation flags:
    static

Template:
    example(Id,Class,AttributeValues)
Mode and number of proofs:
    example(-integer,-atom,-list(pair)) - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

category

1.31.4 probabilistic_classifier_common

Shared predicates for probabilistic classifiers.

Availability:
logtalk_load(classification_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-11

Compilation flags:
static

Implements:
public probabilistic_classifier_protocol
Extends:
public classifier_common

Remarks:
(none)

Inherited public predicates:
check_classifier/1 check_option/1 check_options/1 classifier_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 predict_probabilities/3 print_classifier/1 valid_classifier/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
  - predict_from_probabilities/3

    - max_probability/3
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

predict_from_probabilities/3

Predicts the class label for a new instance by selecting the maximum-probability class returned by predict_probabilities/3.

Compilation flags:
    static

Template:
    predict_from_probabilities(Classifier,Instance,Class)
Mode and number of proofs:
    predict_from_probabilities(+compound,+list,-atom) - one

---

max_probability/3

Selects the class-probability pair with maximum probability from a non-empty list of Class-Probability pairs.

Compilation flags:
    static

Template:
    max_probability(Probabilities,Class,Probability)
Mode and number of proofs:
    max_probability(+list(pair),-atom,-float) - one

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.31.5 probabilistic_classifier_protocol

Protocol for classifiers that can estimate class probabilities.

Availability:
logtalk_load(classification_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-11

Compilation flags:
static

Extends:
public classifier_protocol

Remarks:
(none)

Inherited public predicates:
check_classifier/1 classifier_options/2 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 predict/3 print_classifier/1 valid_classifier/1

- Public predicates
  - predict_probabilities/3
- Protected predicates
- Private predicates
- Operators

Public predicates

predict_probabilities/3

Predicts class probabilities for a new instance using the learned classifier. The instance is a list of Attribute-Value pairs.

Compilation flags:
    static

Template:
    predict_probabilities(Classifier,Instance,Probabilities)
Mode and number of proofs:
    predict_probabilities(+compound,+list,-list) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

classifier_protocol, adaptive_boosting_classifier, gradient_boosting_classifier, kernel_svm_classifier, knn_classifier, logistic_regression_classifier, naive_bayes_classifier, nearest_centroid_classifier, random_forest_classifier, sgd_classifier

1.32 clo_span_pattern_miner

object

1.32.1 clo_span_pattern_miner

Closed sequential pattern miner for sequence datasets using closure-aware projected-database search.

Availability:
logtalk_load(clo_span_pattern_miner(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public sequential_pattern_mining_common
Uses:
format
list

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_pattern_miner/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 mine/2 mine/3 option/2 option/3 pattern_miner_options/2 print_pattern_miner/1 valid_option/1 valid_options/1 valid_pattern_miner/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

pattern_miner_protocol, sequence_dataset_protocol, prefix_span_pattern_miner, gsp_pattern_miner, spade_pattern_miner

1.33 clustering_protocols

category

1.33.1 clusterer_common

Shared predicates for clusterer learning defaults, export, and common dataset and encoding helpers.

Availability:
logtalk_load(clustering_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-30

Compilation flags:
static

Implements:
public clusterer_protocol
Extends:
public options

Uses:

format
list
population
type

Remarks:

(none)

Inherited public predicates:

check_clusterer/1 check_option/1 check_options/1 cluster/3 clusterer_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 print_clusterer/1 valid_clusterer/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
  - dataset_attributes/2
  - valid_attribute_names/1
  - valid_continuous_encoders/1
  - valid_discrete_encoders/1
  - valid_mixed_encoders/1
  - valid_mixed_vectors/2
  - valid_clusterer_metadata/3
  - valid_diagnostic_count/3
  - valid_diagnostic_choice/3
  - check_continuous_attributes/1
  - check_examples_non_empty/2
  - check_examples/3
  - check_example_values/2
  - check_example_attributes/2
  - check_attribute_bindings/2
  - attribute_value/3
  - build_encoders/4
  - known_attribute_values/3
  - examples_to_rows/3
  - encode_instance/3

  - check_encoded_attribute_bindings/2
  - normalize_continuous/4
  - check_cluster_count/2
  - take_first_k/3
  - remove_candidate/3
  - clusterer_diagnostics_data/2
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

dataset_attributes/2

Collects the dataset attribute declarations as Attribute-Values pairs.

Compilation flags:
    static

Template:
    dataset_attributes(Dataset,Attributes)
Mode and number of proofs:
    dataset_attributes(+object_identifier,-list(pair)) - one

---

valid_attribute_names/1

True when a list of attribute names is a proper list of distinct atoms.

Compilation flags:
    static

Template:
    valid_attribute_names(AttributeNames)
Mode and number of proofs:

valid_attribute_names(+list(atom)) - zero_or_one

---

valid_continuous_encoders/1

True when a list of encoders only contains valid continuous encoder terms with distinct attributes.

Compilation flags:
static

Template:
valid_continuous_encoders(Encoders)
Mode and number of proofs:
valid_continuous_encoders(+list(compound)) - zero_or_one

---

valid_discrete_encoders/1

True when a list of encoders only contains valid discrete encoder terms with distinct attributes.

Compilation flags:
static

Template:
valid_discrete_encoders(Encoders)
Mode and number of proofs:
valid_discrete_encoders(+list(compound)) - zero_or_one

---

valid_mixed_encoders/1

True when a list of encoders only contains valid continuous or discrete encoder terms with distinct attributes.

Compilation flags:
static

Template:

valid_mixed_encoders(Encoders)
Mode and number of proofs:
valid_mixed_encoders(+list(compound)) - zero_or_one

---

valid_mixed_vectors/2

True when all vectors conform to the given continuous or discrete encoder specifications.

Compilation flags:
static

Template:
valid_mixed_vectors(Encoders,Vectors)
Mode and number of proofs:
valid_mixed_vectors(+list(compound),+list) - zero_or_one

---

valid_clusterer_metadata/3

True when diagnostics metadata contains the expected model term and records the given effective options.

Compilation flags:
static

Template:
valid_clusterer_metadata(Model,Options,Diagnostics)
Mode and number of proofs:
valid_clusterer_metadata(+atom,+list(compound),+list(compound)) - zero_or_one

---

valid_diagnostic_count/3

True when diagnostics contains a count term with the given functor and integer value.

Compilation flags:
    static

Template:
    valid_diagnostic_count(Functor,Diagnostics,Count)
Mode and number of proofs:
    valid_diagnostic_count(+atom,+list(compound),+integer) - zero_or_one

---

valid_diagnostic_choice/3

True when diagnostics contains a term with the given functor and a value selected from the allowed choices.

Compilation flags:
    static

Template:
    valid_diagnostic_choice(Functor,Diagnostics,Choices)
Mode and number of proofs:
    valid_diagnostic_choice(+atom,+list(compound),+list) - zero_or_one

---

check_continuous_attributes/1

Checks that all declared dataset attributes are continuous.

Compilation flags:
    static

Template:
    check_continuous_attributes(Attributes)
Mode and number of proofs:
    check_continuous_attributes(+list(pair)) - one

---

check_examples_non_empty/2

Checks that a training example collection is not empty.

Compilation flags:
    static

Template:
    check_examples_non_empty(Dataset,Examples)
Mode and number of proofs:
    check_examples_non_empty(+object_identifier,+list) - one

---

check_examples/3

Checks that a continuous training dataset is non-empty and that all example values are numeric.

Compilation flags:
    static

Template:
    check_examples(Dataset,AttributeNames,Examples)
Mode and number of proofs:
    check_examples(+object_identifier,+list(atom),+list) - one

---

check_example_values/2

Checks that all example attribute values are present and numeric for the declared attributes.

Compilation flags:
    static

Template:
    check_example_values(Examples,AttributeNames)
Mode and number of proofs:
    check_example_values(+list,+list(atom)) - one

---

check_example_attributes/2

Checks that a single example contains exactly the declared attributes and that all values are numeric.

Compilation flags:
    static

Template:
    check_example_attributes(AttributeNames,AttributeValues)
Mode and number of proofs:
    check_example_attributes(+list(atom),+list(pair)) - one

---

check_attribute_bindings/2

Checks that an attribute-value list contains each declared attribute exactly once and no undeclared attributes.

Compilation flags:
    static

Template:
    check_attribute_bindings(AttributeNames,AttributeValues)
Mode and number of proofs:
    check_attribute_bindings(+list(atom),+list(pair)) - one

---

attribute_value/3

Looks up an attribute value in a list of Attribute-Value pairs.

Compilation flags:
    static

Template:
    attribute_value(Attribute,AttributeValues,Value)
Mode and number of proofs:
    attribute_value(+atom,+list(pair),-term) - one

---

build_encoders/4

Builds continuous feature encoders by computing per-attribute centering and optional scaling statistics.

Compilation flags:
    static

Template:
    build_encoders(AttributeNames,Examples,Options,Encoders)
Mode and number of proofs:
    build_encoders(+list(atom),+list,+list(compound),-list(compound)) - one

---

known_attribute_values/3

Collects the known numeric values for a given attribute across the training examples.

Compilation flags:
    static

Template:
    known_attribute_values(Examples,Attribute,Values)
Mode and number of proofs:
    known_attribute_values(+list,+atom,-list(number)) - one

---

examples_to_rows/3

Encodes training examples into Id-Features rows using the object-local or imported encoder implementation.

Compilation flags:
    static

Template:
    examples_to_rows(Examples,Encoders,Rows)
Mode and number of proofs:
    examples_to_rows(+list,+list(compound),-list(pair)) - one

---

encode_instance/3

Encodes an instance using the learned continuous attribute encoders after checking that it contains exactly the declared attributes.

Compilation flags:
    static

Template:
    encode_instance(Encoders,AttributeValues,Features)
Mode and number of proofs:
    encode_instance(+list(compound),+list(pair),-list(number)) - one

---

check_encoded_attribute_bindings/2

Checks that an attribute-value list contains each attribute described by the encoders exactly once and no undeclared attributes.

Compilation flags:
    static

Template:
    check_encoded_attribute_bindings(Encoders,AttributeValues)
Mode and number of proofs:
    check_encoded_attribute_bindings(+list(compound),+list(pair)) - one

---

normalize_continuous/4

Normalizes a continuous value using the learned centering and scaling parameters.

Compilation flags:
    static

Template:
    normalize_continuous(Value,Mean,Scale,Feature)
Mode and number of proofs:
    normalize_continuous(+number,+number,+number,-number) - one

---

check_cluster_count/2

Checks that the requested cluster count does not exceed the number of examples.

Compilation flags:
    static

Template:
    check_cluster_count(K,Count)
Mode and number of proofs:
    check_cluster_count(+integer,+integer) - one

---

take_first_k/3

Collects the first K vectors from Id-Vector rows.

Compilation flags:
    static

Template:
    take_first_k(K,Rows,Vectors)
Mode and number of proofs:
    take_first_k(+integer,+list,-list) - one

---

remove_candidate/3

Removes the first matching Id-Vector candidate from a candidate list.

Compilation flags:
    static

Template:
    remove_candidate(Candidate,Candidates,RemainingCandidates)
Mode and number of proofs:

remove_candidate(+pair,+list(pair),-list(pair)) - one

---

clusterer_diagnostics_data/2

Hook predicate that importing clusterer implementations must define in order to expose diagnostics metadata.

Compilation flags:
static

Template:
clusterer_diagnostics_data(Clusterer,Diagnostics)
Mode and number of proofs:
clusterer_diagnostics_data(+compound,-list(compound)) - one

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.33.2 clusterer_protocol

Protocol for machine learning clusterers.

Availability:
logtalk_load(clustering_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-30

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - learn/3
  - learn/2
  - cluster/3
  - diagnostics/2
  - check_clusterer/1
  - valid_clusterer/1
  - diagnostic/2
  - clusterer_options/2
  - export_to_clauses/4
  - export_to_file/4
  - print_clusterer/1
- Protected predicates
- Private predicates
- Operators

Public predicates

learn/3

Learns a clusterer from the given dataset object using the specified options.

Compilation flags:
static

Template:
learn(Dataset,Clusterer,Options)
Mode and number of proofs:

learn(+object_identifier,-compound,+list(compound)) - one

---

learn/2

Learns a clusterer from the given dataset object using default options.

Compilation flags:
static

Template:
learn(Dataset,Clusterer)
Mode and number of proofs:
learn(+object_identifier,-compound) - one

---

cluster/3

Assigns a new instance to a cluster using the learned clusterer. The instance is a list of Attribute-Value pairs.

Compilation flags:
static

Template:
cluster(Clusterer,Instance,Cluster)
Mode and number of proofs:
cluster(+compound,+list,-ground) - one

---

diagnostics/2

Returns diagnostics and metadata associated with a learned clusterer in a representation-independent way.

Compilation flags:
static

Template:
diagnostics(Clusterer,Diagnostics)
Mode and number of proofs:
diagnostics(+compound,-list(compound)) - one

---

check_clusterer/1

Checks that a learned clusterer term is structurally valid for the receiving implementation. Throws an exception when the term is not a valid clusterer representation.

Compilation flags:
static

Template:
check_clusterer(Clusterer)
Mode and number of proofs:
check_clusterer(@compound) - one_or_error

Exceptions:
Clusterer is a variable:
instantiation_error
Clusterer is neither a variable nor a valid clusterer:
domain_error(clusterer,Clusterer)

---

valid_clusterer/1

True when a learned clusterer term is structurally valid for the receiving implementation. Succeeds iff check_clusterer/1 succeeds without throwing an exception.

Compilation flags:
static

Template:
valid_clusterer(Clusterer)
Mode and number of proofs:
valid_clusterer(@compound) - zero_or_one

---

diagnostic/2

Tests or enumerates individual diagnostics metadata terms for a learned clusterer.

Compilation flags:
    static

Template:
    diagnostic(Clusterer,Diagnostic)
Mode and number of proofs:
    diagnostic(+compound,?compound) - zero_or_more

---

clusterer_options/2

Returns the effective training options recorded in a learned clusterer diagnostics list.

Compilation flags:
    static

Template:
    clusterer_options(Clusterer,Options)
Mode and number of proofs:
    clusterer_options(+compound,-list(compound)) - one

---

export_to_clauses/4

Converts a clusterer into a list of predicate clauses. Functor is the functor for the generated predicate clauses.

Compilation flags:
    static

Template:
    export_to_clauses(Dataset,Clusterer,Functor,Clauses)
Mode and number of proofs:
    export_to_clauses(+object_identifier,+compound,+callable,-list(clause)) - one

---

export_to_file/4

Exports a clusterer to a file. Functor is the functor for the generated Functor(Clusterer) predicate clause.

Compilation flags:
    static

Template:
    export_to_file(Dataset,Clusterer,Functor,File)
Mode and number of proofs:
    export_to_file(+object_identifier,+compound,+callable,+atom) - one

---

print_clusterer/1

Prints a clusterer to the current output stream in a human-readable format.

Compilation flags:
    static

Template:
    print_clusterer(Clusterer)
Mode and number of proofs:
    print_clusterer(+compound) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

clustering_dataset_protocol

protocol

1.33.3 clustering_dataset_protocol

Protocol for datasets used with clustering algorithms.

Availability:
logtalk_load(clustering_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-23

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - attribute_values/2
  - example/2
- Protected predicates
- Private predicates
- Operators

Public predicates

attribute_values/2

Enumerates by backtracking the attributes and their possible values. For discrete attributes, Values is a list of possible values. For continuous (numeric) attributes, Values is the atom continuous.

Compilation flags:
    static

Template:
    attribute_values(Attribute,Values)
Mode and number of proofs:
    attribute_values(?atom,-list(atom)) - zero_or_more
    attribute_values(?atom,-atom) - zero_or_more

---

example/2

Enumerates by backtracking the examples in the dataset. Each example has an Id and a list of Attribute-Value pairs.

Compilation flags:
    static

Template:
    example(Id,AttributeValues)
Mode and number of proofs:
    example(-integer,-list(pair)) - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

category

1.33.4 search_indexing

Shared helpers for adaptive search-index construction and range queries used by clustering libraries.

Availability:
logtalk_load(clustering_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-23

Compilation flags:
static

Uses:
avltree
list

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
  - build_auto_search_index/3
  - build_metric_tree/3
  - build_grid_index/3
  - range_query/5

    - search_index_cell_size/2
    - select_metric_pivot/4
    - distance/4
    - split_sorted_rows/5
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

build_auto_search_index/3

Builds an adaptive search index by selecting either a grid index or a metric tree based on dataset shape.

Compilation flags:
    static

Template:
    build_auto_search_index(Rows,Options,SearchIndex)
Mode and number of proofs:
    build_auto_search_index(+list(pair),+list(compound),-compound) - one

---

build_metric_tree/3

Builds a metric tree search index for encoded dataset rows.

Compilation flags:
    static

Template:
    build_metric_tree(Rows,Options,MetricTree)
Mode and number of proofs:
    build_metric_tree(+list(pair),+list(compound),-compound) - one

---

build_grid_index/3

Builds a grid-based search index for encoded dataset rows using the object-defined cell size.

Compilation flags:
    static

Template:
    build_grid_index(Rows,Options,GridIndex)
Mode and number of proofs:
    build_grid_index(+list(pair),+list(compound),-compound) - one

---

range_query/5

Queries a search index for rows within the given epsilon distance of an encoded feature vector.

Compilation flags:
    static

Template:
    range_query(SearchIndex,Vector,Options,Epsilon,Neighbors)
Mode and number of proofs:
    range_query(+compound,+list(number),+list(compound),+number,-list(pair)) - one

---

search_index_cell_size/2

Returns the grid cell size to use when constructing a grid-based search index.

Compilation flags:
    static

Template:
    search_index_cell_size(Options,CellSize)
Mode and number of proofs:

search_index_cell_size(+list(compound),-number) - one

---

select_metric_pivot/4

Selects a pivot row and returns the remaining rows decorated and sorted by distance to that pivot.

Compilation flags:
static

Template:
select_metric_pivot(Rows,Options,Pivot,SortedRows)
Mode and number of proofs:
select_metric_pivot(+list(pair),+list(compound),-pair,-list(pair)) - one

---

distance/4

Computes the distance between two encoded feature vectors using the effective object options.

Compilation flags:
static

Template:
distance(Options,Vector1,Vector2,Distance)
Mode and number of proofs:
distance(+list(compound),+list(number),+list(number),-number) - one

---

split_sorted_rows/5

Splits rows decorated with distances into inner and outer partitions and returns their boundary distances.

Compilation flags:
static

Template:

split_sorted_rows(SortedRows,InnerUpperBound,OuterLowerBound,InnerRows,OuterRows)
Mode and number of proofs:
split_sorted_rows(+list(pair),-number,-number,-list(pair),-list(pair)) - one

---

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.34 code_metrics

object

1.34.1 cc_metric

Cyclomatic complexity metric. All defined predicates that are not called or updated are counted as graph connected components (the reasoning being that these predicates can be considered entry points). The score is represented by a non-negative integer.

Availability:
logtalk_load(code_metrics(loader))

Author: Paulo Moura
Version: 0:5:3
Date: 2026-03-13

Compilation flags:
static, context_switching_calls

Imports:
public code_metrics_utilities
public code_metric
Provides:
logtalk::message_tokens//2
Uses:
list
logtalk

numberlist

Remarks:
(none)

Inherited public predicates:
all/0 all/1 all_score/1 check_option/1 check_options/1 default_option/1 default_options/1 directory/1 directory/2 directory_score/2 entity/1 entity_score/2 file/1 file/2 file_score/2 format_entity_score//2 library/1 library/2 library_score/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory_score/2 rlibrary/1 rlibrary/2 rlibrary_score/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.34.2 code_metric

Core predicates for computing source code metrics.

Availability:
logtalk_load(code_metrics(loader))

Author: Ebrahim Azarisooreh and Paulo Moura
Version: 0:13:0
Date: 2025-10-06

Compilation flags:
static

Extends:
public code_metrics_utilities
public options
Uses:
list
logtalk
os
type

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 default_option/1 default_options/1 option/2 option/3 valid_option/1 valid_options/1

- Public predicates
  - entity/1
  - file/2
  - file/1
  - directory/2
  - directory/1
  - rdirectory/2
  - rdirectory/1
  - library/2

    - library/1
    - rlibrary/2
    - rlibrary/1
    - all/1
    - all/0
    - entity_score/2
    - library_score/2
    - rlibrary_score/2
    - file_score/2
    - directory_score/2
    - rdirectory_score/2
    - all_score/1
    - format_entity_score//2
- Protected predicates
    - process_entity/2
    - process_file/2
    - process_directory/2
    - process_rdirectory/2
    - process_library/2
    - process_rlibrary/2
    - process_all/1
    - sub_directory/2
    - sub_library/2
- Private predicates
- Operators

Public predicates

entity/1

Scans an entity and prints its metric score.

Compilation flags:
    static

Template:

entity(Entity)
Mode and number of proofs:
entity(+term) - zero_or_one

---

file/2

Prints metric scores for all the entities defined in a loaded source file using the given options.

Compilation flags:
static

Template:
file(File,Options)
Mode and number of proofs:
file(+atom,+list(compound)) - zero_or_one

---

file/1

Prints metric scores for all the entities defined in a loaded source file using default options.

Compilation flags:
static

Template:
file(File)
Mode and number of proofs:
file(+atom) - zero_or_one

---

directory/2

Scans a directory and prints metric scores for all entities defined in its loaded source files using the given options.

Compilation flags:
    static

Template:
    directory(Directory,Options)
Mode and number of proofs:
    directory(+atom,+list(compound)) - one

---

directory/1

Scans a directory and prints metric scores for all entities defined in its loaded source files using default options.

Compilation flags:
    static

Template:
    directory(Directory)
Mode and number of proofs:
    directory(+atom) - one

---

rdirectory/2

Recursive version of the directory/1 predicate using the given options.

Compilation flags:
    static

Template:
    rdirectory(Directory,Options)
Mode and number of proofs:
    rdirectory(+atom,+list(compound)) - one

---

rdirectory/1

Recursive version of the directory/1 predicate using default options.

Compilation flags:
    static

Template:
    rdirectory(Directory)
Mode and number of proofs:
    rdirectory(+atom) - one

---

library/2

Prints metrics scores for all loaded entities from a given library using the given options.

Compilation flags:
    static

Template:
    library(Library,Options)
Mode and number of proofs:
    library(+atom,+list(compound)) - one

---

library/1

Prints metrics scores for all loaded entities from a given library using default options.

Compilation flags:
    static

Template:
    library(Library)
Mode and number of proofs:

library(+atom) - one

---

rlibrary/2

Recursive version of the library/1 predicate using the given options.

Compilation flags:
static

Template:
rlibrary(Library,Options)
Mode and number of proofs:
rlibrary(+atom,+list(compound)) - one

---

rlibrary/1

Recursive version of the library/1 predicate using default options.

Compilation flags:
static

Template:
rlibrary(Library)
Mode and number of proofs:
rlibrary(+atom) - one

---

all/1

Scans all loaded entities and prints their metric scores using the given options.

Compilation flags:
static

Template:

all(Options)
Mode and number of proofs:
all(+list(compound)) - one

---

all/0

Scans all loaded entities and prints their metric scores using default options.

Compilation flags:
static

Mode and number of proofs:
all - one

---

entity_score/2

Score is a term that represents the metric score associated with a loaded entity. Fails if the metric does not apply.

Compilation flags:
static

Template:
entity_score(Entity,Score)
Mode and number of proofs:
entity_score(@entity_identifier,-ground) - zero_or_one

---

library_score/2

Score is a term that represents the metric score associated with a loaded library source files. Fails if the metric does not apply.

Compilation flags:
static

Template:
library_score(Library,Score)
Mode and number of proofs:
library_score(@atom,-ground) - zero_or_one

---

rlibrary_score/2

Score is a term that represents the metric score associated with loaded source files from a library and its sub-libraries. Fails if the metric does not apply.

Compilation flags:
static

Template:
rlibrary_score(Library,Score)
Mode and number of proofs:
rlibrary_score(@atom,-ground) - zero_or_one

---

file_score/2

Score is a term that represents the metric score associated with a loaded source file. Fails if the metric does not apply.

Compilation flags:
static

Template:
file_score(File,Score)
Mode and number of proofs:
file_score(@atom,-ground) - zero_or_one

---

directory_score/2

Score is a term that represents the metric score associated with loaded source files from a directory. Fails if the metric does not apply.

Compilation flags:
    static

Template:
    directory_score(Directory,Score)
Mode and number of proofs:
    directory_score(@atom,-ground) - zero_or_one

---

rdirectory_score/2

Score is a term that represents the metric score associated with loaded source files from a directory and its sub-directories. Fails if the metric does not apply.

Compilation flags:
    static

Template:
    rdirectory_score(Directory,Score)
Mode and number of proofs:
    rdirectory_score(@atom,-ground) - zero_or_one

---

all_score/1

Score is a term that represents the metric score associated with all loaded source files. Fails if the metric does not apply.

Compilation flags:
    static

Template:
    all_score(Score)
Mode and number of proofs:

all_score(-ground) - zero_or_one

---

format_entity_score//2

Formats the entity score for pretty printing.

Compilation flags:
static

Template:
format_entity_score(Entity,Score)
Mode and number of proofs:
format_entity_score(@entity_identifier,+ground) - one

Protected predicates

process_entity/2

Processes an entity of the given kind.

Compilation flags:
static

Template:
process_entity(Kind,Entity)
Mode and number of proofs:
process_entity(+atom,@entity_identifier) - one

---

process_file/2

Processes a source file using the given options.

Compilation flags:
    static

Template:
    process_file(Path,Options)
Mode and number of proofs:
    process_file(+atom,+list(compound)) - one

---

process_directory/2

Processes a directory of source files using the given options.

Compilation flags:
    static

Template:
    process_directory(Path,Options)
Mode and number of proofs:
    process_directory(+atom,+list(compound)) - one

---

process_rdirectory/2

Recursively process a directory of source files using the given options.

Compilation flags:
    static

Template:
    process_rdirectory(Path,Options)
Mode and number of proofs:
    process_rdirectory(+atom,+list(compound)) - one

---

process_library/2

Processes a library of source files using the given options.

Compilation flags:
    static

Template:
    process_library(Library,Options)
Mode and number of proofs:
    process_library(+atom,+list(compound)) - one

---

process_rlibrary/2

Recursively process a library of source files using the given options.

Compilation flags:
    static

Template:
    process_rlibrary(Library,Options)
Mode and number of proofs:
    process_rlibrary(+atom,+list(compound)) - one

---

process_all/1

Processes all loaded source code using the given options.

Compilation flags:
    static

Template:
    process_all(Options)
Mode and number of proofs:
    process_all(+list(compound)) - one

---

sub_directory/2

Enumerates, by backtracking, all directory sub-directories containing loaded files.

Compilation flags:
    static

Template:
    sub_directory(Directory,SubDirectory)
Mode and number of proofs:
    sub_directory(+atom,-atom) - one

---

sub_library/2

Enumerates, by backtracking, all library sub-libraries.

Compilation flags:
    static

Template:
    sub_library(Library,SubLibrary)
Mode and number of proofs:
    sub_library(+atom,-atom) - one

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.34.3 code_metrics

Helper object to apply all loaded code metrics.

Availability:
logtalk_load(code_metrics(loader))

Author: Ebrahim Azarisooreh and Paulo Moura
Version: 0:1:0
Date: 2017-12-31

Compilation flags:
static, context_switching_calls

Imports:
public code_metric
Uses:
logtalk

Remarks:
(none)

Inherited public predicates:
all/0 all/1 all_score/1 check_option/1 check_options/1 default_option/1 default_options/1 directory/1 directory/2 directory_score/2 entity/1 entity_score/2 file/1 file/2 file_score/2 format_entity_score//2 library/1 library/2 library_score/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory_score/2 rlibrary/1 rlibrary/2 rlibrary_score/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.34.4 code_metrics_messages

Message translations for the code_metrics tool.

Availability:
logtalk_load(code_metrics(loader))

Author: Ebrahim Azarisooreh and Paulo Moura
Version: 0:8:0
Date: 2022-05-05

Compilation flags:
static

Provides:
logtalk::message_prefix_stream/4
logtalk::message_tokens//2

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.34.5 code_metrics_utilities

Internal predicates for analyzing source code.

Availability:
logtalk_load(code_metrics(loader))

Author: Ebrahim Azarisooreh
Version: 0:7:0
Date: 2024-03-28

Compilation flags:
static

Uses:
list
logtalk

Remarks:

- Usage: This is meant to be imported by any metric added to the system.
- Predicate Scope: This is meant for internal use by metrics only. As such, all provided predicates are protected.

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
  - ancestor/4
  - current_entity/1
  - declares_predicate/2
  - defines_predicate/2
  - defines_predicate/3
  - entity_calls/3
  - entity_kind/2
  - entity_property/2
  - entity_updates/3
  - not_excluded_file/3
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

ancestor/4

True if Entity descends from Ancestor, and EntityKind and AncestorKind unify with their respective entity types.

Compilation flags:
static

Template:
ancestor(EntityKind,Entity,AncestorKind,Ancestor)
Mode and number of proofs:
ancestor(?entity,?entity_identifier,?entity,?entity_identifier) - zero_or_more

---

current_entity/1

True if Entity is a currently loaded entity.

Compilation flags:
static

Template:
current_entity(Entity)
Mode and number of proofs:
current_entity(?entity_identifier) - zero_or_more

---

declares_predicate/2

True if Entity declares Predicate internally.

Compilation flags:
static

Template:
declares_predicate(Entity,Predicate)
Mode and number of proofs:
declares_predicate(?entity_identifier,?predicate_indicator) - zero_or_more

---

defines_predicate/2

True if Entity defines an implementation of Predicate internally. Auxiliary predicates are excluded from results.

Compilation flags:
    static

Template:
    defines_predicate(Entity,Predicate)
Mode and number of proofs:
    defines_predicate(?entity_identifier,?predicate_indicator) - zero_or_more

---

defines_predicate/3

Same as defines_predicate/2, except Property is unified with a property of the predicate.

Compilation flags:
    static

Template:
    defines_predicate(Entity,Predicate,Property)
Mode and number of proofs:
    defines_predicate(?entity_identifier,?predicate_indicator,?term) - zero_or_more

---

entity_calls/3

True if a predicate Caller within Entity makes a Call.

Compilation flags:
    static

Template:
    entity_calls(Entity,Caller,Call)
Mode and number of proofs:
    entity_calls(?entity_identifier,?predicate_indicator,?predicate_indicator) - zero_or_one

---

entity_kind/2

True if Kind defines Entity and is one of category, protocol, or object.

Compilation flags:
    static

Template:
    entity_kind(Entity,Kind)
Mode and number of proofs:
    entity_kind(+entity_identifier,-entity) - zero_or_one

---

entity_property/2

True if Property is a valid property of Entity. Entity can be either a category, a protocol, or an object.

Compilation flags:
    static

Template:
    entity_property(Entity,Property)
Mode and number of proofs:
    entity_property(+entity_identifier,-term) - zero_or_more

---

entity_updates/3

True if a predicate Updater within Entity makes a dynamic update to Updated (by using e.g. the asserta/1 or retract/1 predicates).

Compilation flags:
    static

Template:
    entity_updates(Entity,Updater,Updated)

Mode and number of proofs:
entity_updates(+entity_identifier,?predicate_indicator,?predicate_indicator) - zero_or_one

---

not_excluded_file/3

True if the file is not being excluded.

Compilation flags:
static

Template:
not_excluded_file(ExcludedFiles,Path,Basename)
Mode and number of proofs:
not_excluded_file(+list(atom),+atom,+atom) - zero_or_one

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.34.6 cogc_metric

Cognitive complexity metric (approximation). For each non-auxiliary predicate, the score contribution is the number of extra clauses (i.e. number of clauses minus one, for multi-clause branching) plus one if the predicate is directly recursive. The entity score is the sum of all predicate contributions. Protocols are not scored as they cannot define predicates.

Availability:
logtalk_load(code_metrics(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-14

Compilation flags:
static, context_switching_calls

Imports:
public code_metrics_utilities
public code_metric
Provides:
logtalk::message_tokens//2
Uses:
list
logtalk
numberlist

Remarks:

- Limitations: The reflection API does not expose control constructs such as cuts, conditionals, or disjunctions. This metric approximates cognitive complexity using only clause-level branching and direct recursion.
- Branching: A predicate with N clauses contributes N-1 to the score, representing the N-1 extra choices a reader must track.
- Recursion: A predicate that directly calls itself (within the same entity) contributes an additional 1 to the score.
- Score interpretation: The score is represented by a non-negative integer. Higher scores indicate predicates or entities that are harder to understand due to more branching choices or self-recursion. A score of 0 means all predicates are single-clause and non-recursive.
- Aggregation: File, directory, and library scores are computed by summing the individual entity scores.

Inherited public predicates:
all/0 all/1 all_score/1 check_option/1 check_options/1 default_option/1 default_options/1 directory/1 directory/2 directory_score/2 entity/1 entity_score/2 file/1 file/2 file_score/2 format_entity_score//2 library/1 library/2 library_score/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory_score/2 rlibrary/1 rlibrary/2 rlibrary_score/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.34.7 coupling_metric

Computes entity efferent coupling, afferent coupling, instability, abstractness, and distance from the main sequence.

Availability:
logtalk_load(code_metrics(loader))

Author: Ebrahim Azarisooreh and Paulo Moura
Version: 0:15:0
Date: 2026-03-13

Compilation flags:
static, context_switching_calls

Imports:
public code_metrics_utilities
public code_metric
Uses:
list

Remarks:

- Efferent coupling (Ce): Number of entities that an entity depends on.
- Afferent coupling (Ca): Number of entities that depend on an entity.

- Instability (I): Computed as Ce / (Ce + Ca). Measures the entity resilience to change. Ranging from 0 to 1, with 0 indicating a maximally stable entity and 1 indicating a maximally unstable entity. Ideally, an entity is either maximally stable or maximally unstable.
- Abstractness (A): Computed as the ratio between the number of static predicates with scope directives without a local definition and the number of static predicates with scope directives. Measures the rigidity of an entity. Ranging from 0 to 1, with 0 indicating a fully concrete entity and 1 indicating a fully abstract entity.
- Distance from main sequence (D): Computed as abs(A + I - 1). Measures how far an entity is from the idealized line A + I = 1. Ranging from 0 to 1, with 0 indicating the entity is on the main sequence and 1 indicating it is as far as possible (either maximally abstract and stable, or maximally concrete and unstable).
- Entity score: Represented as the compound term ce_ca_i_a_d(Ce,Ca,I,A,D).
- Dependencies count: Includes direct entity relations plus calls or dynamic updates to predicates in external objects or categories.

Inherited public predicates:

all/0 all/1 all_score/1 check_option/1 check_options/1 default_option/1 default_options/1 directory/1 directory/2 directory_score/2 entity/1 entity_score/2 file/1 file/2 file_score/2 format_entity_score//2 library/1 library/2 library_score/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory_score/2 rlibrary/1 rlibrary/2 rlibrary_score/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.34.8 dit_metric

Analyzes the depth of inheritance for objects, protocols, and categories.

Availability:
logtalk_load(code_metrics(loader))

Author: Ebrahim Azarisooreh
Version: 0:6:2
Date: 2026-03-13

Compilation flags:
static, context_switching_calls

Imports:
public code_metrics_utilities
public code_metric
Uses:
numberlist

Remarks:

- Depth: The depth is the maximum length of a node to the root entity. Lower scores are generally better.
- Inheritance: A level of inheritance defined by either one of specialization, instantiation, extension, importation, or implementation.
- Scoring: The maximum path length is determined for each entity in question.

Inherited public predicates:
all/0 all/1 all_score/1 check_option/1 check_options/1 default_option/1 default_options/1 directory/1 directory/2 directory_score/2 entity/1 entity_score/2 file/1 file/2 file_score/2 format_entity_score//2 library/1 library/2 library_score/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory_score/2 rlibrary/1 rlibrary/2 rlibrary_score/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.34.9 doc_metric

Entity and entity predicates documentation score.

Availability:
logtalk_load(code_metrics(loader))

Author: Paulo Moura
Version: 0:13:1
Date: 2026-03-13

Compilation flags:
static, context_switching_calls

Imports:
public code_metrics_utilities
public code_metric
Uses:

list
numberlist

Remarks:

- Score range: Score is a integer percentage where a 100% score means that all expected documentation information is present.
- Score weights: The score is split by default between 20% for the entity documentation and 80% for the entity predicates documentation, Can be customized using the predicate entity_predicates_weights_hook/2.
- Score customization: The individual scores of entity info/1 pairs and predicate info/2 pairs can be customized using the entity_info_pair_score_hook/3 and predicate_info_pair_score_hook/4 predicates.

Inherited public predicates:

all/0 all/1 all_score/1 check_option/1 check_options/1 default_option/1 default_options/1 directory/1 directory/2 directory_score/2 entity/1 entity_score/2 file/1 file/2 file_score/2 format_entity_score//2 library/1 library/2 library_score/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory_score/2 rlibrary/1 rlibrary/2 rlibrary_score/2 valid_option/1 valid_options/1

- Public predicates
  - entity_predicates_weights_hook/2
  - entity_info_score_hook/2
  - entity_info_pair_score_hook/3
  - predicate_mode_score_hook/3
  - predicate_mode_score_hook/5
  - predicate_info_score_hook/3
  - predicate_info_pair_score_hook/4
- Protected predicates
- Private predicates
- Operators

Public predicates

entity_predicates_weights_hook/2

Relative weight between entity documentation and predicates documentation in percentage. The sum of the two values must be equal to 100.

Compilation flags:
dynamic, multifile

Template:
entity_predicates_weights_hook(EntityWeight,PredicatesWeight)
Mode and number of proofs:
entity_predicates_weights_hook(?integer,?integer) - zero_or_one

---

entity_info_score_hook/2

Maximum score for entity info/1 directives.

Compilation flags:
dynamic, multifile

Template:
entity_info_score_hook(Entity,MaximumScore)
Mode and number of proofs:
entity_info_score_hook(?term,?integer) - zero_or_one

---

entity_info_pair_score_hook/3

Score for relevant entity info/1 directive pairs. If defined, the entity_info_score_hook/2 predicate should be defined accordingly.

Compilation flags:
dynamic, multifile

Template:
entity_info_pair_score_hook(Pair,Entity,Score)

Mode and number of proofs:
entity_info_pair_score_hook(?callable,?term,?integer) - zero_or_more

---

predicate_mode_score_hook/3

Maximum score for predicate mode/2 directives.

Compilation flags:
dynamic, multifile

Template:
predicate_mode_score_hook(Entity,Predicate,MaximumScore)
Mode and number of proofs:
predicate_mode_score_hook(?term,?term,?integer) - zero_or_more

---

predicate_mode_score_hook/5

Score for a predicate mode/2 directive. If defined, the predicate_mode_score_hook/3 predicate should be defined accordingly.

Compilation flags:
dynamic, multifile

Template:
predicate_mode_score_hook(Template,Solutions,Entity,Predicate,Score)
Mode and number of proofs:
predicate_mode_score_hook(?term,?term,?term,?term,?integer) - zero_or_one

---

predicate_info_score_hook/3

Maximum score for predicate info/2 directives.

Compilation flags:
dynamic, multifile

Template:
predicate_info_score_hook(Entity,Predicate,MaximumScore)
Mode and number of proofs:
predicate_info_score_hook(?term,?term,?integer) - zero_or_one

---

predicate_info_pair_score_hook/4

Score for a predicate info/2 directive pairs. If defined, the predicate_info_score_hook/3 predicate should be defined accordingly.

Compilation flags:
dynamic, multifile

Template:
predicate_info_pair_score_hook(Pair,Entity,Predicate,Score)
Mode and number of proofs:
predicate_info_pair_score_hook(?callable,?term,?term,?integer) - zero_or_more

---

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.34.10 halstead_metric

Computes Halstead complexity numbers for an entity using a Stroud of 18.

Availability:
logtalk_load(code_metrics(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2018-06-08

Compilation flags:
static, context_switching_calls

Extends:
public halstead_metric(18)

Remarks:
(none)

Inherited public predicates:
all/0 all/1 all_score/1 check_option/1 check_options/1 default_option/1 default_options/1 directory/1 directory/2 directory_score/2 entity/1 entity_score/2 file/1 file/2 file_score/2 format_entity_score//2 library/1 library/2 library_score/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory_score/2 rlibrary/1 rlibrary/2 rlibrary_score/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.34.11 halstead_metric(Stroud)

- Stroud - Coefficient for computing the time required to program.

Computes Halstead complexity numbers for an entity.

Availability:
logtalk_load(code_metrics(loader))

Author: Paulo Moura
Version: 0:10:0
Date: 2025-01-04

Compilation flags:
static, context_switching_calls

Imports:
public code_metrics_utilities
public code_metric
Uses:
list
numberlist
pairs

Remarks:

- Definition of operators: Predicates declared, user-defined, and called are interpreted as operators. Built-in predicates and built-in control constructs are ignored.
- Definition of operands: Predicate arguments are abstracted and interpreted as operands. Note that this definition of operands is a significant deviation from the original definition, which used syntactic literals.
- Pn: Number of distinct predicates (declared, defined, called, or updated).
- PAn: Number of predicate arguments (assumed distinct).
- Cn: Number of predicate calls/updates + number of clauses.
- CAn: Number of predicate call/update arguments + number of clause head arguments.
- EV: Entity vocabulary: EV = Pn + PAn.
- EL: Entity length: EL = Cn + CAn.
- V: Volume: V = EL * log2(EV).
- D: Difficulty: D = (Pn/2) * (CAn/An).
- E: Effort: E = D * V.
- T: Time required to program: T = E/k seconds (k is the Stroud number; defaults to 18).
- B: Number of delivered bugs: B = V/3000.
- Entity score: Represented as the compound term pn_pan_cn_can_ev_el_v_d_e_t_b/11.

Inherited public predicates:

all/0 all/1 all_score/1 check_option/1 check_options/1 default_option/1 default_options/1 directory/1 directory/2 directory_score/2 entity/1 entity_score/2 file/1 file/2 file_score/2 format_entity_score//2 library/1 library/2 library_score/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory_score/2 rlibrary/1 rlibrary/2 rlibrary_score/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.34.12 lcom_metric

Lack of Cohesion Of Methods metric (LCOM4).

Availability:
logtalk_load(code_metrics(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-30

Compilation flags:
static, context_switching_calls

Imports:
public code_metrics_utilities
public code_metric
Provides:
logtalk::message_tokens//2
Uses:
list
logtalk

Remarks:

- Score computation: Computes the number of connected components in the undirected graph whose nodes are the locally defined predicates and whose edges represent direct internal calls between them.
- Score interpretation: A score of 1 indicates a fully cohesive entity. Higher scores indicate that the entity may benefit from being split. Protocols are not scored as they cannot define predicates.

- Score representation: The score is represented by the compound term lcom(Components, Predicates) where Components is the number of connected components and Predicates is the total number of locally defined (non-auxiliary) predicates.

Inherited public predicates:

all/0 all/1 all_score/1 check_option/1 check_options/1 default_option/1 default_options/1 directory/1 directory/2 directory_score/2 entity/1 entity_score/2 file/1 file/2 file_score/2 format_entity_score//2 library/1 library/2 library_score/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory_score/2 rlibrary/1 rlibrary/2 rlibrary_score/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.34.13 lines_metric

Computes the number of lines of code, comments, and blanks by calling cloc and parsing its report file output.

Availability:

logtalk_load(code_metrics(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-17

Compilation flags:
static, context_switching_calls

Imports:
public code_metrics_utilities
public code_metric
Provides:
logtalk::message_tokens//2
Uses:
atom
list
logtalk
os
user

Remarks:

- Entity line range: Entity scores are computed by querying entity lines(BeginLine, EndLine) and running cloc on a temporary file containing only that line range.
- External dependency: Requires cloc to be available in PATH.
- Entity score: Represented as the compound term lines(Code, Comments, Blanks).

Inherited public predicates:
all/0 all/1 all_score/1 check_option/1 check_options/1 default_option/1 default_options/1 directory/1 directory/2 directory_score/2 entity/1 entity_score/2 file/1 file/2 file_score/2 format_entity_score//2 library/1 library/2 library_score/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory_score/2 rlibrary/1 rlibrary/2 rlibrary_score/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.34.14 mi_metric

Computes the maintainability index metric based on the Cyclomatic Complexity and the Halstead Volume metrics an entity using a Stroud of 18.

Availability:
logtalk_load(code_metrics(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-14

Compilation flags:
static, context_switching_calls

Extends:
public mi_metric(18)

Remarks:
(none)

Inherited public predicates:
all/0 all/1 all_score/1 check_option/1 check_options/1 default_option/1 default_options/1 directory/1 directory/2 directory_score/2 entity/1 entity_score/2 file/1 file/2 file_score/2 format_entity_score//2 library/1 library/2 library_score/2 option/2 option/3 rdirectory/1

rdirectory/2 rdirectory_score/2 rlibrary/1 rlibrary/2 rlibrary_score/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.34.15 mi_metric(Stroud)

- Stroud - Halstead metric coefficient for computing the time required to program.

Computes the maintainability index metric based on the Cyclomatic Complexity, Halstead Volume, and lines of code metrics for an entity.

Availability:
logtalk_load(code_metrics(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-14

Compilation flags:
static, context_switching_calls

Imports:

public code_metrics_utilities

public code_metric

Uses:

cc_metric

halstead_metric(Stroud)

lines_metric

Remarks:

- Formula: The original equation is used: MI = 171 - 5.2 * ln(V) - 0.23 * C - 16.2 * ln(L). V is the Halstead volume, C is the Cyclomatic Complexity, and L is the number of code lines.
- Entity score: Represented as the compound term mi(MI).

Inherited public predicates:

all/0 all/1 all_score/1 check_option/1 check_options/1 default_option/1 default_options/1 directory/1 directory/2 directory_score/2 entity/1 entity_score/2 file/1 file/2 file_score/2 format_entity_score//2 library/1 library/2 library_score/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory_score/2 rlibrary/1 rlibrary/2 rlibrary_score/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.34.16 noc_metric

Number of entity clauses metric. The score is represented using the compound term number_of_clauses(Total, User).

Availability:
logtalk_load(code_metrics(loader))

Author: Ebrahim Azarisooreh and Paulo Moura
Version: 0:14:2
Date: 2026-03-13

Compilation flags:
static, context_switching_calls

Imports:
public code_metrics_utilities
public code_metric
Provides:
logtalk::message_tokens//2
Uses:
list
logtalk

Remarks:
(none)

Inherited public predicates:
all/0 all/1 all_score/1 check_option/1 check_options/1 default_option/1 default_options/1 directory/1 directory/2 directory_score/2 entity/1 entity_score/2 file/1 file/2 file_score/2 format_entity_score//2 library/1 library/2 library_score/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory_score/2 rlibrary/1 rlibrary/2 rlibrary_score/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.34.17 nor_metric

Number of entity rules metric. The score is represented using the compound term number_of_rules(Total, User).

Availability:
logtalk_load(code_metrics(loader))

Author: Paulo Moura
Version: 0:5:2
Date: 2026-03-13

Compilation flags:
static, context_switching_calls

Imports:
public code_metrics_utilities
public code_metric

Provides:
logtalk::message_tokens//2
Uses:
list
logtalk

Remarks:
(none)

Inherited public predicates:
all/0 all/1 all_score/1 check_option/1 check_options/1 default_option/1 default_options/1 directory/1 directory/2 directory_score/2 entity/1 entity_score/2 file/1 file/2 file_score/2 format_entity_score//2 library/1 library/2 library_score/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory_score/2 rlibrary/1 rlibrary/2 rlibrary_score/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.34.18 rfc_metric

Response For a Class (RFC) metric. The score is the number of distinct predicates in the response set: the locally defined (non-auxiliary) predicates plus all distinct predicates they directly call or update. Protocols are not scored as they cannot define predicates. The score is represented by a non-negative integer.

Availability:
logtalk_load(code_metrics(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-13

Compilation flags:
static, context_switching_calls

Imports:
public code_metrics_utilities
public code_metric
Provides:
logtalk::message_tokens//2
Uses:
list
logtalk
numberlist

Remarks:

- Response set: The response set RS(E) for an entity E is the union of its locally defined (non-auxiliary) predicates and all predicates directly called or updated by those predicates.
- Score interpretation: Higher scores indicate entities with more potential execution paths in response to a message, which may increase testing effort.
- Aggregation: When computing scores for files, directories, and libraries, the individual entity scores are summed.

Inherited public predicates:
all/0 all/1 all_score/1 check_option/1 check_options/1 default_option/1 default_options/1 directory/1 directory/2 directory_score/2 entity/1 entity_score/2 file/1 file/2 file_score/2 format_entity_score//2 library/1 library/2 library_score/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory_score/2 rlibrary/1 rlibrary/2 rlibrary_score/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.34.19 size_metric

Source code size metric. Returned scores are upper bounds and based solely in source file sizes (expressed in bytes).

Availability:
logtalk_load(code_metrics(loader))

Author: Paulo Moura
Version: 0:7:2
Date: 2026-03-13

Compilation flags:
static, context_switching_calls

Imports:
public code_metrics_utilities
public code_metric
Provides:

logtalk::message_tokens//2

Uses:

list
logtalk
numberlist
os

Remarks:

(none)

Inherited public predicates:

all/0 all/1 all_score/1 check_option/1 check_options/1 default_option/1 default_options/1 directory/1 directory/2 directory_score/2 entity/1 entity_score/2 file/1 file/2 file_score/2 format_entity_score//2 library/1 library/2 library_score/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory_score/2 rlibrary/1 rlibrary/2 rlibrary_score/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.34.20 upn_metric

Number of unique predicates nodes metric. The nodes include called and updated predicates independently of where they are defined. The score is represented by a non-negative integer.

Availability:
logtalk_load(code_metrics(loader))

Author: Paulo Moura
Version: 0:6:3
Date: 2026-03-13

Compilation flags:
static, context_switching_calls

Imports:
public code_metrics_utilities
public code_metric
Provides:
logtalk::message_tokens//2
Uses:
list
logtalk
numberlist

Remarks:
(none)

Inherited public predicates:
all/0 all/1 all_score/1 check_option/1 check_options/1 default_option/1 default_options/1 directory/1 directory/2 directory_score/2 entity/1 entity_score/2 file/1 file/2 file_score/2 format_entity_score//2 library/1 library/2 library_score/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory_score/2 rlibrary/1 rlibrary/2 rlibrary_score/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.34.21 wmc_metric

Weighted Methods per Class (WMC) metric. Uses unit weights, i.e., the score is the number of locally defined (non-auxiliary) predicates. Protocols are not scored as they cannot define predicates. The score is represented by a non-negative integer.

Availability:
logtalk_load(code_metrics(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-13

Compilation flags:
static, context_switching_calls

Imports:
public code_metrics_utilities
public code_metric

Provides:
logtalk::message_tokens//2
Uses:
list
logtalk
numberlist

Remarks:

- Unit weights: Each predicate is assigned a weight of 1. A cyclomatic-complexity-weighted variant is not provided as the Logtalk reflection API does not expose intra-clause branching structure (if-then-else, disjunction), which would produce the same approximations as the cc_metric.
- Interpretation: Higher scores indicate an entity with more responsibilities that may benefit from being split.
- Aggregation: When computing scores for files, directories, and libraries, the individual entity scores are summed.

Inherited public predicates:
all/0 all/1 all_score/1 check_option/1 check_options/1 default_option/1 default_options/1 directory/1 directory/2 directory_score/2 entity/1 entity_score/2 file/1 file/2 file_score/2 format_entity_score//2 library/1 library/2 library_score/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory_score/2 rlibrary/1 rlibrary/2 rlibrary_score/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.35 colley_ranker

object

1.35.1 colley_ranker

Colley pairwise preference ranker. Learns one deterministic rating per item from a dataset object implementing the pairwise_ranking_dataset_protocol protocol by solving the Colley linear system built from aggregated pairwise outcomes and returns a self-describing ranker term with diagnostics that can be used for ranking and export.

Availability:
logtalk_load(colley_ranker(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public ranking_dataset_common
public score_ranker_model_common
public pairwise_strength_ranker_common
Uses:
avltree
list
numberlist

Remarks:
(none)

Inherited public predicates:

check_option/1 check_options/1 check_ranker/1 default_option/1 default_options/1
diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 grouped_dataset_groups/2
grouped_dataset_items/2 grouped_dataset_items/3 grouped_dataset_summary/2 learn/2
learn/3 option/2 option/3 pairwise_dataset_connected_components/2
pairwise_dataset_declared_items/2 pairwise_dataset_items/2 pairwise_dataset_matchups/2
pairwise_dataset_preferences/2 pairwise_dataset_summary/2 pairwise_dataset_win_totals/2
pairwise_measurement_dataset_connected_components/2
pairwise_measurement_dataset_declared_items/2 pairwise_measurement_dataset_items/2
pairwise_measurement_dataset_measurements/2 pairwise_measurement_dataset_summary/2
print_ranker/1 rank/3 ranker_options/2 scores/2
temporal_pairwise_dataset_connected_components/2
temporal_pairwise_dataset_declared_items/2 temporal_pairwise_dataset_games/2
temporal_pairwise_dataset_games/3 temporal_pairwise_dataset_items/2
temporal_pairwise_dataset_periods/2 temporal_pairwise_dataset_summary/2 valid_option/1
valid_options/1 valid_ranker/1 validate_grouped_dataset/1 validate_grouped_dataset/2
validate_pairwise_dataset/1 validate_pairwise_dataset/2
validate_pairwise_measurement_dataset/1 validate_pairwise_measurement_dataset/2
validate_temporal_pairwise_dataset/1 validate_temporal_pairwise_dataset/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

pairwise_ranking_dataset_protocol, ranker_protocol, copeland_ranker, rank_centrality

1.36 combinations

object

1.36.1 combinations

Implementation of combinations operations over lists.

Availability:
logtalk_load(combinations(loader))

Author: Paulo Moura
Version: 2:0:0
Date: 2026-05-12

Compilation flags:
static, context_switching_calls

Implements:
public combinations_protocol
Uses:
fast_random(Algorithm)
list
natural

Remarks:
(none)

Inherited public predicates:
combination/3 combination/4 combination_index/4 combination_index/5 combinations/3 combinations/4 count_combinations/3 count_distinct_combinations/3 distinct_combination/3 distinct_combination/4 distinct_combination_index/4 distinct_combinations/3 distinct_combinations/4 next_combination/3 nth_combination/4 nth_combination/5 nth_distinct_combination/4 previous_combination/3 random_combination/3 random_distinct_combination/3 sample_combinations/4 sample_distinct_combinations/4

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.36.2 combinations_protocol

Protocol for combinations operations over lists.

Availability:
logtalk_load(combinations(loader))

Author: Paulo Moura
Version: 2:0:0
Date: 2026-05-12

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - combinations/3
  - combination/3
  - combinations/4
  - combination/4
  - distinct_combinations/3
  - distinct_combination/3
  - distinct_combinations/4
  - distinct_combination/4
  - nth_combination/4
  - nth_combination/5
  - combination_index/4
  - combination_index/5
  - count_combinations/3
  - count_distinct_combinations/3
  - nth_distinct_combination/4
  - distinct_combination_index/4
  - random_combination/3
  - sample_combinations/4
  - random_distinct_combination/3
  - sample_distinct_combinations/4
  - next_combination/3
  - previous_combination/3
- Protected predicates
- Private predicates
- Operators

Public predicates

combinations/3

Generates all K-element combinations of a list using default order.

Compilation flags:
    static

Template:
combinations(K,List,Combinations)
Mode and number of proofs:
combinations(+integer,+list,-list) - one

---

combination/3

True iff the third argument is a K-element combination of a list using default order.

Compilation flags:
static

Template:
combination(K,List,Combination)
Mode and number of proofs:
combination(+integer,+list,-list) - one_or_more

---

combinations/4

Generates all K-element combinations with the given order: default, lexicographic, or shortlex.

Compilation flags:
static

Template:
combinations(K,List,Order,Combinations)
Mode and number of proofs:
combinations(+integer,+list,+atom,-list) - one

---

combination/4

True iff the fourth argument is a K-element combination with the given order: default, lexicographic, or shortlex.

Compilation flags:
    static

Template:
    combination(K,List,Order,Combination)
Mode and number of proofs:
    combination(+integer,+list,+atom,-list) - one_or_more

---

distinct_combinations/3

Generates all distinct K-element combinations of a list (deduplicating equal-valued combinations) using default order.

Compilation flags:
    static

Template:
    distinct_combinations(K,List,Combinations)
Mode and number of proofs:
    distinct_combinations(+integer,+list,-list) - one

---

distinct_combination/3

True iff the third argument is a distinct K-element combination of a list using default order.

Compilation flags:
    static

Template:
    distinct_combination(K,List,Combination)
Mode and number of proofs:
    distinct_combination(+integer,+list,-list) - one_or_more

---

distinct_combinations/4

Generates all distinct K-element combinations with the given order: default, lexicographic, or shortlex.

Compilation flags:
    static

Template:
    distinct_combinations(K,List,Order,Combinations)
Mode and number of proofs:
    distinct_combinations(+integer,+list,+atom,-list) - one

---

distinct_combination/4

True iff the fourth argument is a distinct K-element combination with the given order: default, lexicographic, or shortlex.

Compilation flags:
    static

Template:
    distinct_combination(K,List,Order,Combination)
Mode and number of proofs:
    distinct_combination(+integer,+list,+atom,-list) - one_or_more

---

nth_combination/4

Returns the combination at a given zero-based index using default order.

Compilation flags:
    static

Template:
    nth_combination(K,List,Index,Combination)

Mode and number of proofs:
nth_combination(+integer,+list,+integer,-list) - zero_or_one

---

nth_combination/5

Returns the combination at a given zero-based index in the given order: default, lexicographic, or shortlex.

Compilation flags:
static

Template:
nth_combination(K,List,Order,Index,Combination)
Mode and number of proofs:
nth_combination(+integer,+list,+atom,+integer,-list) - zero_or_one

---

combination_index/4

Returns the zero-based index of a combination using default order.

Compilation flags:
static

Template:
combination_index(K,List,Combination,Index)
Mode and number of proofs:
combination_index(+integer,+list,+list,-integer) - zero_or_one

---

combination_index/5

Returns the zero-based index of a combination in the given order: default, lexicographic, or shortlex.

Compilation flags:
static

Template:
combination_index(K,List,Order,Combination,Index)
Mode and number of proofs:
combination_index(+integer,+list,+atom,+list,-integer) - zero_or_one

---

count_combinations/3

Counts the number of K-element combinations of a list.

Compilation flags:
static

Template:
count_combinations(K,List,Count)
Mode and number of proofs:
count_combinations(+integer,+list,-integer) - one

---

count_distinct_combinations/3

Counts the number of distinct K-element combinations of a list (deduplicating equal-valued combinations).

Compilation flags:
static

Template:
count_distinct_combinations(K,List,Count)
Mode and number of proofs:
count_distinct_combinations(+integer,+list,-integer) - one

---

nth_distinct_combination/4

Returns the distinct combination at a given zero-based index in default generation order.

Compilation flags:
    static

Template:
    nth_distinct_combination(K,List,Index,Combination)
Mode and number of proofs:
    nth_distinct_combination(+integer,+list,+integer,-list) - zero_or_one

---

distinct_combination_index/4

Returns the zero-based index of a distinct combination in default generation order.

Compilation flags:
    static

Template:
    distinct_combination_index(K,List,Combination,Index)
Mode and number of proofs:
    distinct_combination_index(+integer,+list,+list,-integer) - zero_or_one

---

random_combination/3

Returns a random K-element combination of a list.

Compilation flags:
    static

Template:
    random_combination(K,List,Combination)
Mode and number of proofs:
    random_combination(+integer,+list,-list) - zero_or_one

---

sample_combinations/4

Returns SampleCount random K-element combinations of a list, sampled with replacement.

Compilation flags:
    static

Template:
    sample_combinations(K,List,SampleCount,Samples)
Mode and number of proofs:
    sample_combinations(+integer,+list,+integer,-list) - zero_or_one

---

random_distinct_combination/3

Returns a random distinct K-element combination of a list (deduplicating equal-valued combinations).

Compilation flags:
    static

Template:
    random_distinct_combination(K,List,Combination)
Mode and number of proofs:
    random_distinct_combination(+integer,+list,-list) - zero_or_one

---

sample_distinct_combinations/4

Returns SampleCount random distinct K-element combinations of a list, sampled with replacement after deduplicating equal-valued combinations.

Compilation flags:
    static

Template:
    sample_distinct_combinations(K,List,SampleCount,Samples)
Mode and number of proofs:
    sample_distinct_combinations(+integer,+list,+integer,-list) - zero_or_one

---

next_combination/3

Returns the next distinct combination value in lexicographic order induced by the first argument.

Compilation flags:
    static

Template:
    next_combination(List,Combination,Next)
Mode and number of proofs:
    next_combination(+list,+list,-list) - zero_or_one

---

previous_combination/3

Returns the previous distinct combination value in lexicographic order induced by the first argument.

Compilation flags:
    static

Template:
    previous_combination(List,Combination,Previous)
Mode and number of proofs:
    previous_combination(+list,+list,-list) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.37 command_line_options

category

1.37.1 command_line_option

Category for defining command-line options. Import this category into objects that represent individual command-line options and override the predicates as needed.

Availability:
logtalk_load(command_line_options(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-20

Compilation flags:
static

Uses:
type

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - check/0
  - valid/0

    - name/1
    - short_flags/1
    - long_flags/1
    - type/1
    - default/1
    - meta/1
    - help/1
- Protected predicates
- Private predicates
- Operators

Public predicates

check/0

Checks if the command-line option definition is valid. Throws an error if the definition is invalid.

Compilation flags:
    static

Mode and number of proofs:
    check - one_or_error

---

valid/0

Succeeds if the command-line option definition is valid. Fails otherwise.

Compilation flags:
    static

Mode and number of proofs:
    valid - zero_or_one

---

name/1

Name used to identify this option in the parsed results. This predicate must be overridden. No default.

Compilation flags:
    static

Template:
    name(Name)
Mode and number of proofs:
    name(?atom) - zero_or_one

---

short_flags/1

List of single-character short flags for this option (e.g., [v] for -v). Default is an empty list.

Compilation flags:
    static

Template:
    short_flags(Flags)
Mode and number of proofs:
    short_flags(-list(character)) - one

---

long_flags/1

List of long flags for this option (e.g., [verbose] for --verbose). Default is an empty list.

Compilation flags:
    static

Template:
    long_flags(Flags)
Mode and number of proofs:
    long_flags(-list(atom)) - one

---

type/1

Option value type. One of boolean, atom, integer, float, or term. Default is term.

Compilation flags:
    static

Template:
    type(Type)
Mode and number of proofs:
    type(-atom) - one

---

default/1

Default value for this option if any.

Compilation flags:
    static

Template:
    default(Default)
Mode and number of proofs:
    default(-term) - zero_or_one

---

meta/1

Metasyntactic variable name for the help text (e.g., 'FILE'). Default is an empty atom.

Compilation flags:
    static

Template:
    meta(Meta)
Mode and number of proofs:
    meta(-atom) - one

---

help/1

Help text for this option. Can be an atom or a list of atoms for pre-broken lines. Default is an empty atom.

Compilation flags:
    static

Template:
    help(Help)
Mode and number of proofs:
    help(-atom) - one
    help(-list(atom)) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

command_line_options

object

1.37.2 command_line_options

Command line options parsing predicates. Uses object-based option specifications with the command_line_option category.

Availability:
    logtalk_load(command_line_options(loader))

Author: Marcus Uneson and Paulo Moura
Version: 1:0:0

Date: 2026-02-20

Compilation flags:
static, context_switching_calls

Imports:
public options
Uses:
atom
list
meta
numberlist
term_io
type
user

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 default_option/1 default_options/1 option/2 option/3 valid_option/1 valid_options/1

- Public predicates
  - parse/4
  - parse/5
  - help/2
  - help/3
- Protected predicates
- Private predicates
- Operators

Public predicates

parse/4

Parses the arguments ApplArguments according to the option objects OptionObjects using default parsing options.

Compilation flags:
    static

Template:
    parse(OptionObjects,ApplArguments,Options,PositionalArguments)
Mode and number of proofs:
    parse(+list(object),+list(atom),-list,-list(atom)) - one_or_error

---

parse/5

Parses the arguments ApplArguments according to the option objects OptionObjects and the parsing options ParseOptions. Options is a list of parsed options as Name(Value) terms by default (or Func(Name,Value) when the output_functor(Func) parse option is used). PositionalArguments are the remaining non-dashed arguments. ParseOptions include output_functor(Func), duplicated_flags(Keep) (one of keepfirst, keeplast, keepall; default keeplast), and allow_empty_flag_spec(Bool) (default true).

Compilation flags:
    static

Template:
    parse(OptionObjects,ApplArguments,Options,PositionalArguments,ParseOptions)
Mode and number of proofs:
    parse(+list(object),+list(atom),-list,-list(atom),+list) - one_or_error

---

help/2

Synthesizes a help text Help as an atom from the option objects OptionObjects using default help options.

Compilation flags:
    static

Template:
    help(OptionObjects,Help)
Mode and number of proofs:
    help(+list(object),-atom) - one_or_error

---

help/3

Synthesizes a help text Help as an atom from the option objects OptionObjects using the given HelpOptions. HelpOptions include line_width(Width) (default 80), min_help_width(Width) (default 40), break_long_flags(Boolean) (default false), and suppress_empty_meta(Boolean) (default true).

Compilation flags:
    static

Template:
    help(OptionObjects,Help,HelpOptions)
Mode and number of proofs:
    help(+list(object),-atom,+list) - one_or_error

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

command_line_option

1.38 copeland_ranker

object

1.38.1 copeland_ranker

Copeland pairwise preference ranker. Learns one deterministic score per item from a dataset object implementing the pairwise_ranking_dataset_protocol protocol by comparing aggregated head-to-head outcomes for each observed matchup and returns a self-describing ranker term with diagnostics that can be used for ranking and export.

Availability:
logtalk_load(copeland_ranker(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public ranking_dataset_common
public score_ranker_model_common

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_ranker/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 grouped_dataset_groups/2 grouped_dataset_items/2 grouped_dataset_items/3 grouped_dataset_summary/2 learn/2 learn/3 option/2 option/3 pairwise_dataset_connected_components/2 pairwise_dataset_declared_items/2 pairwise_dataset_items/2 pairwise_dataset_matchups/2 pairwise_dataset_preferences/2 pairwise_dataset_summary/2 pairwise_dataset_win_totals/2

pairwise_measurement_dataset_connected_components/2
pairwise_measurement_dataset_declared_items/2 pairwise_measurement_dataset_items/2
pairwise_measurement_dataset_measurements/2 pairwise_measurement_dataset_summary/2
print_ranker/1 rank/3 ranker_options/2 scores/2
temporal_pairwise_dataset_connected_components/2
temporal_pairwise_dataset_declared_items/2 temporal_pairwise_dataset_games/2
temporal_pairwise_dataset_games/3 temporal_pairwise_dataset_items/2
temporal_pairwise_dataset_periods/2 temporal_pairwise_dataset_summary/2 valid_option/1
valid_options/1 valid_ranker/1 validate_grouped_dataset/1 validate_grouped_dataset/2
validate_pairwise_dataset/1 validate_pairwise_dataset/2
validate_pairwise_measurement_dataset/1 validate_pairwise_measurement_dataset/2
validate_temporal_pairwise_dataset/1 validate_temporal_pairwise_dataset/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

pairwise_ranking_dataset_protocol, ranker_protocol, bradley_terry_ranker

1.39 core

category

1.39.1 core_messages

Logtalk core (compiler and runtime) default message tokenization.

Availability:
built_in

Author: Paulo Moura
Version: 1:147:0
Date: 2026-05-04

Compilation flags:
static

Provides:
logtalk::message_prefix_stream/4
logtalk::message_tokens//2

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.39.2 expanding

Term and goal expansion protocol.

Availability:
built_in

Author: Paulo Moura
Version: 1:1:0
Date: 2016-07-12

Compilation flags:
static, built_in

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - goal_expansion/2
  - term_expansion/2
- Protected predicates
- Private predicates
- Operators

Public predicates

goal_expansion/2

Defines a goal expansion. Called recursively until a fixed point is reached on goals found while compiling a source file (except for goals wrapped using the {}/1 compiler bypass control construct).

Compilation flags:
    static

Template:
    goal_expansion(Goal,ExpandedGoal)
Mode and number of proofs:
    goal_expansion(+callable,-callable) - zero_or_one

---

term_expansion/2

Defines a term expansion. Called until it succeeds on all terms read while compiling a source file (except for terms skipped by using the conditional compilation directives or wrapped using the {}/1 compiler bypass control construct).

Compilation flags:
    static

Template:
    term_expansion(Term,ExpandedTerms)
Mode and number of proofs:
    term_expansion(+term,-term) - zero_or_one
    term_expansion(+term,-list(term)) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

protocol

1.39.3 forwarding

Message forwarding protocol.

Availability:
built_in

Author: Paulo Moura
Version: 1:0:1
Date: 2025-05-15

Compilation flags:
static, built_in

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - forward/1
- Protected predicates

- Private predicates
- Operators

Public predicates

forward/1

User-defined message forwarding handler, automatically called (if defined) by the runtime for any message that the receiving object does not understand.

Compilation flags:
    static

Template:
    forward(Message)
Mode and number of proofs:
    forward(+callable) - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

object

1.39.4 logtalk

Built-in object providing message printing, debugging, library, source file, halt predicates, and hacking methods.

Availability:
    built_in

Author: Paulo Moura
Version: 3:5:0
Date: 2026-05-04

Compilation flags:
    static, built_in, context_switching_calls, threaded

Dependencies:
    (none)

Remarks:

- Default message kinds: silent, silent(Key), banner, help, comment, comment(Key), information, information(Key), warning, warning(Key), error, error(Key), debug, debug(Key), question, and question(Key).
- Printing of silent messages: By default, silent messages are not printed. These messages are only useful when intercepted.
- Printing of banner and comment messages: By default, banner and comment messages are only printed when the report flag is turned on.
- Printing of help, information, and question messages: These messages are always printed by default as they provide requested output.
- Printing of warning messages: By default, warning messages are not printed when the report flag is turned off.
- Printing of error messages: These messages are always printed by default.
- Printing of debug messages: By default, debug messages are only printed when the debug flag is turned on. The compiler suppresses debug message printing goals when compiling in optimized mode.
- Meta messages: A meta message is a message that have another message as argument and is typically used for debugging messages. Meta messages avoid the need of defining tokenizer rules for every message but can be intercepted as any other message.
- Meta message @Message: By default, the message is printed as passed to the write/1 predicate followed by a newline.
- Meta message Key-Value: By default, the message is printed as "Key: Value" followed by a newline. The key is printed as passed to the write/1 predicate while the value is printed as passed to the writeq/1 predicate.
- Meta message Format+Arguments: By default, the message is printed as passed to the format/2 predicate.
- Meta message List: By default, the list items are printed indented one per line. The items are preceded by a dash and can be @Message, Key-Value, or Format+Arguments messages. If that is not the case, the item is printed as passed to the writeq/1 predicate.
- Meta message Title::List: By default, the title is printed followed by a newline and the indented list items, one per line. The items are printed as in the List meta message.

- Meta message [Stream,Prefix]>>Goal: By default, call user-defined Goal in the context of user. The use of a lambda expression allows passing the message stream and prefix. Printing the prefix is delegated to the goal.
- Meta message [Stream]>>Goal: By default, call user-defined Goal in the context of user. The use of a lambda expression allows passing the message stream.
- Message tokens: at_same_line, tab(Expression), nl, flush, Format-Arguments, term(Term,Options), ansi(Attributes,Format,Arguments), begin(Kind,Variable), and end(Variable).
- Multi-threading applications: Predicates calling methods such as print_message/3, ask_question/5, or compile_aux_clauses/1 may need to be declared synchronized in order to avoid race conditions.

Inherited public predicates:
(none)

- Public predicates
  - print_message/3
  - print_message_tokens/3
  - print_message_token/4
  - message_tokens//2
  - message_prefix_stream/4
  - message_prefix_file/6
  - message_hook/4
  - ask_question/5
  - question_hook/6
  - question_prompt_stream/4
  - trace_event/2
  - debug_handler/1
  - active_debug_handler/1
  - activate_debug_handler/1
  - deactivate_debug_handler/0
  - debug_handler/3
  - expand_library_path/2
  - loaded_file/1
  - loaded_file_property/2
  - loaded_files_topological_sort/1
  - loaded_files_topological_sort/2
  - file_type_extension/2

    - halt/1
    - halt/0
    - halt_hook/2
    - compile_aux_clauses/1
    - entity_prefix/2
    - compile_predicate_heads/4
    - compile_predicate_indicators/3
    - decompile_predicate_heads/4
    - decompile_predicate_indicators/4
    - execution_context/7
- Protected predicates
- Private predicates
    - active_debug_handler_/1
- Operators

Public predicates

print_message/3

Prints a message of the given kind for the specified component.

Compilation flags:
    static

Template:
    print_message(Kind,Component,Message)
Mode and number of proofs:
    print_message(+nonvar,+nonvar,+nonvar) - one

print_message_tokens/3

Print the messages tokens to the given stream, prefixing each line with the specified atom.

Compilation flags:
    static

Template:
    print_message_tokens(Stream,Prefix,Tokens)
Mode and number of proofs:
    print_message_tokens(@stream_or_alias,+atom,@list(nonvar)) - one

---

print_message_token/4

User-defined hook predicate for printing a message token (see this object remarks).

Compilation flags:
    dynamic, multifile

Template:
    print_message_token(Stream,Prefix,Token,Tokens)
Mode and number of proofs:
    print_message_token(@stream_or_alias,@atom,@nonvar,@list(nonvar)) - zero_or_one

---

message_tokens//2

User-defined hook grammar rule for converting a message into a list of tokens (see this object remarks).

Compilation flags:
    dynamic, multifile

Template:
    message_tokens(Message,Component)
Mode and number of proofs:
    message_tokens(+nonvar,+nonvar) - zero_or_one

---

message_prefix_stream/4

Message line prefix and output stream to be used when printing a message given its kind and component.

Compilation flags:
    dynamic, multifile

Template:
    message_prefix_stream(Kind,Component,Prefix,Stream)
Mode and number of proofs:
    message_prefix_stream(?nonvar,?nonvar,?atom,?stream_or_alias) - zero_or_more

---

message_prefix_file/6

Message line prefix and output file to be used when printing a message given its kind and component.

Compilation flags:
    dynamic, multifile

Template:
    message_prefix_file(Kind,Component,Prefix,File,Mode,Options)
Mode and number of proofs:
    message_prefix_file(?nonvar,?nonvar,?atom,?atom,?atom,?list(compound)) - zero_or_more

---

message_hook/4

User-defined hook predicate for intercepting message printing calls.

Compilation flags:
    dynamic, multifile

Template:
    message_hook(Message,Kind,Component,Tokens)
Mode and number of proofs:
    message_hook(+nonvar,+nonvar,+nonvar,+list(nonvar)) - zero_or_one

---

ask_question/5

Asks a question and reads the answer until the check predicate is true.

Compilation flags:
    static

Template:
    ask_question(Kind,Component,Question,Check,Answer)
Meta-predicate template:
    ask_question(*,*,*,1,*)
Mode and number of proofs:
    ask_question(+nonvar,+nonvar,+nonvar,+callable,-term) - one

---

question_hook/6

User-defined hook predicate for intercepting question asking calls.

Compilation flags:
    dynamic, multifile

Template:
    question_hook(Question,Kind,Component,Tokens,Check,Answer)
Meta-predicate template:
    question_hook(*,*,*,*,1,*)
Mode and number of proofs:
    question_hook(+nonvar,+nonvar,+nonvar,+list(nonvar),+callable,-term) - zero_or_one

---

question_prompt_stream/4

Prompt and input stream to be used when asking a question given its kind and component.

Compilation flags:
    dynamic, multifile

Template:
    question_prompt_stream(Kind,Component,Prompt,Stream)

Mode and number of proofs:
question_prompt_stream(?nonvar,?nonvar,?atom,?stream_or_alias) - zero_or_more

---

trace_event/2

Trace event handler. The runtime calls all trace event handlers using a failure-driven loop before calling the debug event handler.

Compilation flags:
dynamic, multifile

Template:
trace_event(Event,ExecutionContext)
Mode and number of proofs:
trace_event(@callable,@execution_context) - zero

Remarks:

- Unification events: Generated after a successful unification with a fact - fact(Entity,Fact,Clause,File,Line) - or a rule head - rule(Entity,Head,Clause,File,Line).
- Goal events: Generated when calling a goal: top_goal(Goal,CompiledGoal) or goal(Goal,CompiledGoal).

---

debug_handler/1

Enumerates, by backtracking, all declared debug handler providers. Define a clause for this predicate to declare a new debug handler provider.

Compilation flags:
static, multifile

Template:
debug_handler(Provider)
Mode and number of proofs:
debug_handler(?object_identifier) - zero_or_more
debug_handler(?category_identifier) - zero_or_more

---

active_debug_handler/1

Current active debug handler provider if any. There is at most one active debug handler provider at any given moment.

Compilation flags:
    static

Template:
    active_debug_handler(Provider)
Mode and number of proofs:
    active_debug_handler(?category_identifier) - zero_or_one
    active_debug_handler(?category_identifier) - zero_or_one

---

activate_debug_handler/1

Activates the given debug handler provider. There is at most one active debug handler provider at any given moment. Fails if the object or category is not declared as a debug handler provider.

Compilation flags:
    static

Template:
    activate_debug_handler(Provider)
Mode and number of proofs:
    activate_debug_handler(@object_identifier) - zero_or_one
    activate_debug_handler(@category_identifier) - zero_or_one

---

deactivate_debug_handler/0

Deactivates the current debug handler provider if any.

Compilation flags:
    static

Mode and number of proofs:
    deactivate_debug_handler - one

debug_handler/3

Debug event handler. Called by the runtime when the given provider is active.

Compilation flags:
    static, multifile

Template:
    debug_handler(Provider,Event,ExecutionContext)
Mode and number of proofs:
    debug_handler(+object_identifier,+callable,+execution_context) - zero_or_more
    debug_handler(+category_identifier,+callable,+execution_context) - zero_or_more

Remarks:

- Unification events: Generated after a successful unification with a fact - fact(Entity,Fact,Clause,File,Line) - or a rule head - rule(Entity,Head,Clause,File,Line).
- Goal events: Generated when calling a goal: top_goal(Goal,CompiledGoal) or goal(Goal,CompiledGoal).

---

expand_library_path/2

Expands a library alias (an atom) or a compound term (using library notation) into its absolute path. Uses a depth bound to prevent loops.

Compilation flags:
    static

Template:
    expand_library_path(LibraryAlias,AbsolutePath)
Mode and number of proofs:
    expand_library_path(+atom,?atom) - zero_or_one
    expand_library_path(+callable,?atom) - zero_or_one

---

loaded_file/1

Enumerates, by backtracking, all loaded files, returning their full paths.

Compilation flags:
    static

Template:
    loaded_file(Path)
Mode and number of proofs:
    loaded_file(?atom) - zero_or_more

---

loaded_file_property/2

Enumerates, by backtracking, loaded file properties.

Compilation flags:
    static

Template:
    loaded_file_property(Path,Property)
Mode and number of proofs:
    loaded_file_property(?atom,?compound) - zero_or_more

Remarks:

- Property basename/1: Basename of the file (includes the file extension, if any).
- Property directory/1: Directory of the file (ending with a slash).
- Property mode/1: Compilation mode of the file (possible values are optimal, normal, and debug).
- Property flags/1: Explicit flags used for compiling the file.
- Property text_properties/1: List of the file text properties (encoding/1 and bom/1). Empty if no encoding/1 directive is present and the stream used for reading the file does not have a bom/1 (or equivalent) property.
- Property target/1: Full path of the generated intermediate Prolog file.
- Property modified/1: File modification time stamp (should be regarded as an opaque but otherwise comparable term).
- Property parent/1: Full path of the parent file that loaded the file.
- Property includes/2: Full path of a file included by the file and the line of the include/1 directive.

- Property includes/1: Full path of a file included by the file.
- Property library/1: Library alias for the library that includes the file.
- Property object/3: Identifier for an object defined in the file and the start and end lines of its definition.
- Property object/1: Identifier for an object defined in the file.
- Property protocol/3: Identifier for a protocol defined in the file and the start and end lines of its definition.
- Property protocol/1: Identifier for a protocol defined in the file.
- Property category/3: Identifier for a category defined in the file and the start and end lines of its definition.
- Property category/1: Identifier for a category defined in the file.

---

loaded_files_topological_sort/1

Returns a list of full paths for all loaded user-defined files sorted by dependencies.

Compilation flags:
    static

Template:
    loaded_files_topological_sort(Sorted)
Mode and number of proofs:
    loaded_files_topological_sort(--list(atom)) - one

---

loaded_files_topological_sort/2

Sorts a list of full paths for loaded files by dependencies.

Compilation flags:
    static

Template:
    loaded_files_topological_sort(Paths,Sorted)
Mode and number of proofs:
    loaded_files_topological_sort(+list(atom),--list(atom)) - one

---

file_type_extension/2

Enumerates, by backtracking, all defined file type extensions. The defined types are: source, object, logtalk, prolog, and tmp. The source type returns both logtalk and prolog type extensions.

Compilation flags:
    static

Template:
    file_type_extension(Type,Extension)
Mode and number of proofs:
    file_type_extension(?atom,?atom) - zero_or_more

---

halt/1

Halts the Logtalk process with the given exit code. Calls all defined halt hook goals whose exit code unifies with its argument. A warning message is printed in case of goal failures or errors (which are otherwise ignored).

Compilation flags:
    static

Template:
    halt(ExitCode)
Mode and number of proofs:
    halt(+integer) - one

---

halt/0

Halts the Logtalk process with the given exit code. Calls all defined halt hook goals whose exit code unifies with the integer zero. A warning message is printed in case of goal failures or errors (which are otherwise ignored).

Compilation flags:
    static

Mode and number of proofs:

halt - one

---

halt_hook/2

Hook predicate defining a goal to be called when sending the halt/0-1 messages.

Compilation flags:
dynamic, multifile

Template:
halt_hook(ExitCode,Goal)
Mode and number of proofs:
halt_hook(?var,?callable) - zero_or_more
halt_hook(?integer,?callable) - zero_or_more

---

compile_aux_clauses/1

Compiles a list of auxiliary clauses. Can only be called during source file compilation, usually from term_expansion/2 or goal_expansion/2 hook predicate definitions.

Compilation flags:
static

Template:
compile_aux_clauses(Clauses)
Mode and number of proofs:
compile_aux_clauses(@list(clause)) - one

---

entity_prefix/2

Converts between an entity identifier and the entity prefix that is used for its compiled code. When none of the arguments is instantiated, it returns the identifier and the prefix of the entity under compilation, if any.

Compilation flags:
    static

Template:
    entity_prefix(Entity,Prefix)
Mode and number of proofs:
    entity_prefix(?entity_identifier,?atom) - zero_or_one

---

compile_predicate_heads/4

Compiles clause heads. The heads are compiled in the context of the entity under compilation when the entity argument is not instantiated.

Compilation flags:
    static

Template:
    compile_predicate_heads(Heads,Entity,CompiledHeads,ExecutionContext)
Mode and number of proofs:
    compile_predicate_heads(@list(callable),?entity_identifier,-list(callable),@execution_context) - zero_or_one
    compile_predicate_heads(@conjunction(callable),?entity_identifier,-conjunction(callable),@execution_context) - zero_or_one
    compile_predicate_heads(@callable,?entity_identifier,-callable,@execution_context) - zero_or_one

---

compile_predicate_indicators/3

Compiles predicate indicators. The predicate are compiled in the context of the entity under compilation when the entity argument is not instantiated.

Compilation flags:
    static

Template:
compile_predicate_indicators(PredicateIndicators,Entity,CompiledPredicateIndicators)
Mode and number of proofs:
compile_predicate_indicators(@list(predicate_indicator),?entity_identifier,
-list(predicate_indicator)) - zero_or_one
compile_predicate_indicators(@conjunction(predicate_indicator),?entity_identifier,
-conjunction(predicate_indicator)) - zero_or_one
compile_predicate_indicators(@predicate_indicator,?entity_identifier,-predicate_indicator) -
zero_or_one

---

decompile_predicate_heads/4

Decompiles clause heads. All compiled clause heads must belong to the same entity, which must be loaded.

Compilation flags:
static

Template:
decompile_predicate_heads(CompiledHeads,Entity,Type,Heads)
Mode and number of proofs:
decompile_predicate_heads(@list(callable),-entity_identifier,-atom,-list(callable)) - zero_or_one
decompile_predicate_heads(@conjunction(callable),-entity_identifier,-atom,-conjunction(callable)) -
zero_or_one
decompile_predicate_heads(@callable,-entity_identifier,-atom,-callable) - zero_or_one

---

decompile_predicate_indicators/4

Decompiles predicate indicators. All compiled predicate indicators must belong to the same entity, which must be loaded.

Compilation flags:
static

Template:
decompile_predicate_indicators(CompiledPredicateIndicators,Entity,Type,PredicateIndicators)
Mode and number of proofs:
decompile_predicate_indicators(@list(predicate_indicator),-entity_identifier,-atom,
-list(predicate_indicator)) - zero_or_one

decompile_predicate_indicators(@conjunction(predicate_indicator),-entity_identifier,-atom,-conjunction(predicate_indicator)) - zero_or_one
decompile_predicate_indicators(@predicate_indicator,-entity_identifier,-atom,-predicate_indicator) - zero_or_one

---

execution_context/7

Execution context term data. Execution context terms should be considered opaque terms subject to change without notice.

Compilation flags:
static

Template:
execution_context(ExecutionContext,Entity,Sender,This,Self,MetaCallContext,CoinductionStack)
Mode and number of proofs:
execution_context(?nonvar,?entity_identifier,?object_identifier,?object_identifier,?object_identifier,@list(callable),@list(callable)) - zero_or_one

---

Protected predicates

(none)

Private predicates

active_debug_handler_/1

Current active debug handler provider. There is at most one active debug handler provider at any given moment.

Compilation flags:
dynamic

Template:
active_debug_handler_(Provider)
Mode and number of proofs:
active_debug_handler_(?entity_identifier) - zero_or_one

Operators

(none)

protocol

1.39.5 monitoring

Event handlers protocol. The handlers are automatically called by the runtime for messages sent using the ::/2 control construct from objects or categories compiled with the events flag set to allow.

Availability:
built_in

Author: Paulo Moura
Version: 1:2:0
Date: 2018-11-29

Compilation flags:
static, built_in

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - before/3
  - after/3
- Protected predicates
- Private predicates
- Operators

Public predicates

before/3

Event handler for before events. A before event handler may prevent a method from being looked up or called by failing.

Compilation flags:
    static

Template:
    before(Object,Message,Sender)
Mode and number of proofs:
    before(?term,?term,?term) - zero_or_more

---

after/3

Event handler for after events. An after event handler may prevent a method from succeeding by failing.

Compilation flags:
    static

Template:
    after(Object,Message,Sender)
Mode and number of proofs:
    after(?term,?term,?term) - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

object

1.39.6 user

Pseudo-object representing the plain Prolog database. Can be used as a monitor by defining before/3 and after/3 predicates. Can be used as a hook object by defining term_expansion/2 and goal_expansion/2 multifile and dynamic predicates.

Availability:
built_in

Author: Paulo Moura
Version: 1:6:0
Date: 2024-11-11

Compilation flags:
static, built_in, context_switching_calls, dynamic_declarations, threaded

Implements:
public expanding
public forwarding
public monitoring
Uses:
user

Remarks:
(none)

Inherited public predicates:
after/3 before/3 forward/1 goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates

- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.40 coroutining

object

1.40.1 coroutining

Coroutining predicates.

Availability:
logtalk_load(coroutining(loader))

Author: Paulo Moura
Version: 0:5:0
Date: 2021-12-17

Compilation flags:
static, context_switching_calls

Dependencies:
(none)

Remarks:

- Supported backend Prolog systems: ECLiPSe, SICStus Prolog, SWI-Prolog, Trealla Prolog, XVM, and YAP.

Inherited public predicates:
(none)

- Public predicates
  - dif/2
  - dif/1
  - freeze/2
  - frozen/2
  - when/2
- Protected predicates
- Private predicates
- Operators

Public predicates

dif/2

Sets a constraint that is true iff the two terms are different.

Compilation flags:
static

Template:
dif(Term1,Term2)
Mode and number of proofs:
dif(+term,+term) - zero_or_one

dif/1

Sets a set of constraints that are true iff all terms in a list are different.

Compilation flags:
    static

Template:
    dif(Terms)
Mode and number of proofs:
    dif(+list(term)) - zero_or_one

---

freeze/2

Delays the execution of a goal until a variable is bound.

Compilation flags:
    static

Template:
    freeze(Variable,Goal)
Meta-predicate template:
    freeze(*,0)
Mode and number of proofs:
    freeze(+term,+callable) - zero_or_more

---

frozen/2

Unifies Goal with the goal delayed by Variable. When no goals are frozen on Variable, Goal is unified with true.

Compilation flags:
    static

Template:
    frozen(Variable,Goal)
Mode and number of proofs:

frozen(@var,--callable) - one

---

when/2

Calls Goal when Condition becomes true. The portable conditions are: nonvar/1, ground/1, (,)/2, and (;)/2.

Compilation flags:
static

Template:
when(Condition,Goal)
Meta-predicate template:
when(*,0)
Mode and number of proofs:
when(+callable,+callable) - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.41 crs_projections

object

1.41.1 crs_projections

Coordinate reference systems and common transformations for 2D and 3D WGS84 geodetic coordinates, Web Mercator, World Mercator, ECEF, local ENU tangent-plane, Lambert azimuthal equal-area, and UTM coordinates.

Availability:
logtalk_load(crs_projections(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-10

Compilation flags:
static, context_switching_calls

Implements:
public crs_projections_protocol
Uses:
geospatial

Remarks:

- Geographic coordinates: WGS84 coordinates are represented as geographic(Latitude,Longitude) in degrees.
- 3D geodetic coordinates: WGS84 3D coordinates are represented as geographic(Latitude,Longitude, EllipsoidalHeight) using degrees for angular axes and meters for height.
- Projected coordinates: Web Mercator, World Mercator, and Lambert azimuthal equal-area coordinates are represented as meter-based projected(X,Y) pairs. UTM coordinates are represented as grid(Easting, Northing) pairs.
- Geocentric coordinates: ECEF coordinates are represented as ecef(X,Y,Z) triples in meters.
- Local tangent-plane coordinates: ENU coordinates are represented as enu(East,North,Up) triples in meters relative to an origin coordinate.
- UTM coverage: UTM helpers and transformations only succeed for latitudes in the [-80.0,84.0[range.

Inherited public predicates:
crs_dimensions/2 crs_epsg/2 crs_kind/2 crs_name/2 crs_units/2 epsg_crs/2 lambert_azimuthal_equal_area/3 lambert_azimuthal_equal_area_inverse/3 local_tangent_plane/3 local_tangent_plane_inverse/3 transform/4 utm_crs/2 utm_zone/2 valid_coordinate/2 valid_crs/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

crs_projections_protocol, geospatial

protocol

1.41.2 crs_projections_protocol

Coordinate reference systems and coordinate transformation predicates protocol.

Availability:
logtalk_load(crs_projections(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-10

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - valid_crs/1
  - crs_name/2
  - crs_kind/2
  - crs_units/2
  - crs_dimensions/2
  - crs_epsg/2
  - epsg_crs/2
  - valid_coordinate/2
  - local_tangent_plane/3
  - local_tangent_plane_inverse/3
  - lambert_azimuthal_equal_area/3
  - lambert_azimuthal_equal_area_inverse/3
  - utm_zone/2
  - utm_crs/2
  - transform/4
- Protected predicates
- Private predicates
- Operators

Public predicates

valid_crs/1

True when the argument is a supported coordinate reference system. Supported values are wgs84, wgs84_3d, web_mercator, world_mercator, ecef, enu(OriginCoordinate), lambert_azimuthal_equal_area(OriginCoordinate), and utm(Zone,Hemisphere) terms.

Compilation flags:
    static

Template:
    valid_crs(CRS)
Mode and number of proofs:
    valid_crs(@term) - zero_or_one

---

crs_name/2

Maps a supported coordinate reference system to a human-readable name.

Compilation flags:
    static

Template:
    crs_name(CRS,Name)
Mode and number of proofs:
    crs_name(+term,-atom) - zero_or_one

---

crs_kind/2

Classifies a supported coordinate reference system as geographic, projected, geocentric, or local_tangent_plane.

Compilation flags:
    static

Template:

crs_kind(CRS,Kind)
Mode and number of proofs:
crs_kind(+term,-atom) - zero_or_one

---

crs_units/2

Returns the coordinate axis units used by a supported coordinate reference system. Supported values are degrees, meters, and [degrees,degrees,meters].

Compilation flags:
static

Template:
crs_units(CRS,Units)
Mode and number of proofs:
crs_units(+term,-term) - zero_or_one

---

crs_dimensions/2

Returns the coordinate dimensionality for a supported coordinate reference system.

Compilation flags:
static

Template:
crs_dimensions(CRS,Dimensions)
Mode and number of proofs:
crs_dimensions(+term,-positive_integer) - zero_or_one

---

crs_epsg/2

Maps a supported coordinate reference system to the corresponding EPSG code.

Compilation flags:
    static

Template:
    crs_epsg(CRS,EPSG)
Mode and number of proofs:
    crs_epsg(+term,-positive_integer) - zero_or_one

---

epsg_crs/2

Maps a supported EPSG code to the corresponding coordinate reference system term.

Compilation flags:
    static

Template:
    epsg_crs(EPSG,CRS)
Mode and number of proofs:
    epsg_crs(+positive_integer,-term) - zero_or_one

---

valid_coordinate/2

True when the coordinate is valid for the given coordinate reference system. wgs84 coordinates are represented as geographic(Latitude,Longitude) in degrees, wgs84_3d coordinates as geographic(Latitude,Longitude,EllipsoidalHeight) in degrees and meters, web_mercator, world_mercator, and lambert_azimuthal_equal_area(OriginCoordinate) coordinates as projected(X,Y) pairs in meters, ecef coordinates as ecef(X,Y,Z) triples in meters, enu(OriginCoordinate) coordinates as enu(East,North,Up) triples in meters, and utm(Zone,Hemisphere) coordinates as grid(Easting,Northing) pairs in meters.

Compilation flags:
    static

Template:

valid_coordinate(CRS,Coordinate)
Mode and number of proofs:
valid_coordinate(+term,@compound) - zero_or_one

---

local_tangent_plane/3

Projects a WGS84 geographic coordinate to local enu(East,North,Up) coordinates in meters using an origin coordinate. For height-preserving 3D geodetic workflows, use transform/4 with wgs84_3d.

Compilation flags:
static

Template:
local_tangent_plane(Origin,Coordinate,LocalCoordinate)
Mode and number of proofs:
local_tangent_plane(+compound,+compound,-compound) - zero_or_one

---

local_tangent_plane_inverse/3

Converts local enu(East,North,Up) coordinates in meters back to a WGS84 geographic coordinate using an origin coordinate. For height-preserving 3D geodetic workflows, use transform/4 with wgs84_3d.

Compilation flags:
static

Template:
local_tangent_plane_inverse(Origin,LocalCoordinate,Coordinate)
Mode and number of proofs:
local_tangent_plane_inverse(+compound,+compound,-compound) - zero_or_one

---

lambert_azimuthal_equal_area/3

Projects a WGS84 geographic coordinate to a Lambert azimuthal equal-area plane centered on an origin coordinate, returning projected(X,Y) coordinates in meters.

Compilation flags:
    static

Template:
    lambert_azimuthal_equal_area(Origin,Coordinate,ProjectedCoordinate)
Mode and number of proofs:
    lambert_azimuthal_equal_area(+compound,+compound,-compound) - zero_or_one

---

lambert_azimuthal_equal_area_inverse/3

Converts Lambert azimuthal equal-area projected(X,Y) coordinates back to a WGS84 geographic coordinate using an origin coordinate.

Compilation flags:
    static

Template:
    lambert_azimuthal_equal_area_inverse(Origin,ProjectedCoordinate,Coordinate)
Mode and number of proofs:
    lambert_azimuthal_equal_area_inverse(+compound,+compound,-compound) - zero_or_one

---

utm_zone/2

Infers the UTM longitudinal zone for a WGS84 geographic(Latitude,Longitude) coordinate. Uses the standard Norway and Svalbard special-zone rules and only succeeds for coordinates within the UTM latitude coverage.

Compilation flags:
    static

Template:
    utm_zone(Coordinate,Zone)

Mode and number of proofs:
utm_zone(+compound,-integer) - zero_or_one

---

utm_crs/2

Infers the native UTM coordinate reference system term utm(Zone,Hemisphere) for a WGS84 geographic(Latitude,Longitude) coordinate.

Compilation flags:
static

Template:
utm_crs(Coordinate,CRS)
Mode and number of proofs:
utm_crs(+compound,-compound) - zero_or_one

---

transform/4

Transforms a coordinate between supported coordinate reference systems. Transformations between projected systems are computed by converting through wgs84.

Compilation flags:
static

Template:
transform(SourceCRS,TargetCRS,Coordinate,TransformedCoordinate)
Mode and number of proofs:
transform(+term,+term,+compound,-compound) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

crs_projections, geospatial_protocol

1.42 csv

object

1.42.1 csv

CSV files reading and writing predicates using the options Header - keep, Separator - comma, and Ignore-Quotes - false.

Availability:
logtalk_load(csv(loader))

Author: Jacinto Dávila
Version: 1:0:0
Date: 2021-02-02

Compilation flags:
static, context_switching_calls

Extends:
public csv(keep,comma,false,false)

Remarks:
(none)

Inherited public predicates:

guess_arity/2 guess_separator/2 read_file/2 read_file/3 read_file_by_line/2 read_file_by_line/3 read_stream/2 read_stream/3 read_stream_by_line/2 read_stream_by_line/3 write_file/3 write_stream/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.42.2 csv(Header,Separator,IgnoreQuotes)

- Header - Header handling option with possible values missing, skip, and keep (default).
- Separator - Separator handling option with possible values comma (default for non .tsv and non .tab files or when no file name extension is available), tab (default for .tsv and .tab files), semicolon, and colon.
- IgnoreQuotes - Double-quotes handling option to ignore (true) or preserve (false; default) double quotes surrounding data.

Backward-compatible parametric object equivalent to using csv(_Header_, _Separator_, _IgnoreQuotes_, false).

Availability:
logtalk_load(csv(loader))

Author: Paulo Moura

Version: 1:0:1
Date: 2026-02-27

Compilation flags:
static, context_switching_calls

Extends:
public csv(Header,Separator,IgnoreQuotes,false)

Remarks:
(none)

Inherited public predicates:
guess_arity/2 guess_separator/2 read_file/2 read_file/3 read_file_by_line/2 read_file_by_line/3 read_stream/2 read_stream/3 read_stream_by_line/2 read_stream_by_line/3 write_file/3 write_stream/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.42.3 csv(Header,Separator,IgnoreQuotes,Comments)

- Header - Header handling option with possible values missing, skip, and keep (default).
- Separator - Separator handling option with possible values comma (default for non .tsv and non .tab files or when no file name extension is available), tab (default for .tsv and .tab files), semicolon, and colon.
- IgnoreQuotes - Double-quotes handling option to ignore (true) or preserve (false; default) double quotes surrounding data.
- Comments - Comment handling option with possible values true and false (default). When true, lines starting with # are skipped when reading CSV files and streams.

CSV file and stream reading and writing predicates.

Availability:
logtalk_load(csv(loader))

Author: Jacinto Dávila and Paulo Moura
Version: 2:2:0
Date: 2026-02-25

Compilation flags:
static, context_switching_calls

Implements:
public csv_protocol
Uses:
list
logtalk
os
reader
type

Remarks:
(none)

Inherited public predicates:

guess_arity/2 guess_separator/2 read_file/2 read_file/3 read_file_by_line/2
read_file_by_line/3 read_stream/2 read_stream/3 read_stream_by_line/2
read_stream_by_line/3 write_file/3 write_stream/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.42.4 csv_guess_questions

Support for asking questions when guessing the separator and the record arity of CSV files.

Availability:
logtalk_load(csv(loader))

Author: Jacinto Dávila
Version: 1:0:0
Date: 2021-02-03

Compilation flags:
static

Provides:
logtalk::message_tokens//2
logtalk::question_prompt_stream/4

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.42.5 csv_protocol

CSV file and stream reading and writing protocol.

Availability:
logtalk_load(csv(loader))

Author: Jacinto Dávila and Paulo Moura

Version: 2:0:1
Date: 2025-05-07

Compilation flags:
static

Dependencies:
(none)

Remarks:

- Type-checking: Some of the predicate file and stream argument type-checking exceptions depend on the Prolog backend compliance with standards.

Inherited public predicates:
(none)

- Public predicates
  - read_file/3
  - read_stream/3
  - read_file/2
  - read_stream/2
  - read_file_by_line/3
  - read_stream_by_line/3
  - read_file_by_line/2
  - read_stream_by_line/2
  - write_file/3
  - write_stream/3
  - guess_separator/2
  - guess_arity/2
- Protected predicates
- Private predicates
- Operators

Public predicates

read_file/3

Reads a CSV file saving the data as clauses for the specified object predicate. Fails if the file cannot be parsed.

Compilation flags:
    static

Template:
    read_file(File,Object,Predicate)
Mode and number of proofs:
    read_file(+atom,+object_identifier,+predicate_indicator) - zero_or_one

Exceptions:
    File is a variable:
        instantiation_error
    File is neither a variable nor an atom:
        type_error(atom,File)
    File is an atom but not an existing file:
        existence_error(file,File)
    File is an existing file but cannot be opened for reading:
        permission_error(open,source_sink,File)
    Object is a variable:
        instantiation_error
    Object is neither a variable nor an object identifier:
        type_error(object_identifier,Object)
    Object is a valid object identifier but not an existing object:
        existence_error(object,Object)
    Predicate is a variable:
        instantiation_error
    Predicate is neither a variable nor a predicate indicator:
        type_error(predicate_indicator,Predicate)
    Predicate is a valid predicate indicator but not an existing public predicate:
        existence_error(predicate,Predicate)

read_stream/3

Reads a CSV stream saving the data as clauses for the specified object predicate. Fails if the stream cannot be parsed.

Compilation flags:
    static

Template:
    read_stream(Stream,Object,Predicate)
Mode and number of proofs:
    read_stream(+stream_or_alias,+object_identifier,+predicate_indicator) - zero_or_one

Exceptions:
    Stream is a variable:
        instantiation_error
    Stream is neither a variable nor a stream-term or alias:
        domain_error(stream_or_alias,Stream)
    Stream is not an open stream:
        existence_error(stream,Stream)
    Stream is an output stream:
        permission_error(input,stream,Stream)
    Stream is a binary stream:
        permission_error(input,binary_stream,Stream)
    Object is a variable:
        instantiation_error
    Object is neither a variable nor an object identifier:
        type_error(object_identifier,Object)
    Object is a valid object identifier but not an existing object:
        existence_error(object,Object)
    Predicate is a variable:
        instantiation_error
    Predicate is neither a variable nor a predicate indicator:
        type_error(predicate_indicator,Predicate)
    Predicate is a valid predicate indicator but not an existing public predicate:
        existence_error(predicate,Predicate)

read_file/2

Reads a CSV file returning the data as a list of rows, each row a list of fields. Fails if the file cannot be parsed.

Compilation flags:
    static

Template:
    read_file(File,Rows)
Mode and number of proofs:
    read_file(+atom,-list(list)) - zero_or_one

Exceptions:
    File is a variable:
        instantiation_error
    File is neither a variable nor an atom:
        type_error(atom,File)
    File is an atom but not an existing file:
        existence_error(file,File)
    File is an existing file but cannot be opened for reading:
        permission_error(open,source_sink,File)

---

read_stream/2

Reads a CSV stream returning the data as a list of rows, each row a list of fields. Fails if the stream cannot be parsed.

Compilation flags:
    static

Template:
    read_stream(Stream,Rows)
Mode and number of proofs:
    read_stream(+stream_or_alias,-list(list)) - zero_or_one

Exceptions:
    Stream is a variable:
        instantiation_error
    Stream is neither a variable nor a stream-term or alias:

domain_error(stream_or_alias,Stream)
Stream is not an open stream:
existence_error(stream,Stream)
Stream is an output stream:
permission_error(input,stream,Stream)
Stream is a binary stream:
permission_error(input,binary_stream,Stream)

---

read_file_by_line/3

Reads a CSV file saving the data as clauses for the specified object predicate. The file is read line by line. Fails if the file cannot be parsed.

Compilation flags:
static

Template:
read_file_by_line(File,Object,Predicate)
Mode and number of proofs:
read_file_by_line(+atom,+object_identifier,+predicate_indicator) - zero_or_one

Exceptions:
File is a variable:
instantiation_error
File is neither a variable nor an atom:
type_error(atom,File)
File is an atom but not an existing file:
existence_error(file,File)
File is an existing file but cannot be opened for reading:
permission_error(open,source_sink,File)
Object is a variable:
instantiation_error
Object is neither a variable nor an object identifier:
type_error(object_identifier,Object)
Object is a valid object identifier but not an existing object:
existence_error(object,Object)
Predicate is a variable:
instantiation_error
Predicate is neither a variable nor a predicate indicator:
type_error(predicate_indicator,Predicate)
Predicate is a valid predicate indicator but not an existing public predicate:
existence_error(predicate,Predicate)

read_stream_by_line/3

Reads a CSV stream saving the data as clauses for the specified object predicate. The stream is read line by line. Fails if the stream cannot be parsed.

Compilation flags:
    static

Template:
    read_stream_by_line(Stream,Object,Predicate)
Mode and number of proofs:
    read_stream_by_line(+stream_or_alias,+object_identifier,+predicate_indicator) - zero_or_one

Exceptions:
    Stream is a variable:
        instantiation_error
    Stream is neither a variable nor a stream-term or alias:
        domain_error(stream_or_alias,Stream)
    Stream is not an open stream:
        existence_error(stream,Stream)
    Stream is an output stream:
        permission_error(input,stream,Stream)
    Stream is a binary stream:
        permission_error(input,binary_stream,Stream)
    Object is a variable:
        instantiation_error
    Object is neither a variable nor an object identifier:
        type_error(object_identifier,Object)
    Object is a valid object identifier but not an existing object:
        existence_error(object,Object)
    Predicate is a variable:
        instantiation_error
    Predicate is neither a variable nor a predicate indicator:
        type_error(predicate_indicator,Predicate)
    Predicate is a valid predicate indicator but not an existing public predicate:
        existence_error(predicate,Predicate)

read_file_by_line/2

Reads a CSV file returning the data as a list of rows, each row a list of fields. The file is read line by line. Fails if the file cannot be parsed.

Compilation flags:
    static

Template:
    read_file_by_line(File,Rows)
Mode and number of proofs:
    read_file_by_line(+atom,-list(list)) - zero_or_one

Exceptions:
    File is a variable:
        instantiation_error
    File is neither a variable nor an atom:
        type_error(atom,File)
    File is an atom but not an existing file:
        existence_error(file,File)
    File is an existing file but cannot be opened for reading:
        permission_error(open,source_sink,File)

---

read_stream_by_line/2

Reads a CSV stream returning the data as a list of rows, each row a list of fields. The stream is read line by line. Fails if the stream cannot be parsed.

Compilation flags:
    static

Template:
    read_stream_by_line(Stream,Rows)
Mode and number of proofs:
    read_stream_by_line(+stream_or_alias,-list(list)) - zero_or_one

Exceptions:
    Stream is a variable:
        instantiation_error
    Stream is neither a variable nor a stream-term or alias:

domain_error(stream_or_alias,Stream)
Stream is not an open stream:
existence_error(stream,Stream)
Stream is an output stream:
permission_error(input,stream,Stream)
Stream is a binary stream:
permission_error(input,binary_stream,Stream)

---

write_file/3

Writes a CSV file with the data represented by the solutions to the specified object predicate.

Compilation flags:
static

Template:
write_file(File,Object,Predicate)
Mode and number of proofs:
write_file(+atom,+object_identifier,+predicate_indicator) - one

Exceptions:
File is a variable:
instantiation_error
File is neither a variable nor an atom:
type_error(atom,File)
File is an atom but cannot be opened for writing:
permission_error(open,source_sink,File)
Object is a variable:
instantiation_error
Object is neither a variable nor an object identifier:
type_error(object_identifier,Object)
Object is a valid object identifier but not an existing object:
existence_error(object,Object)
Predicate is a variable:
instantiation_error
Predicate is neither a variable nor a predicate indicator:
type_error(predicate_indicator,Predicate)
Predicate is a valid predicate indicator but not an existing public predicate:
existence_error(predicate,Predicate)

---

write_stream/3

Writes a CSV stream with the data represented by the solutions to the specified object predicate.

Compilation flags:
    static

Template:
    write_stream(Stream,Object,Predicate)
Mode and number of proofs:
    write_stream(+stream_or_alias,+object_identifier,+predicate_indicator) - one

Exceptions:
    Stream is a variable:
        instantiation_error
    Stream is neither a variable nor a stream-term or alias:
        domain_error(stream_or_alias,Stream)
    Stream is not an open stream:
        existence_error(stream,Stream)
    Stream is an input stream:
        permission_error(output,stream,Stream)
    Stream is a binary stream:
        permission_error(output,binary_stream,Stream)
    Object is a variable:
        instantiation_error
    Object is neither a variable nor an object identifier:
        type_error(object_identifier,Object)
    Object is a valid object identifier but not an existing object:
        existence_error(object,Object)
    Predicate is a variable:
        instantiation_error
    Predicate is neither a variable nor a predicate indicator:
        type_error(predicate_indicator,Predicate)
    Predicate is a valid predicate indicator but not an existing public predicate:
        existence_error(predicate,Predicate)

guess_separator/2

Guesses the separator used in a given file, asking the user to confirm.

Compilation flags:
    static

Template:
    guess_separator(File,Separator)
Mode and number of proofs:
    guess_separator(+atom,-atom) - one

Exceptions:
    File is a variable:
        instantiation_error
    File is neither a variable nor an atom:
        type_error(atom,File)
    File is an atom but not an existing file:
        existence_error(file,File)
    File is an atom but cannot be opened for writing:
        permission_error(open,source_sink,File)

---

guess_arity/2

Guesses the arity of records in a given file, asking the user to confirm.

Compilation flags:
    static

Template:
    guess_arity(File,Arity)
Mode and number of proofs:
    guess_arity(+atom,-number) - one

Exceptions:
    File is a variable:
        instantiation_error
    File is neither a variable nor an atom:
        type_error(atom,File)
    File is an atom but not an existing file:

existence_error(file,File)
File is an atom but cannot be opened for writing:
permission_error(open,source_sink,File)

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.43 cuid2

object

1.43.1 cuid2

Cuid2 generator using atom representation, 24 symbols, and a lowercase alphanumeric alphabet.

Availability:
logtalk_load(cuid2(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-26

Compilation flags:
static, context_switching_calls

Extends:
public cuid2(atom,24,abcdefghijklmnopqrstuvwxyz0123456789)

Remarks:
(none)

Inherited public predicates:
generate/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

cuid2(Representation,Size,Alphabet), ids, ksuid, nanoid, snowflakeid, ulid, uuid

object

1.43.2 cuid2(Representation,Size,Alphabet)

- Representation - Text representation for the Cuid2 identifier. Possible values are atom, chars, and codes.
- Size - Number of symbols in the Cuid2 identifier.
- Alphabet - Alphabet used for generating Cuid2 identifiers represented as an atom, list of characters, or list of character codes.

Cuid2 generator.

Availability:
logtalk_load(cuid2(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-26

Compilation flags:
static, context_switching_calls

Implements:
public cuid2_protocol
Uses:
fast_random(Algorithm)
list
os

Remarks:
(none)

Inherited public predicates:
generate/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

cuid2, ids(Representation,Bytes), ksuid(Representation,Alphabet), nanoid(Representation,Size,Alphabet), snowflakeid(Representation,EpochMilliseconds,TimeUnitMilliseconds,TimestampBits,NodeBits,SequenceBits,Node), ulid(Representation), uuid(Representation)

protocol

1.43.3 cuid2_protocol

Cuid2 generator protocol.

Availability:
logtalk_load(cuid2(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-26

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - generate/1
- Protected predicates
- Private predicates
- Operators

Public predicates

generate/1

Returns a Cuid2 identifier.

Compilation flags:
    static

Template:
    generate(Cuid2)
Mode and number of proofs:
    generate(--ground) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.44 cusum_anomaly_detector

object

1.44.1 cusum_anomaly_detector

CUSUM (Cumulative Sum Control Chart) anomaly detector for continuous sequence-like datasets. Learns per-step population mean and standard deviation from baseline training examples, or from the baseline subset of a labeled dataset selected by learn-time class-label options, using a dataset object implementing the anomaly_dataset_protocol protocol. Returns a detector term that can be used for scoring, prediction, and export.

Availability:
logtalk_load(cusum_anomaly_detector(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public anomaly_detector_common
Uses:
format
list
pairs
population
type

Remarks:
(none)

Inherited public predicates:
anomaly_detector_options/2 check_anomaly_detector/1 check_option/1 check_options/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 predict/4 print_anomaly_detector/1 score/3 score_all/3 valid_anomaly_detector/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

anomaly_dataset_protocol, anomaly_detector_protocol, isolation_forest_anomaly_detector, knn_distance_anomaly_detector, lof_anomaly_detector, modified_z_score_anomaly_detector, z_score_anomaly_detector

1.45 datalog

object

1.45.1 datalog

Portable Datalog engine with stratified negation and incremental updates.

Availability:
logtalk_load(datalog(loader))

Author: Paulo Moura
Version: 0:1:1
Date: 2026-03-30

Compilation flags:
static, context_switching_calls

Implements:
public datalog_protocol
Uses:
list
numberlist
type
varlist

Remarks:
(none)

Inherited public predicates:
add_rule/3 assert_fact/1 begin/0 clear/0 commit/0 explain/2 facts/1 load_program/1 materialize/0 predicate_stratum/3 query/1 query/2 remove_rule/1 retract_fact/1 rollback/0 rules/1 strata/1 update/3

- Public predicates
- Protected predicates
- Private predicates
  - rule_/3
  - edb_fact_/1
  - idb_fact_/1
  - support_count_/2
  - support_edge_/3
  - predicate_stratum_/3
  - snapshot_/6
  - restore_snapshot/6
  - restore_edb_facts/1
  - restore_idb_facts/1
  - restore_support_counts/1
  - restore_support_edges/1
  - restore_predicate_strata/1
  - strata_from_numbers/2
  - predicates_in_stratum/2

    - has_aggregate_rules/0
    - aggregate_body_predicate/2
    - derive_aggregate_literal/2
    - derive_aggregate_goals/1
    - optimize_rule_body/2
    - partition_body_literals/5
    - literal_bucket/2
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

rule_/3

Table of loaded rules represented as rule id, head, and body literals.

Compilation flags:
    dynamic

Template:
    rule_(Id,Head,Body)
Mode and number of proofs:
    rule_(?nonvar,?callable,?list) - zero_or_more

edb_fact_/1

Table of extensional database (EDB) facts.

Compilation flags:
dynamic

Template:
edb_fact_(Fact)
Mode and number of proofs:
edb_fact_(?callable) - zero_or_more

---

idb_fact_/1

Table of intensional database (IDB) currently derived facts.

Compilation flags:
dynamic

Template:
idb_fact_(Fact)
Mode and number of proofs:
idb_fact_(?callable) - zero_or_more

---

support_count_/2

Table of derivation support counts for currently derived facts.

Compilation flags:
dynamic

Template:
support_count_(Fact,Count)
Mode and number of proofs:
support_count_(?callable,?integer) - zero_or_more

---

support_edge_/3

Table of concrete derivation edges as fact, rule id, and supporting literals.

Compilation flags:
    dynamic

Template:
    support_edge_(Fact,RuleId,Supports)
Mode and number of proofs:
    support_edge_(?callable,?nonvar,?list) - zero_or_more

---

predicate_stratum_/3

Table of computed predicate strata represented as name, arity, and stratum number.

Compilation flags:
    dynamic

Template:
    predicate_stratum_(Name,Arity,Stratum)
Mode and number of proofs:
    predicate_stratum_(?atom,?integer,?integer) - zero_or_more

---

snapshot_/6

Transaction snapshot of rules, EDB facts, IDB facts, support counts, support edges, and predicate strata.

Compilation flags:
    dynamic

Template:
    snapshot_(Rules,EdbFacts,IdbFacts,SupportCounts,SupportEdges,PredicateStrata)
Mode and number of proofs:
    snapshot_(?list,?list,?list,?list,?list,?list) - zero_or_one

---

restore_snapshot/6

Restores a saved transaction snapshot into the current engine state.

Compilation flags:
    static

Template:
    restore_snapshot(Rules,EdbFacts,IdbFacts,SupportCounts,SupportEdges,PredicateStrata)
Mode and number of proofs:
    restore_snapshot(+list,+list,+list,+list,+list,+list) - one

---

restore_edb_facts/1

Restores EDB facts from a saved list.

Compilation flags:
    static

Template:
    restore_edb_facts(Facts)
Mode and number of proofs:
    restore_edb_facts(+list(callable)) - one

---

restore_idb_facts/1

Restores IDB facts from a saved list.

Compilation flags:
    static

Template:
    restore_idb_facts(Facts)
Mode and number of proofs:
    restore_idb_facts(+list(callable)) - one

---

restore_support_counts/1

Restores support counts from a saved list.

Compilation flags:
    static

Template:
    restore_support_counts(Supports)
Mode and number of proofs:
    restore_support_counts(+list) - one

---

restore_support_edges/1

Restores support edges from a saved list.

Compilation flags:
    static

Template:
    restore_support_edges(Edges)
Mode and number of proofs:
    restore_support_edges(+list) - one

---

restore_predicate_strata/1

Restores predicate strata from a saved list.

Compilation flags:
    static

Template:
    restore_predicate_strata(Strata)
Mode and number of proofs:
    restore_predicate_strata(+list) - one

---

strata_from_numbers/2

Builds grouped strata terms from a sorted list of stratum numbers.

Compilation flags:
    static

Template:
    strata_from_numbers(StratumNumbers,Strata)
Mode and number of proofs:
    strata_from_numbers(+list(integer),-list) - one

---

predicates_in_stratum/2

Returns predicates in a given stratum as sorted predicate(Name, Arity) terms.

Compilation flags:
    static

Template:
    predicates_in_stratum(Stratum,Predicates)
Mode and number of proofs:
    predicates_in_stratum(+integer,-list) - one

---

has_aggregate_rules/0

True when at least one loaded rule body contains an aggregate literal.

Compilation flags:
    static

Mode and number of proofs:
    has_aggregate_rules - zero_or_one

---

aggregate_body_predicate/2

Enumerates predicate indicators referenced in aggregate goals from a body literal list.

Compilation flags:
    static

Template:
    aggregate_body_predicate(Body,Predicate)
Mode and number of proofs:
    aggregate_body_predicate(+list,-compound) - zero_or_more

---

derive_aggregate_literal/2

Evaluates an aggregate literal and returns a normalized support term.

Compilation flags:
    static

Template:
    derive_aggregate_literal(AggregateLiteral,Support)
Mode and number of proofs:
    derive_aggregate_literal(+compound,-nonvar) - zero_or_one

---

derive_aggregate_goals/1

Succeeds when all aggregate goals are true for the current bindings.

Compilation flags:
    static

Template:
    derive_aggregate_goals(Goals)
Mode and number of proofs:
    derive_aggregate_goals(+list(callable)) - zero_or_one

---

optimize_rule_body/2

Normalizes a rule body by placing positive ground literals first, then positive non-ground literals, then aggregates, and finally negative literals.

Compilation flags:
    static

Template:
    optimize_rule_body(Body,OptimizedBody)
Mode and number of proofs:
    optimize_rule_body(+list,-list) - one

---

partition_body_literals/5

Partitions body literals into positive-ground, positive-non-ground, aggregate, and negative lists preserving relative order.

Compilation flags:
    static

Template:
    partition_body_literals(Body,PositiveGround,PositiveNonGround,Aggregates,Negatives)
Mode and number of proofs:
    partition_body_literals(+list,-list,-list,-list,-list) - one

---

literal_bucket/2

Classifies a body literal into one of the normalization buckets.

Compilation flags:
    static

Template:
    literal_bucket(Literal,Bucket)
Mode and number of proofs:
    literal_bucket(+nonvar,-atom) - one

Operators

(none)

protocol

1.45.2 datalog_protocol

Datalog and incremental rule engine protocol (stratified negation subset).

Availability:
logtalk_load(datalog(loader))

Author: Paulo Moura
Version: 0:1:0
Date: 2026-02-13

Compilation flags:
static

Dependencies:
(none)

Remarks:

- Rules: Rules are represented as rule(Id, Head, Body) where Body is a list of literals using Term for positive, neg(Term) for negative, and agg(Op, Template, Goals, Result) for aggregates where Op is one of count, sum, min, or max.
- Facts: EDB facts are represented by callable and ground terms.
- Limitations: Current version requires aggregate dependencies to be in lower strata.

Inherited public predicates:
(none)

- Public predicates
  - clear/0
  - load_program/1

  - add_rule/3
  - remove_rule/1
  - begin/0
  - commit/0
  - rollback/0
  - assert_fact/1
  - retract_fact/1
  - materialize/0
  - update/3
  - query/1
  - query/2
  - explain/2
  - rules/1
  - facts/1
  - predicate_stratum/3
  - strata/1
- Protected predicates
- Private predicates
- Operators

Public predicates

clear/0

Clears all loaded rules, base facts, derived facts, and explanation supports.

Compilation flags:
    static

Mode and number of proofs:
    clear - one

load_program/1

Loads a full program represented as a list of rule(Id,Head,Body) and fact(Fact) terms, replacing current engine state.

Compilation flags:
    static

Template:
    load_program(Program)
Mode and number of proofs:
    load_program(+list) - one

---

add_rule/3

Adds or replaces a rule. Rule safety is checked.

Compilation flags:
    static

Template:
    add_rule(Id,Head,Body)
Mode and number of proofs:
    add_rule(+nonvar,+callable,+list(callable)) - one

---

remove_rule/1

Removes all rules matching a rule identifier.

Compilation flags:
    static

Template:
    remove_rule(Id)
Mode and number of proofs:
    remove_rule(+nonvar) - one

---

begin/0

Starts a transaction by saving the current engine state snapshot.

Compilation flags:
    static

Mode and number of proofs:
    begin - one

---

commit/0

Commits a transaction by discarding the saved state snapshot.

Compilation flags:
    static

Mode and number of proofs:
    commit - one

---

rollback/0

Rolls back a transaction by restoring the saved state snapshot.

Compilation flags:
    static

Mode and number of proofs:
    rollback - one

---

assert_fact/1

Asserts a ground EDB fact if not already present.

Compilation flags:
    static

Template:
    assert_fact(Fact)
Mode and number of proofs:
    assert_fact(+callable) - one

---

retract_fact/1

Retracts an EDB fact if present.

Compilation flags:
    static

Template:
    retract_fact(Fact)
Mode and number of proofs:
    retract_fact(+callable) - one

---

materialize/0

Computes rule closure from current EDB facts and loaded rules using a fixpoint algorithm.

Compilation flags:
    static

Mode and number of proofs:
    materialize - one

---

update/3

Applies incremental EDB updates and propagates derivation additions/removals; returns the resulting truth delta.

Compilation flags:
    static

Template:
    update(Inserts,Deletes,Delta)
Mode and number of proofs:
    update(+list(callable),+list(callable),-compound) - one

---

query/1

Enumerates currently true facts (EDB + IDB).

Compilation flags:
    static

Template:
    query(Goal)
Mode and number of proofs:
    query(?callable) - zero_or_more

---

query/2

Same as query/1 while returning the unified goal as the second argument.

Compilation flags:
    static

Template:
    query(Goal,Bindings)
Mode and number of proofs:
    query(?callable,?callable) - zero_or_more

---

explain/2

Returns one explanation for a currently true fact.

Compilation flags:
    static

Template:
    explain(Fact,Explanation)
Mode and number of proofs:
    explain(+callable,-nonvar) - zero_or_more

---

rules/1

Returns the loaded rules.

Compilation flags:
    static

Template:
    rules(Rules)
Mode and number of proofs:
    rules(-list) - one

---

facts/1

Returns all currently true facts as a sorted list.

Compilation flags:
    static

Template:
    facts(Facts)
Mode and number of proofs:

facts(-list(callable)) - one

---

predicate_stratum/3

Enumerates predicate strata as functor, arity, and stratum number.

Compilation flags:
static

Template:
predicate_stratum(Functor,Arity,Stratum)
Mode and number of proofs:
predicate_stratum(?atom,?integer,?integer) - zero_or_more

---

strata/1

Returns all strata grouped by stratum number and contained predicates.

Compilation flags:
static

Template:
strata(Strata)
Mode and number of proofs:
strata(-list) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.46 dates

object

1.46.1 date

Date predicates.

Availability:
logtalk_load(dates(loader))

Author: Paulo Moura
Version: 2:3:0
Date: 2026-04-08

Compilation flags:
static, context_switching_calls

Implements:
public datep
Uses:
list
os

Remarks:
(none)

Inherited public predicates:
add_duration/3 after/2 before/2 compare_date_time/3 date_time_to_unix/2 day_of_year/2 day_of_year_date/3 days_in_month/3 duration_between/3 format_date_time/4 leap_year/1 local_to_utc/3 month_weekday_date/5 name_of_day/3 name_of_month/3 normalize_date_time/2 same_instant/2 subtract_duration/3 today/3 unix_to_date_time/2 utc_to_local/3 valid/3 valid_date_time/1 week_of_year_iso/2 weekday/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.46.2 datep

Date protocol.

Availability:
logtalk_load(dates(loader))

Author: Paulo Moura
Version: 2:3:0
Date: 2026-04-08

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - today/3
  - leap_year/1
  - name_of_day/3
  - name_of_month/3
  - days_in_month/3
  - valid/3
  - date_time_to_unix/2
  - unix_to_date_time/2
  - add_duration/3
  - subtract_duration/3
  - duration_between/3
  - utc_to_local/3
  - local_to_utc/3
  - format_date_time/4
  - day_of_year/2
  - day_of_year_date/3
  - month_weekday_date/5
  - week_of_year_iso/2
  - weekday/2
  - normalize_date_time/2
  - valid_date_time/1
  - before/2
  - after/2
  - same_instant/2
  - compare_date_time/3
- Protected predicates
- Private predicates
- Operators

Public predicates

today/3

Returns current date.

Compilation flags:
    static

Template:
    today(Year,Month,Day)
Mode and number of proofs:
    today(-integer,-integer,-integer) - one

---

leap_year/1

True if the argument is a leap year.

Compilation flags:
    static

Template:
    leap_year(Year)
Mode and number of proofs:
    leap_year(+integer) - zero_or_one

---

name_of_day/3

Name and short name of day using ISO weekday numbering (Monday=1, ..., Sunday=7).

Compilation flags:
    static

Template:
    name_of_day(Index,Name,Short)
Mode and number of proofs:
    name_of_day(?integer,?atom,?atom) - zero_or_more

---

name_of_month/3

Name and short name of month.

Compilation flags:
    static

Template:
    name_of_month(Index,Name,Short)
Mode and number of proofs:
    name_of_month(?integer,?atom,?atom) - zero_or_more

---

days_in_month/3

Number of days in a month.

Compilation flags:
    static

Template:
    days_in_month(Month,Year,Days)
Mode and number of proofs:
    days_in_month(?integer,+integer,?integer) - zero_or_more

---

valid/3

True if the arguments represent a valid date.

Compilation flags:
    static

Template:
    valid(Year,Month,Day)
Mode and number of proofs:

valid(@integer,@integer,@integer) - zero_or_one

---

date_time_to_unix/2

Converts a UTC date-time term date_time(Year,Month,Day,Hours,Minutes,Seconds) to Unix epoch seconds.

Compilation flags:
static

Template:
date_time_to_unix(DateTime,UnixTime)
Mode and number of proofs:
date_time_to_unix(+compound,-integer) - zero_or_one

---

unix_to_date_time/2

Converts Unix epoch seconds to a UTC date-time term date_time(Year,Month,Day,Hours,Minutes,Seconds).

Compilation flags:
static

Template:
unix_to_date_time(UnixTime,DateTime)
Mode and number of proofs:
unix_to_date_time(+integer,-compound) - one

---

add_duration/3

Adds a duration to a datetime. Duration can be integer seconds, duration(Days,Hours,Minutes,Seconds), or a calendar-aware duration(Years,Months,Days,Hours,Minutes,Seconds). For the 6-arity form, the year and month delta is applied first using calendar arithmetic, clamping the day to the last valid day of the resulting month when necessary (e.g. January 31 plus one month gives February 28 or 29), and the remaining day and time delta is then applied via fixed-length arithmetic.

Compilation flags:

static

Template:
add_duration(DateTime,Duration,ResultDateTime)
Mode and number of proofs:
add_duration(+compound,+nonvar,-compound) - zero_or_one

---

subtract_duration/3

Subtracts a duration from a datetime. Duration can be integer seconds, duration(Days,Hours,Minutes,Seconds), or a calendar-aware duration(Years,Months,Days,Hours,Minutes,Seconds). For the 6-arity form, the year and month delta is subtracted first using calendar arithmetic with end-of-month day clamping, and the remaining day and time delta is then subtracted via fixed-length arithmetic.

Compilation flags:
static

Template:
subtract_duration(DateTime,Duration,ResultDateTime)
Mode and number of proofs:
subtract_duration(+compound,+nonvar,-compound) - zero_or_one

---

duration_between/3

Computes the difference between two datetimes as integer seconds, as duration(Days,Hours,Minutes,Seconds), or as a calendar-aware duration(Years,Months,Days,Hours,Minutes,Seconds). For the 6-arity form, the year and month components count the largest whole number of calendar months between the two datetimes (consistent with the day-clamping semantics of add_duration/3), and the remaining days and time fields are the exact residual. For backward intervals all fields are negative.

Compilation flags:
static

Template:
duration_between(StartDateTime,EndDateTime,Duration)
Mode and number of proofs:
duration_between(+compound,+compound,?term) - zero_or_one

---

utc_to_local/3

Converts a UTC datetime to a local datetime using an explicit timezone offset atom (Z or ±HH:MM).

Compilation flags:
    static

Template:
    utc_to_local(UTCDateTime,Offset,LocalDateTime)
Mode and number of proofs:
    utc_to_local(+compound,+atom,-compound) - zero_or_one

---

local_to_utc/3

Converts a local datetime to UTC using an explicit timezone offset atom (Z or ±HH:MM).

Compilation flags:
    static

Template:
    local_to_utc(LocalDateTime,Offset,UTCDateTime)
Mode and number of proofs:
    local_to_utc(+compound,+atom,-compound) - zero_or_one

---

format_date_time/4

Formats a date-time using an explicit UTC offset in seconds. Supported format identifiers are rfc3339, iso8601, atom, rfc2822, rfc5322, rss, http_date, rfc1123, unix_date, common_log, date_short, date_medium, date_long, date_full, time_short, time_medium, time_long, time_full, date_time_short, date_time_medium, date_time_long, and date_time_full. RFC 3339, ISO 8601, Atom, and the date_* and date_time_* styles require a four-digit non-negative year. Formats that include numeric offsets require an offset expressible in whole minutes. HTTP-date and RFC 1123 output are always normalized to GMT. The style presets are English-only presentation formats. Fails if the format or date-time are not valid.

Compilation flags:
    static

Template:

format_date_time(DateTime,OffsetSeconds,Format,String)
Mode and number of proofs:
format_date_time(+compound,+integer,+atom,-atom) - zero_or_one

---

day_of_year/2

Computes the day of year (1-366) for a date(Year,Month,Day) or date_time(...) term.

Compilation flags:
static

Template:
day_of_year(DateLike,DayOfYear)
Mode and number of proofs:
day_of_year(+compound,?integer) - zero_or_one

---

day_of_year_date/3

Computes the calendar date corresponding to a year and day of year (1-366) as date(Year,Month,Day).

Compilation flags:
static

Template:
day_of_year_date(Year,DayOfYear,Date)
Mode and number of proofs:
day_of_year_date(+integer,+integer,-compound) - zero_or_one

---

month_weekday_date/5

Computes the calendar date for the nth or last weekday in a month as date(Year,Month,Day). Week values 1-4 select the nth weekday and week value 5 selects the last one. Weekday uses ISO numbering (Monday=1, …, Sunday=7).

Compilation flags:
    static

Template:
    month_weekday_date(Year,Month,Week,Weekday,Date)
Mode and number of proofs:
    month_weekday_date(+integer,+integer,+integer,+integer,-compound) - zero_or_one

---

week_of_year_iso/2

Computes ISO week for a date(Year,Month,Day) or date_time(...) term as week(Week,Year).

Compilation flags:
    static

Template:
    week_of_year_iso(DateLike,ISOWeek)
Mode and number of proofs:
    week_of_year_iso(+compound,?compound) - zero_or_one

---

weekday/2

Computes ISO weekday number (Monday=1, …, Sunday=7) for a date(Year,Month,Day) or date_time(...) term.

Compilation flags:
    static

Template:
    weekday(DateLike,Weekday)
Mode and number of proofs:

weekday(+compound,?integer) - zero_or_one

---

normalize_date_time/2

Normalizes a datetime term by carrying overflows/underflows in date and time fields.

Compilation flags:
static

Template:
normalize_date_time(DateTime,NormalizedDateTime)
Mode and number of proofs:
normalize_date_time(+compound,-compound) - one

---

valid_date_time/1

True iff a datetime term is valid in strict mode.

Compilation flags:
static

Template:
valid_date_time(DateTime)
Mode and number of proofs:
valid_date_time(@compound) - zero_or_one

---

before/2

True iff DateTime1 represents an instant strictly before DateTime2.

Compilation flags:
static

Template:

before(DateTime1,DateTime2)
Mode and number of proofs:
before(+compound,+compound) - zero_or_one

---

after/2

True iff DateTime1 represents an instant strictly after DateTime2.

Compilation flags:
static

Template:
after(DateTime1,DateTime2)
Mode and number of proofs:
after(+compound,+compound) - zero_or_one

---

same_instant/2

True iff DateTime1 and DateTime2 represent the same instant (equal Unix epoch seconds).

Compilation flags:
static

Template:
same_instant(DateTime1,DateTime2)
Mode and number of proofs:
same_instant(+compound,+compound) - zero_or_one

---

compare_date_time/3

Three-way comparison of two datetime terms. Order is unified with <, =, or >. Suitable for use with sort/3.

Compilation flags:
    static

Template:
    compare_date_time(Order,DateTime1,DateTime2)
Mode and number of proofs:
    compare_date_time(?atom,+compound,+compound) - zero_or_one

---

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

date, timep

object

1.46.3 time

Time predicates.

Availability:
    logtalk_load(dates(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2014-09-27

Compilation flags:
static, context_switching_calls

Implements:
public timep
Uses:
os

Remarks:
(none)

Inherited public predicates:
cpu_time/1 now/3 valid/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

datep

protocol

1.46.4 timep

Time protocol.

Availability:
logtalk_load(dates(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2000-07-24

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - now/3
  - cpu_time/1
  - valid/3
- Protected predicates
- Private predicates
- Operators

Public predicates

now/3

Returns current time.

Compilation flags:
    static

Template:
    now(Hours,Mins,Secs)
Mode and number of proofs:
    now(-integer,-integer,-integer) - one

---

cpu_time/1

Returns the current cpu time.

Compilation flags:
    static

Template:
    cpu_time(Time)
Mode and number of proofs:
    cpu_time(-number) - one

---

valid/3

True if the arguments represent a valid time value.

Compilation flags:
    static

Template:
    valid(Hours,Mins,Secs)
Mode and number of proofs:
    valid(+integer,+integer,+integer) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

time, datep

1.47 dates_tz

object

1.47.1 dates_tz

Zone-aware date-time conversion bridging the dates and tzif libraries.

Availability:
logtalk_load(dates_tz(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-08

Compilation flags:
static, context_switching_calls

Implements:
public dates_tz_protocol
Uses:
date
tzif

Remarks:
(none)

Inherited public predicates:
convert_zones/4 convert_zones_with_resolution/5 local_to_utc_tz/3
local_to_utc_tz_with_resolution/4 utc_to_local_tz/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

dates_tz_protocol, date, tzif

protocol

1.47.2 dates_tz_protocol

Protocol for zone-aware date-time conversion using cached TZif data.

Availability:
logtalk_load(dates_tz(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-08

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - utc_to_local_tz/3
  - local_to_utc_tz/3
  - local_to_utc_tz_with_resolution/4
  - convert_zones/4
  - convert_zones_with_resolution/5
- Protected predicates
- Private predicates
- Operators

Public predicates

utc_to_local_tz/3

Converts a UTC date-time to the civil local date-time in the named zone. Requires the zone to be present in the cached TZif data.

Compilation flags:
    static

Template:
    utc_to_local_tz(UTCDateTime,Zone,LocalDateTime)
Mode and number of proofs:
    utc_to_local_tz(+compound,+atom,-compound) - one_or_error

---

local_to_utc_tz/3

Converts a civil local date-time in the named zone to UTC. This strict variant fails silently if the local time falls in a DST gap (non-existent time) or a DST fold (ambiguous time). Requires the zone to be present in the cached TZif data.

Compilation flags:
    static

Template:
    local_to_utc_tz(LocalDateTime,Zone,UTCDateTime)
Mode and number of proofs:
    local_to_utc_tz(+compound,+atom,-compound) - zero_or_one_or_error

---

local_to_utc_tz_with_resolution/4

Converts a civil local date-time in the named zone to UTC using an explicit resolution mode for ambiguous or non-existent times. The resolution mode can be strict (fail unless exactly one interpretation), first (prefer the earliest valid interpretation), second (prefer the latest valid interpretation), or all (enumerate all valid interpretations). Requires the zone to be present in the cached TZif data.

Compilation flags:
    static

Template:
local_to_utc_tz_with_resolution(LocalDateTime,Zone,ResolutionMode,UTCDateTime)
Mode and number of proofs:
local_to_utc_tz_with_resolution(+compound,+atom,+atom,-compound) - zero_or_more

---

convert_zones/4

Converts a civil local date-time in one zone to the civil local date-time in another zone. Uses strict interpretation: fails if the source local time is in a DST gap or fold. Requires both zones to be present in the cached TZif data.

Compilation flags:
static

Template:
convert_zones(LocalDateTime,FromZone,ToZone,ResultDateTime)
Mode and number of proofs:
convert_zones(+compound,+atom,+atom,-compound) - zero_or_one_or_error

---

convert_zones_with_resolution/5

Converts a civil local date-time in one zone to the civil local date-time in another zone using an explicit resolution mode for the source zone. The resolution mode is applied when the source local time is ambiguous or non-existent. Requires both zones to be present in the cached TZif data.

Compilation flags:
static

Template:
convert_zones_with_resolution(LocalDateTime,FromZone,ResolutionMode,ToZone,
ResultDateTime)
Mode and number of proofs:
convert_zones_with_resolution(+compound,+atom,+atom,+atom,-compound) - zero_or_more

---

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

dates_tz

1.48 dbscan_clusterer

object

1.48.1 dbscan_clusterer

DBSCAN clusterer for continuous datasets. Learns from a dataset object implementing the clustering_dataset_protocol protocol and returns a clusterer term that can be used for assigning new instances to clusters and exported as predicate clauses.

Availability:
logtalk_load(dbscan_clusterer(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public clusterer_common
public search_indexing
Uses:
avltree
deque
format

list
numberlist
pairs
type

Remarks:
(none)

Inherited public predicates:
check_clusterer/1 check_option/1 check_options/1 cluster/3 clusterer_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 print_clusterer/1 valid_clusterer/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

clusterer_protocol, clustering_dataset_protocol, kcenters_clusterer, kmeans_clusterer, kmedoids_clusterer, kmedians_clusterer

1.49 dead_code_scanner

object

1.49.1 dead_code_scanner

A tool for detecting likely dead code in compiled Logtalk entities and Prolog modules compiled as objects.

Availability:
logtalk_load(dead_code_scanner(loader))

Author: Barry Evans and Paulo Moura
Version: 0:18:2
Date: 2026-05-05

Compilation flags:
static, context_switching_calls

Imports:
public tool_diagnostics_common
public options
Uses:
list
logtalk
os
term_io
type
user

Remarks:

- Dead code: A predicate or non-terminal that is not called (directly or indirectly) by any scoped predicate or non-terminal. These predicates and non-terminals are not used, cannot be called without breaking encapsulation, and are thus considered dead code.
- Diagnostics targets: The diagnostics predicates accept the targets all, entity(Entity), file(File), directory(Directory), rdirectory(Directory), library(Library), and rlibrary(Library). The entity(Entity) target requires a loaded object or category. The file(File) target accepts a loaded source file specified by name, basename, full path, or library notation. Directory and library targets restrict diagnostics to loaded files found in the given directory, recursively in its sub-directories, in the given library, or recursively in its sub-libraries.
- Known issues: Use of local meta-calls with goal arguments only know at runtime can result in false positives. Calls from non-standard meta-predicates may be missed if the meta-calls are not optimized.

- Requirements: Source files must be compiled with the source_data flag turned on. To avoid false positives do to meta-calls, compilation of source files with the optimized flag turned on is also advised.

Inherited public predicates:

check_option/1 check_options/1 default_option/1 default_options/1 diagnostic/2 diagnostic/3 diagnostic_rule/5 diagnostic_rules/1 diagnostic_target/1 diagnostics/2 diagnostics/3 diagnostics_preflight/2 diagnostics_preflight/3 diagnostics_summary/2 diagnostics_summary/3 diagnostics_tool/5 option/2 option/3 valid_option/1 valid_options/1

- Public predicates
  - entity/1
  - entity/2
  - file/2
  - file/1
  - directory/2
  - directory/1
  - rdirectory/2
  - rdirectory/1
  - library/2
  - library/1
  - rlibrary/2
  - rlibrary/1
  - all/1
  - all/0
  - predicates/2
  - predicates/3
  - predicate/2
  - predicate/3
- Protected predicates
- Private predicates
- Operators

Public predicates

entity/1

Scans a loaded entity for dead code. Fails if the entity does not exist.

Compilation flags:
    static

Template:
    entity(Entity)
Mode and number of proofs:
    entity(+entity_identifier) - zero_or_one

---

entity/2

Scans a loaded entity for dead code using the given options. Fails if the entity does not exist.

Compilation flags:
    static

Template:
    entity(Entity,Options)
Mode and number of proofs:
    entity(+entity_identifier,+list(compound)) - zero_or_one

---

file/2

Scans all entities in a loaded source file for dead code using the given options. The file can be given by name, basename, full path, or using library notation. Fails if the file is not loaded.

Compilation flags:
    static

Template:
    file(File,Options)
Mode and number of proofs:

file(+atom,+list(compound)) - zero_or_one

---

file/1

Scans all entities in a loaded source file for dead code using default options. The file can be given by name, basename, full path, or using library notation. Fails if the file is not loaded.

Compilation flags:
static

Template:
file(File)
Mode and number of proofs:
file(+atom) - zero_or_one

---

directory/2

Scans all entities in all loaded files from a given directory for dead code using the given options.

Compilation flags:
static

Template:
directory(Directory,Options)
Mode and number of proofs:
directory(+atom,+list(compound)) - one

---

directory/1

Scans all entities in all loaded files from a given directory for dead code using default options.

Compilation flags:
static

Template:
directory(Directory)
Mode and number of proofs:
directory(+atom) - one

---

rdirectory/2

Scans all entities in all loaded files from a given directory and its sub-directories for dead code using the given options.

Compilation flags:
static

Template:
rdirectory(Directory,Options)
Mode and number of proofs:
rdirectory(+atom,+list(compound)) - one

---

rdirectory/1

Scans all entities in all loaded files from a given directory and its sub-directories for dead code using default options.

Compilation flags:
static

Template:
rdirectory(Directory)
Mode and number of proofs:
rdirectory(+atom) - one

---

library/2

Scans all entities in all loaded files from a given library for dead code using the given options.

Compilation flags:
    static

Template:
    library(Library,Options)
Mode and number of proofs:
    library(+atom,+list(compound)) - one

---

library/1

Scans all entities in all loaded files from a given library for dead code using default options.

Compilation flags:
    static

Template:
    library(Library)
Mode and number of proofs:
    library(+atom) - one

---

rlibrary/2

Scans all entities in all loaded files in a loaded library and its sub-libraries for dead code using the given options.

Compilation flags:
    static

Template:
    rlibrary(Library,Options)
Mode and number of proofs:
    rlibrary(+atom,+list(compound)) - one

---

rlibrary/1

Scans all entities in all loaded files in a loaded library and its sub-libraries for dead code using default options.

Compilation flags:
    static

Template:
    rlibrary(Library)
Mode and number of proofs:
    rlibrary(+atom) - one

---

all/1

Scans all entities for dead code using the given options.

Compilation flags:
    static

Template:
    all(Options)
Mode and number of proofs:
    all(+list(compound)) - one

---

all/0

Scans all entities for dead code using default options.

Compilation flags:
    static

Mode and number of proofs:
    all - one

---

predicates/2

Returns an ordered set of local predicates (and non-terminals) that are not used, directly or indirectly, by scoped predicates for a loaded entity.

Compilation flags:
    static

Template:
    predicates(Entity,Predicates)
Mode and number of proofs:
    predicates(+entity_identifier,-list(predicate_indicator)) - one

---

predicates/3

Returns an ordered set of local predicates (and non-terminals) that are not used, directly or indirectly, by scoped predicates for a loaded entity using the given options.

Compilation flags:
    static

Template:
    predicates(Entity,Predicates,Options)
Mode and number of proofs:
    predicates(+entity_identifier,-list(predicate_indicator),+list(compound)) - one

---

predicate/2

Enumerates, by backtracking, local predicates (and non-terminals) that are not used, directly or indirectly, by scoped predicates for a loaded entity.

Compilation flags:
    static

Template:
    predicate(Entity,Predicate)
Mode and number of proofs:

predicate(+entity_identifier,?predicate_indicator) - zero_or_more

---

predicate/3

Enumerates, by backtracking, local predicates (and non-terminals) that are not used, directly or indirectly, by scoped predicates for a loaded entity using the given options.

Compilation flags:
static

Template:
predicate(Entity,Predicate,Options)
Mode and number of proofs:
predicate(+entity_identifier,?predicate_indicator,+list(compound)) - zero_or_more

---

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.49.2 dead_code_scanner_messages

Logtalk dead_code_scanner tool default message translations.

Availability:
logtalk_load(dead_code_scanner(loader))

Author: Barry Evans and Paulo Moura
Version: 0:9:1

Date: 2026-05-05

Compilation flags:
static

Provides:
logtalk::message_prefix_stream/4
logtalk::message_tokens//2

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.50 debug_messages

object

1.50.1 debug_messages

Supports selective enabling and disabling of debug and debug(Group) messages.

Availability:
logtalk_load(debug_messages(loader))

Author: Paulo Moura
Version: 1:0:1
Date: 2022-05-05

Compilation flags:
static, context_switching_calls

Provides:
logtalk::message_hook/4
Uses:
logtalk

Remarks:

- Limitations: Debug messages are suppressed by the compiler when the optimize flag is turned on and thus cannot be enabled in this case.

Inherited public predicates:
(none)

- Public predicates
  - enable/1
  - disable/1
  - enabled/1

    - enable/2
    - disable/2
    - enabled/2
- Protected predicates
- Private predicates
    - enabled_/1
    - enabled_/2
- Operators

Public predicates

enable/1

Enables all debug and debug(Group) messages for the given component.

Compilation flags:
    static

Template:
    enable(Component)
Mode and number of proofs:
    enable(@term) - one

---

disable/1

Disables all debug and debug(Group) messages for the given component.

Compilation flags:
    static

Template:
    disable(Component)
Mode and number of proofs:
    disable(@term) - one

---

enabled/1

Enumerates by backtracking the components with enabled debug and debug(Group) messages.

Compilation flags:
    static

Template:
    enabled(Component)
Mode and number of proofs:
    enabled(?term) - zero_or_more

---

enable/2

Enables debug(Group) messages for the given component and group.

Compilation flags:
    static

Template:
    enable(Component,Group)
Mode and number of proofs:
    enable(@term,@term) - one

---

disable/2

Disables debug(Group) messages for the given component and group.

Compilation flags:
    static

Template:
    disable(Component,Group)
Mode and number of proofs:
    disable(@term,@term) - one

---

enabled/2

Enumerates by backtracking the enabled debug(Group) messages for each component.

Compilation flags:
    static

Template:
    enabled(Component,Group)
Mode and number of proofs:
    enabled(?term,?term) - zero_or_more

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

enabled_/1

Table of components with currently enabled debug and debug(Group) messages.

Compilation flags:
    dynamic

Template:
    enabled_(Component)
Mode and number of proofs:
    enabled_(?term) - zero_or_more

---

enabled_/2

Table of currently enabled debug(Group) per component.

Compilation flags:
    dynamic

Template:
enabled_(Component,Group)
Mode and number of proofs:
enabled_(?term,?term) - zero_or_more

Operators

(none)

1.51 debugger

object

1.51.1 debugger

Command-line debugger based on an extended procedure box model supporting execution tracing and spy points.

Availability:
logtalk_load(debugger(loader))

Author: Paulo Moura
Version: 8:0:1
Date: 2026-02-07

Compilation flags:
static, context_switching_calls

Implements:
public debuggerp
Provides:
logtalk::debug_handler/1
logtalk::debug_handler/3
Uses:
logtalk

Remarks:
(none)

Inherited public predicates:

debug/0 debugging/0 debugging/1 leash/1 leashing/1 log/3 logging/3 nodebug/0 nolog/3 nologall/0 nospy/1 nospy/3 nospy/4 nospyall/0 notrace/0 reset/0 set_write_max_depth/1 spy/1 spy/3 spy/4 spying/1 spying/3 spying/4 trace/0 write_max_depth/1

- Public predicates
- Protected predicates
- Private predicates
  - debugging_/0
  - tracing_/0
  - explicit_tracing_/0
  - skipping_/0
  - skipping_unleashed_/1
  - quasi_skipping_/0
  - leaping_/1
  - leashing_/1
  - invocation_number_/1
  - jump_to_invocation_number_/1
  - zap_to_port_/1
  - write_max_depth_/1
  - log_point_/3
  - clause_breakpoint_/2
  - predicate_breakpoint_/3
  - entity_predicate_breakpoint_/4
  - context_breakpoint_/4
  - conditional_breakpoint_/3
  - triggered_breakpoint_/4
  - triggered_breakpoint_enabled_/2
  - file_line_hit_count_/3
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

debugging_/0

True iff debug is on.

Compilation flags:
    dynamic

Mode and number of proofs:
    debugging_ - zero_or_one

---

tracing_/0

True iff tracing is on.

Compilation flags:
    dynamic

Mode and number of proofs:
    tracing_ - zero_or_one

---

explicit_tracing_/0

True iff tracing is on due to a call to the trace/0 predicate.

Compilation flags:
    dynamic

Mode and number of proofs:
explicit_tracing_ - zero_or_one

---

skipping_/0

True iff skipping.

Compilation flags:
dynamic

Mode and number of proofs:
skipping_ - zero_or_one

---

skipping_unleashed_/1

True iff skipping (a goal with invocation number N) but showing intermediate ports as unleashed.

Compilation flags:
dynamic

Template:
skipping_unleashed_(N)
Mode and number of proofs:
skipping_unleashed_(?integer) - zero_or_one

---

quasi_skipping_/0

True iff quasi-skipping.

Compilation flags:
dynamic

Mode and number of proofs:

quasi_skipping_ - zero_or_one

---

leaping_/1

True iff leaping in tracing or debugging mode.

Compilation flags:
dynamic

Template:
leaping_(Mode)
Mode and number of proofs:
leaping_(?atom) - zero_or_one

---

leashing_/1

Table of currently leashed ports.

Compilation flags:
dynamic

Template:
leashing_(Port)
Mode and number of proofs:
leashing_(?atom) - zero_or_more

---

invocation_number_/1

Current call stack invocation number.

Compilation flags:
dynamic

Template:

invocation_number_(N)
Mode and number of proofs:
invocation_number_(?integer) - zero_or_one

---

jump_to_invocation_number_/1

Invocation number to jump to.

Compilation flags:
dynamic

Template:
jump_to_invocation_number_(N)
Mode and number of proofs:
jump_to_invocation_number_(?integer) - zero_or_one

---

zap_to_port_/1

Port to zap to.

Compilation flags:
dynamic

Template:
zap_to_port_(Port)
Mode and number of proofs:
zap_to_port_(?integer) - zero_or_one

---

write_max_depth_/1

Current term write maximum depth.

Compilation flags:
dynamic

Template:
write_max_depth_(MaxDepth)
Mode and number of proofs:
write_max_depth_(?non_negative_integer) - zero_or_one

---

log_point_/3

Table of log points.

Compilation flags:
dynamic

Template:
log_point_(Entity,Line,Message)
Mode and number of proofs:
log_point_(?object_identifier,?integer,?atom) - zero_or_more
log_point_(?category_identifier,?integer,?atom) - zero_or_more

---

clause_breakpoint_/2

Table of clause breakpoints.

Compilation flags:
dynamic

Template:
clause_breakpoint_(Entity,Line)
Mode and number of proofs:
clause_breakpoint_(?object_identifier,?integer) - zero_or_more
clause_breakpoint_(?category_identifier,?integer) - zero_or_more

---

predicate_breakpoint_/3

Table of predicate breakpoints.

Compilation flags:
    dynamic

Template:
    predicate_breakpoint_(Functor,Arity,Original)
Mode and number of proofs:
    predicate_breakpoint_(?atom,?integer,?predicate_indicator) - zero_or_more
    predicate_breakpoint_(?atom,?integer,?non_terminal_indicator) - zero_or_more

---

entity_predicate_breakpoint_/4

Table of entity predicate breakpoints.

Compilation flags:
    dynamic

Template:
    entity_predicate_breakpoint_(Entity,Functor,Arity,Original)
Mode and number of proofs:
    entity_predicate_breakpoint_(?callable,?atom,?integer,?qualified_predicate_indicator) - zero_or_more
    entity_predicate_breakpoint_(?callable,?atom,?integer,?qualified_non_terminal_indicator) - zero_or_more

---

context_breakpoint_/4

Table of context breakpoints.

Compilation flags:
dynamic

Template:
context_breakpoint_(Sender,This,Self,Goal)
Mode and number of proofs:
context_breakpoint_(?object_identifier,?object_identifier,?object_identifier,?callable) - zero_or_more

---

conditional_breakpoint_/3

Table of conditional breakpoints.

Compilation flags:
dynamic

Template:
conditional_breakpoint_(Entity,Line,Condition)
Mode and number of proofs:
conditional_breakpoint_(?object_identifier,?integer,?callable) - zero_or_more
conditional_breakpoint_(?category_identifier,?integer,?callable) - zero_or_more

---

triggered_breakpoint_/4

Table of defined triggered breakpoints.

Compilation flags:
dynamic

Template:
triggered_breakpoint_(Entity,Line,TriggerEntity,TriggerLine)
Mode and number of proofs:
triggered_breakpoint_(?object_identifier,?integer,?object_identifier,?integer) - zero_or_more

triggered_breakpoint_(?object_identifier,?integer,?category_identifier,?integer) - zero_or_more
triggered_breakpoint_(?category_identifier,?integer,?object_identifier,?integer) - zero_or_more
triggered_breakpoint_(?category_identifier,?integer,?category_identifier,?integer) - zero_or_more

---

triggered_breakpoint_enabled_/2

Table of enabled triggered breakpoints.

Compilation flags:
dynamic

Template:
triggered_breakpoint_enabled_(Entity,Line)
Mode and number of proofs:
triggered_breakpoint_enabled_(?object_identifier,?integer) - zero_or_more
triggered_breakpoint_enabled_(?category_identifier,?integer) - zero_or_more

---

file_line_hit_count_/3

Table of file and line hit counts (successful unifications with clause heads).

Compilation flags:
dynamic

Template:
file_line_hit_count_(File,Line,Count)
Mode and number of proofs:
file_line_hit_count_(?atom,?integer,?integer) - zero_or_one

Operators

(none)

category

1.51.2 debugger_messages

Logtalk debugger tool default message translations.

Availability:
logtalk_load(debugger(loader))

Author: Paulo Moura
Version: 4:0:0
Date: 2025-12-18

Compilation flags:
static

Provides:
logtalk::message_prefix_stream/4
logtalk::question_prompt_stream/4
logtalk::message_tokens//2

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.51.3 debuggerp

Debugger protocol.

Availability:
logtalk_load(debugger(loader))

Author: Paulo Moura
Version: 3:6:0
Date: 2025-09-05

Compilation flags:
static

Dependencies:
(none)

Remarks:

- Debugger help: Type the character h (condensed help) or the character ? (extended help) at a leashed port.
- Predicate breakpoint: Specified as a ground term Functor/Arity.
- Non-terminal breakpoint: Specified as a ground term Functor//Arity.
- Entity predicate breakpoint: Specified as a term Entity::Functor/Arity. Entity must be an object or category and may not be ground if parametric.

- Entity non-terminal breakpoint: Specified as a term Entity::Functor//Arity. Entity must be an object or category and may not be ground if parametric.
- Clause breakpoint: Specified as an Entity-Line term with both Entity and Line bound. Line must be the first source file line of an entity clause.
- Conditional breakpoint: Specified using Entity and Line for the clause head and a condition. Line must be the first source file line of an entity clause.
- Hit count breakpoint: Specified using Entity and Line for the clause head and an unification count expression as a condition. Line must be the first source file line of an entity clause.
- Triggered breakpoint: Specified using Entity and Line for the clause head and another breakpoint as a condition. Line must be the first source file line of an entity clause.
- Context breakpoint: Specified as a (Sender, This, Self, Goal) tuple.
- Log point: Specified using Entity and Line for the clause head and a message.
- Leash port shorthands: none - [], loose - [fact,rule,call], half - [fact,rule,call,redo], tight - [fact,rule,call,redo,fail,exception], and full - [fact,rule,call,exit,redo,fail,exception].

Inherited public predicates:
(none)

- Public predicates
  - reset/0
  - debug/0
  - nodebug/0
  - debugging/0
  - debugging/1
  - trace/0
  - notrace/0
  - leash/1
  - leashing/1
  - spy/1
  - spying/1
  - nospy/1
  - spy/3
  - spying/3
  - nospy/3
  - spy/4
  - spying/4
  - nospy/4

  - nospyall/0
  - log/3
  - logging/3
  - nolog/3
  - nologall/0
  - write_max_depth/1
  - set_write_max_depth/1
- Protected predicates
- Private predicates
- Operators

Public predicates

reset/0

Resets all debugging settings (including breakpoints, log points, and leashed ports) and turns off debugging.

Compilation flags:
    static

Mode and number of proofs:
    reset - one

See also:
    nospyall/0

---

debug/0

Starts debugging for all defined breakpoints.

Compilation flags:
    static

Mode and number of proofs:
    debug - one

---

nodebug/0

Stops debugging for all defined breakpoints. Also turns off tracing. Does not remove defined breakpoints.

Compilation flags:
    static

Mode and number of proofs:
    nodebug - one

See also:
    reset/0

---

debugging/0

Reports current debugging settings, including breakpoints and log points.

Compilation flags:
    static

Mode and number of proofs:
    debugging - one

---

debugging/1

Enumerates, by backtracking, all entities compiled in debug mode.

Compilation flags:
    static

Template:
    debugging(Entity)
Mode and number of proofs:

debugging(?entity_identifier) - zero_or_more

---

trace/0

Starts tracing all calls compiled in debug mode.

Compilation flags:
static

Mode and number of proofs:
trace - one

---

notrace/0

Stops tracing of calls compiled in debug mode. Debugger will still stop at defined breakpoints.

Compilation flags:
static

Mode and number of proofs:
notrace - one

---

leash/1

Sets the debugger leash ports using an abbreviation (none, loose, half, tight, or full) or a list of ports (valid ports are fact, rule, call, exit, redo, fail, and exception).

Compilation flags:
static

Template:
leash(Ports)
Mode and number of proofs:
leash(+atom) - one

leash(+list(atom)) - one

---

leashing/1

Enumerates, by backtracking, all leashed ports (valid ports are fact, rule, call, exit, redo, fail, and exception).

Compilation flags:
static

Template:
leashing(Port)
Mode and number of proofs:
leashing(?atom) - zero_or_more

---

spy/1

Sets a predicate or clause breakpoint (removing any existing log point or breakpoint defined for the same location, or a list of breakpoints. Fails if a breakpoint is invalid.

Compilation flags:
static

Template:
spy(Breakpoint)
Mode and number of proofs:
spy(@spy_point) - zero_or_one
spy(@list(spy_point)) - zero_or_one

---

spying/1

Enumerates, by backtracking, all defined predicate and clause breakpoints.

Compilation flags:
    static

Template:
    spying(Breakpoint)
Mode and number of proofs:
    spying(?spy_point) - zero_or_more

---

nospy/1

Removes all matching predicate and clause breakpoints.

Compilation flags:
    static

Template:
    nospy(Breakpoint)
Mode and number of proofs:
    nospy(@var) - one
    nospy(@spy_point) - one
    nospy(@list(spy_point)) - one

---

spy/3

Sets a conditional or triggered breakpoint (removing any existing log point or breakpoint defined for the same location) at a clause head. The condition can be a unification count expression, a lambda expression, or another breakpoint. Fails if the breakpoint is invalid.

Compilation flags:
    static

Template:
    spy(Entity,Line,Condition)

Mode and number of proofs:
spy(+atom,+integer,@callable) - zero_or_one

Remarks:

- Unification count expression conditions: >(Count), >=(Count), =:=(Count), =<(Count), <(Count), mod(M), and Count.
- Lambda expression conditions: [Count,N,Goal]>>Condition and [Goal]>>Condition where Count is the unification count, N is the invocation number, and Goal is the goal that unified with the clause head; Condition is called in the context of user.
- Triggered breakpoint conditions: Entity-Line.

---

spying/3

Enumerates, by backtracking, all conditional and triggered breakpoints.

Compilation flags:
static

Template:
spying(Entity,Line,Condition)
Mode and number of proofs:
spying(?atom,?integer,?callable) - zero_or_more

---

nospy/3

Removes all matching conditional and triggered breakpoints.

Compilation flags:
static

Template:
nospy(Entity,Line,Condition)
Mode and number of proofs:
nospy(@term,@term,@term) - one

---

spy/4

Sets a context breakpoint.

Compilation flags:
    static

Template:
    spy(Sender,This,Self,Goal)
Mode and number of proofs:
    spy(@term,@term,@term,@term) - one

---

spying/4

Enumerates, by backtracking, all defined context breakpoints.

Compilation flags:
    static

Template:
    spying(Sender,This,Self,Goal)
Mode and number of proofs:
    spying(?term,?term,?term,?term) - zero_or_more

---

nospy/4

Removes all matching context breakpoints.

Compilation flags:
    static

Template:
    nospy(Sender,This,Self,Goal)
Mode and number of proofs:
    nospy(@term,@term,@term,@term) - one

---

nospyall/0

Removes all breakpoints and log points.

Compilation flags:
    static

Mode and number of proofs:
    nospyall - one

See also:
    reset/0

---

log/3

Sets a log point (removing any existing breakpoint defined for the same location) at a clause head. Fails if the log point is invalid.

Compilation flags:
    static

Template:
    log(Entity,Line,Message)
Mode and number of proofs:
    log(@object_identifier,+integer,+atom) - zero_or_one
    log(@category_identifier,+integer,+atom) - zero_or_one

---

logging/3

Enumerates, by backtracking, all defined log points.

Compilation flags:
    static

Template:
    logging(Entity,Line,Message)

Mode and number of proofs:
logging(?object_identifier,?integer,?atom) - zero_or_more
logging(?category_identifier,?integer,?atom) - zero_or_more

---

nolog/3

Removes all matching log points.

Compilation flags:
static

Template:
nolog(Entity,Line,Message)
Mode and number of proofs:
nolog(@var_or(object_identifier),@var_or(integer),@var_or(atom)) - one
nolog(@var_or(category_identifier),@var_or(integer),@var_or(atom)) - one

---

nologall/0

Removes all log points.

Compilation flags:
static

Mode and number of proofs:
nologall - one

See also:
reset/0

---

write_max_depth/1

Current term write maximum depth. When not defined, the backend default is used.

Compilation flags:
    static

Template:
    write_max_depth(MaxDepth)
Mode and number of proofs:
    write_max_depth(?non_negative_integer) - zero_or_one

---

set_write_max_depth/1

Sets the default term maximum write depth. For most backends, a value of zero means that the whole term is written.

Compilation flags:
    static

Template:
    set_write_max_depth(MaxDepth)
Mode and number of proofs:
    set_write_max_depth(+non_negative_integer) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

debugger

object

1.51.4 dump_trace

Simple solution for redirecting a debugger trace to a file.

Availability:
logtalk_load(debugger(loader))

Author: Paulo Moura
Version: 1:0:1
Date: 2021-11-12

Compilation flags:
static, context_switching_calls

Uses:
debugger

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - start_redirect_to_file/2
  - stop_redirect_to_file/0
- Protected predicates
- Private predicates
- Operators

Public predicates

start_redirect_to_file/2

Starts redirecting debugger trace messages to a file.

Compilation flags:
    static

Template:
    start_redirect_to_file(File,Goal)
Meta-predicate template:
    start_redirect_to_file(*,0)
Mode and number of proofs:
    start_redirect_to_file(+atom,+callable) - zero_or_more

---

stop_redirect_to_file/0

Stops redirecting debugger trace messages to a file.

Compilation flags:
    static

Mode and number of proofs:
    stop_redirect_to_file - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.52 dependents

category

1.52.1 observer

Smalltalk dependent protocol.

Availability:
logtalk_load(dependents(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2003-02-09

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - update/1
- Protected predicates
- Private predicates
- Operators

Public predicates

update/1

Called when an observed object is updated.

Compilation flags:
    static

Template:
    update(Change)
Mode and number of proofs:
    update(?nonvar) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

subject

category

1.52.2 subject

Smalltalk dependent handling predicates.

Availability:
    logtalk_load(dependents(loader))

Author: Paulo Moura
Version: 1:0:0

Date: 2003-02-09

Compilation flags:
    static

Dependencies:
    (none)

Remarks:
    (none)

Inherited public predicates:
    (none)

- Public predicates
  - changed/0
  - changed/1
  - dependents/1
  - addDependent/1
  - removeDependent/1
- Protected predicates
- Private predicates
  - dependent_/1
- Operators

Public predicates

changed/0

Receiver changed in some way. Notify all dependents.

Compilation flags:
    static

Mode and number of proofs:
    changed - one

---

changed/1

Receiver changed as specified in the argument. Notify all dependents.

Compilation flags:
    static

Template:
    changed(Change)
Mode and number of proofs:
    changed(?nonvar) - one

---

dependents/1

Returns a list of all dependent objects.

Compilation flags:
    static

Template:
    dependents(Dependents)
Mode and number of proofs:
    dependents(-list) - one

---

addDependent/1

Adds a new dependent object.

Compilation flags:
    static

Template:
    addDependent(Dependent)
Mode and number of proofs:

addDependent(@object) - one

---

removeDependent/1

Removes a dependent object.

Compilation flags:
static

Template:
removeDependent(Dependent)
Mode and number of proofs:
removeDependent(?object) - zero_or_more

Protected predicates

(none)

Private predicates

dependent_/1

Table of dependent objects.

Compilation flags:
dynamic

Template:
dependent_(Dependent)
Mode and number of proofs:
dependent_(?object) - zero_or_more

Operators

(none)

See also

observer

1.53 deques

object

1.53.1 deque

Double-ended queue (deque) implementation using difference lists to provide O(1) operations at both ends.

Availability:
logtalk_load(deques(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-09

Compilation flags:
static, context_switching_calls

Implements:
public deque_protocol
Extends:
public compound
Uses:
list

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 as_deque/2 as_list/2 check/1 depth/2 empty/1 ground/1 length/2 map/2 map/3 new/1 numbervars/1 numbervars/3 occurs/2 peek_back/2 peek_front/2 pop_back/3 pop_front/3 push_back/3 push_front/3 singletons/2 subsumes/2 subterm/2 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.53.2 deque_protocol

Extracted protocol entity.

Availability:
logtalk_load(deques(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-08

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - empty/1
  - push_front/3
  - push_back/3
  - pop_front/3
  - pop_back/3
  - peek_front/2
  - peek_back/2
  - length/2
  - map/2
  - map/3
  - as_list/2
  - as_deque/2
- Protected predicates
- Private predicates
- Operators

Public predicates

empty/1

True iff the deque is empty.

Compilation flags:
static

Template:
empty(Deque)
Mode and number of proofs:
empty(+deque) - zero_or_one

push_front/3

Adds an element to the front of the deque.

Compilation flags:
    static

Template:
    push_front(Element,DequeIn,DequeOut)
Mode and number of proofs:
    push_front(+term,+deque,-deque) - one

---

push_back/3

Adds an element to the back of the deque.

Compilation flags:
    static

Template:
    push_back(Element,DequeIn,DequeOut)
Mode and number of proofs:
    push_back(+term,+deque,-deque) - one

---

pop_front/3

Removes and returns the front element.

Compilation flags:
    static

Template:
    pop_front(DequeIn,Element,DequeOut)
Mode and number of proofs:
    pop_front(+deque,-term,-deque) - zero_or_one

---

pop_back/3

Removes and returns the back element.

Compilation flags:
    static

Template:
    pop_back(DequeIn,Element,DequeOut)
Mode and number of proofs:
    pop_back(+deque,-term,-deque) - zero_or_one

---

peek_front/2

Returns the front element without removing it.

Compilation flags:
    static

Template:
    peek_front(Deque,Element)
Mode and number of proofs:
    peek_front(+deque,-term) - zero_or_one

---

peek_back/2

Returns the back element without removing it.

Compilation flags:
    static

Template:
    peek_back(Deque,Element)
Mode and number of proofs:
    peek_back(+deque,-term) - zero_or_one

---

length/2

Returns the number of elements in the deque.

Compilation flags:
    static

Template:
    length(Deque,Length)
Mode and number of proofs:
    length(+deque,-integer) - one

---

map/2

Applies a closure to all elements of a deque.

Compilation flags:
    static

Template:
    map(Closure,Deque)
Meta-predicate template:
    map(1,*)
Mode and number of proofs:
    map(+callable,+deque) - zero_or_one

---

map/3

Applies a closure to all elements of a deque constructing a new deque.

Compilation flags:
    static

Template:
    map(Closure,Deque,NewQueue)
Meta-predicate template:
    map(2,*,*)

Mode and number of proofs:
    map(+callable,+deque,?deque) - zero_or_one

---

as_list/2

Converts a deque to a list.

Compilation flags:
    static

Template:
    as_list(Deque,List)
Mode and number of proofs:
    as_list(+deque,-list) - one

---

as_deque/2

Converts a list to a deque.

Compilation flags:
    static

Template:
    as_deque(List,Deque)
Mode and number of proofs:
    as_deque(+list,-deque) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.54 derangements

object

1.54.1 derangements

Implementation of derangement operations over lists.

Availability:
logtalk_load(derangements(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-12

Compilation flags:
static, context_switching_calls

Implements:
public derangements_protocol
Uses:
fast_random(Algorithm)
list
natural
permutations

Remarks:
(none)

Inherited public predicates:
count_derangements/2 count_distinct_derangements/2 count_partial_derangements/3 derangement/2 derangement/3 derangement_index/3 derangement_index/4 derangements/2 derangements/3 distinct_derangement/2 distinct_derangement/3 distinct_derangement_index/3 distinct_derangements/2 distinct_derangements/3 next_derangement/3 nth_derangement/3 nth_derangement/4 nth_distinct_derangement/3 previous_derangement/3

random_derangement/2 random_distinct_derangement/2 sample_derangements/3
sample_distinct_derangements/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.54.2 derangements_protocol

Protocol for derangement operations over lists.

Availability:
logtalk_load(derangements(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-12

Compilation flags:
static

Dependencies:

(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - derangements/2
  - derangement/2
  - derangements/3
  - derangement/3
  - distinct_derangements/2
  - distinct_derangement/2
  - distinct_derangements/3
  - distinct_derangement/3
  - nth_derangement/3
  - nth_derangement/4
  - derangement_index/3
  - derangement_index/4
  - count_derangements/2
  - count_partial_derangements/3
  - count_distinct_derangements/2
  - nth_distinct_derangement/3
  - distinct_derangement_index/3
  - random_derangement/2
  - sample_derangements/3
  - random_distinct_derangement/2
  - sample_distinct_derangements/3
  - next_derangement/3
  - previous_derangement/3
- Protected predicates
- Private predicates
- Operators

Public predicates

derangements/2

Generates all derangements of a list using default order.

Compilation flags:
    static

Template:
    derangements(List,Derangements)
Mode and number of proofs:
    derangements(+list,-list) - one

---

derangement/2

True iff the second argument is a derangement of the first argument using default order.

Compilation flags:
    static

Template:
    derangement(List,Derangement)
Mode and number of proofs:
    derangement(+list,-list) - zero_or_more

---

derangements/3

Generates all derangements with the given order: default, lexicographic, or shortlex.

Compilation flags:
    static

Template:
    derangements(List,Order,Derangements)
Mode and number of proofs:
    derangements(+list,+atom,-list) - one

---

derangement/3

True iff the third argument is a derangement with the given order: default, lexicographic, or shortlex.

Compilation flags:
    static

Template:
    derangement(List,Order,Derangement)
Mode and number of proofs:
    derangement(+list,+atom,-list) - zero_or_more

---

distinct_derangements/2

Generates all distinct derangements of a list using default order.

Compilation flags:
    static

Template:
    distinct_derangements(List,Derangements)
Mode and number of proofs:
    distinct_derangements(+list,-list) - one

---

distinct_derangement/2

True iff the second argument is a distinct derangement of the first argument using default order.

Compilation flags:
    static

Template:
    distinct_derangement(List,Derangement)
Mode and number of proofs:

distinct_derangement(+list,-list) - zero_or_more

---

distinct_derangements/3

Generates all distinct derangements with the given order: default, lexicographic, or shortlex.

Compilation flags:
static

Template:
distinct_derangements(List,Order,Derangements)
Mode and number of proofs:
distinct_derangements(+list,+atom,-list) - one

---

distinct_derangement/3

True iff the third argument is a distinct derangement with the given order: default, lexicographic, or shortlex.

Compilation flags:
static

Template:
distinct_derangement(List,Order,Derangement)
Mode and number of proofs:
distinct_derangement(+list,+atom,-list) - zero_or_more

---

nth_derangement/3

Returns the derangement at a given zero-based index using default order.

Compilation flags:
static

Template:

nth_derangement(List,Index,Derangement)
Mode and number of proofs:
nth_derangement(+list,+integer,-list) - zero_or_one

---

nth_derangement/4

Returns the derangement at a given zero-based index in the given order: default, lexicographic, or shortlex.

Compilation flags:
static

Template:
nth_derangement(List,Order,Index,Derangement)
Mode and number of proofs:
nth_derangement(+list,+atom,+integer,-list) - zero_or_one

---

derangement_index/3

Returns the zero-based index of a derangement using default order.

Compilation flags:
static

Template:
derangement_index(List,Derangement,Index)
Mode and number of proofs:
derangement_index(+list,+list,-integer) - zero_or_one

---

derangement_index/4

Returns the zero-based index of a derangement in the given order: default, lexicographic, or shortlex.

Compilation flags:
    static

Template:
    derangement_index(List,Order,Derangement,Index)
Mode and number of proofs:
    derangement_index(+list,+atom,+list,-integer) - zero_or_one

---

count_derangements/2

Counts the number of derangements of a list.

Compilation flags:
    static

Template:
    count_derangements(List,Count)
Mode and number of proofs:
    count_derangements(+list,-integer) - one

---

count_partial_derangements/3

Counts the number of permutations of a list with exactly the given number of fixed points.

Compilation flags:
    static

Template:
    count_partial_derangements(FixedPoints,List,Count)
Mode and number of proofs:
    count_partial_derangements(+integer,+list,-integer) - one

---

count_distinct_derangements/2

Counts the number of distinct derangements of a list (deduplicating equal-valued derangements).

Compilation flags:
    static

Template:
    count_distinct_derangements(List,Count)
Mode and number of proofs:
    count_distinct_derangements(+list,-integer) - one

---

nth_distinct_derangement/3

Returns the distinct derangement at a given zero-based index in default generation order.

Compilation flags:
    static

Template:
    nth_distinct_derangement(List,Index,Derangement)
Mode and number of proofs:
    nth_distinct_derangement(+list,+integer,-list) - zero_or_one

---

distinct_derangement_index/3

Returns the zero-based index of a distinct derangement in default generation order.

Compilation flags:
    static

Template:
    distinct_derangement_index(List,Derangement,Index)
Mode and number of proofs:
    distinct_derangement_index(+list,+list,-integer) - zero_or_one

---

random_derangement/2

Returns a random derangement of a list.

Compilation flags:
    static

Template:
    random_derangement(List,Derangement)
Mode and number of proofs:
    random_derangement(+list,-list) - zero_or_one

---

sample_derangements/3

Returns SampleCount random derangements of a list, sampled with replacement.

Compilation flags:
    static

Template:
    sample_derangements(List,SampleCount,Samples)
Mode and number of proofs:
    sample_derangements(+list,+integer,-list) - zero_or_one

---

random_distinct_derangement/2

Returns a random distinct derangement of a list (deduplicating equal-valued derangements).

Compilation flags:
    static

Template:
    random_distinct_derangement(List,Derangement)
Mode and number of proofs:
    random_distinct_derangement(+list,-list) - zero_or_one

---

sample_distinct_derangements/3

Returns SampleCount random distinct derangements of a list, sampled with replacement after deduplicating equal-valued derangements.

Compilation flags:
    static

Template:
    sample_distinct_derangements(List,SampleCount,Samples)
Mode and number of proofs:
    sample_distinct_derangements(+list,+integer,-list) - zero_or_one

---

next_derangement/3

Returns the next distinct derangement value in lexicographic order induced by the first argument.

Compilation flags:
    static

Template:
    next_derangement(List,Derangement,Next)
Mode and number of proofs:
    next_derangement(+list,+list,-list) - zero_or_one

---

previous_derangement/3

Returns the previous distinct derangement value in lexicographic order induced by the first argument.

Compilation flags:
    static

Template:
    previous_derangement(List,Derangement,Previous)
Mode and number of proofs:
    previous_derangement(+list,+list,-list) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.55 diagrams

object

1.55.1 caller_diagram

Predicates for generating caller diagrams in DOT format.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-12

Compilation flags:
static, context_switching_calls

Extends:
public caller_diagram(dot)

Remarks:
(none)

Inherited public predicates:

all_files/0 all_files/1 all_libraries/0 all_libraries/1 check_option/1 check_options/1 default_option/1 default_options/1 diagram_description/1 diagram_name_suffix/1 directories/2 directories/3 directory/1 directory/2 directory/3 files/1 files/2 files/3 format_object/1 libraries/1 libraries/2 libraries/3 library/1 library/2 option/2 option/3 predicate/1 predicate/2 rdirectory/1 rdirectory/2 rdirectory/3 rlibrary/1 rlibrary/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

xref_diagram, entity_diagram

object

1.55.2 caller_diagram(Format)

- Format - Graph language file format.

Predicates for generating caller diagrams showing direct and indirect callers of a predicate or a non-terminal.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-12

Compilation flags:
static, context_switching_calls

Imports:
public diagram(Format)
Uses:
list
logtalk
user

Remarks:
(none)

Inherited public predicates:
all_files/0 all_files/1 all_libraries/0 all_libraries/1 check_option/1 check_options/1 default_option/1 default_options/1 diagram_description/1 diagram_name_suffix/1 directories/2 directories/3 directory/1 directory/2 directory/3 files/1 files/2 files/3 format_object/1 libraries/1 libraries/2 libraries/3 library/1 library/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory/3 rlibrary/1 rlibrary/2 valid_option/1 valid_options/1

- Public predicates
  - predicate/2
  - predicate/1
- Protected predicates
- Private predicates
  - included_caller_/1
- Operators

Public predicates

predicate/2

Creates a caller diagram for the given predicate or non-terminal using the specified options. Predicates are specified as Entity::Name/Arity. Non-terminals are specified as Entity::Name//Arity.

Compilation flags:
    static

Template:
    predicate(QualifiedIndicator,Options)
Mode and number of proofs:
    predicate(+qualified_predicate_indicator,+list(compound)) - one

---

predicate/1

Creates a caller diagram for the given predicate using default options. Predicates are specified as Entity::Name/Arity. Non-terminals are specified as Entity::Name//Arity.

Compilation flags:
    static

Template:
    predicate(QualifiedIndicator)
Mode and number of proofs:
    predicate(+qualified_predicate_indicator) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

included_caller_/1

Table of callers already included in the diagram.

Compilation flags:
dynamic

Template:
included_caller_(Caller)
Mode and number of proofs:
included_caller_(?qualified_predicate_indicator) - zero_or_more
included_caller_(?predicate_indicator) - zero_or_more

Operators

(none)

See also

xref_diagram(Format), entity_diagram(Format)

object

1.55.3 cytoscapejs_graph_language

Predicates for generating diagram files in the Cytoscape Exchange (CX2) JSON format.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-03-14

Compilation flags:
static, context_switching_calls

Implements:

public graph_language_protocol

Imports:

public options

Provides:

graph_language_registry::language_object/2

Uses:

list
os
term_io
user

Remarks:

(none)

Inherited public predicates:

check_option/1 check_options/1 default_option/1 default_options/1 edge/6 file_footer/3 file_header/3 graph_footer/5 graph_header/5 node/7 option/2 option/3 output_file_name/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
  - section_started_/1
  - in_nodes_section_/0
  - parent_stack_/1
  - edge_counter_/1
  - node_counter_/1
  - node_id_/2
  - containment_edge_/2
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

section_started_/1

Tracks whether a JSON array section (nodes or edges) has already emitted at least one element.

Compilation flags:
    dynamic

Template:
    section_started_(Section)
Mode and number of proofs:
    section_started_(?atom) - zero_or_more

---

in_nodes_section_/0

Flag indicating if the exporter is currently writing the nodes section.

Compilation flags:
    dynamic

Mode and number of proofs:
    in_nodes_section_ - zero_or_one

---

parent_stack_/1

Current stack of open graph container identifiers.

Compilation flags:
    dynamic

Template:
    parent_stack_(Stack)
Mode and number of proofs:
    parent_stack_(?list(nonvar)) - zero_or_one

---

edge_counter_/1

Current counter used to generate unique edge identifiers.

Compilation flags:
    dynamic

Template:
    edge_counter_(Counter)
Mode and number of proofs:
    edge_counter_(?integer) - zero_or_one

---

node_counter_/1

Current counter used to generate unique node identifiers.

Compilation flags:
    dynamic

Template:
    node_counter_(Counter)
Mode and number of proofs:
    node_counter_(?integer) - zero_or_one

---

node_id_/2

Maps source node identifiers to generated CX2 numeric identifiers.

Compilation flags:
    dynamic

Template:
    node_id_(Identifier,NumericId)
Mode and number of proofs:
    node_id_(?nonvar,?integer) - zero_or_more

---

containment_edge_/2

Stores pending containment edges between parent and child nodes.

Compilation flags:
    dynamic

Template:
    containment_edge_(ParentId,ChildId)
Mode and number of proofs:
    containment_edge_(?integer,?integer) - zero_or_more

Operators

(none)

object

1.55.4 d2_graph_language

Predicates for generating graph files in the DOT language (version 2.36.0 or later).

Availability:
    logtalk_load(diagrams(loader))

Author: Paulo Moura

Version: 1:5:0
Date: 2026-03-14

Compilation flags:
static, context_switching_calls

Implements:
public graph_language_protocol
Imports:
public options
Provides:
graph_language_registry::language_object/2
Uses:
list
os
term_io
user

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 default_option/1 default_options/1 edge/6 file_footer/3 file_header/3 graph_footer/5 graph_header/5 node/7 option/2 option/3 output_file_name/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.55.5 diagram(Format)

- Format - Graph language file format.

Common predicates for generating diagrams.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 3:18:0
Date: 2026-03-13

Compilation flags:
static

Extends:
public options
Provides:
logtalk::message_prefix_stream/4
logtalk::message_tokens//2
Uses:
graph_language_registry
list
logtalk
modules_diagram_support
os
pairs
type
user

Remarks:

(none)

Inherited public predicates:

check_option/1 check_options/1 default_option/1 default_options/1 option/2 option/3 valid_option/1 valid_options/1

- Public predicates
  - libraries/3
  - libraries/2
  - libraries/1
  - all_libraries/1
  - all_libraries/0
  - rlibrary/2
  - rlibrary/1
  - library/2
  - library/1
  - directories/3
  - directories/2
  - rdirectory/3
  - rdirectory/2
  - rdirectory/1
  - directory/3
  - directory/2
  - directory/1
  - files/3
  - files/2
  - files/1
  - all_files/1
  - all_files/0
  - format_object/1
  - diagram_description/1
  - diagram_name_suffix/1
- Protected predicates
  - diagram_caption/3

    - output_rlibrary/3
    - output_library/3
    - output_rdirectory/3
    - output_externals/1
    - output_files/2
    - output_file/4
    - output_sub_diagrams/1
    - reset/0
    - output_node/6
    - node/6
    - edge/5
    - output_edges/1
    - save_edge/5
    - output_missing_externals/1
    - not_excluded_file/4
    - output_file_path/4
    - locate_library/2
    - locate_directory/2
    - locate_file/5
    - ground_entity_identifier/3
    - filter_file_extension/3
    - filter_external_file_extension/3
    - add_link_options/3
    - supported_editor_url_scheme_prefix/1
    - omit_path_prefix/3
    - add_node_zoom_option/4
    - message_diagram_description/1
- Private predicates
    - node_/6
    - node_path_/2
    - edge_/5
    - cycle_edge_/2
- Operators

Public predicates

libraries/3

Creates a diagram for a set of libraries using the specified options. The Project argument is used as a prefix for the diagram file name.

Compilation flags:
    static

Template:
    libraries(Project,Libraries,Options)
Mode and number of proofs:
    libraries(+atom,+list(atom),+list(compound)) - one

---

libraries/2

Creates a diagram for a set of libraries using the default options. The Project argument is used as a prefix for the diagram file name.

Compilation flags:
    static

Template:
    libraries(Project,Libraries)
Mode and number of proofs:
    libraries(+atom,+list(atom)) - one

---

libraries/1

Creates a diagram for a set of libraries using the default options. The prefix libraries is used for the diagram file name.

Compilation flags:
    static

Template:

libraries(Libraries)
Mode and number of proofs:
libraries(+list(atom)) - one

---

all_libraries/1

Creates a diagram for all loaded libraries using the specified options.

Compilation flags:
static

Template:
all_libraries(Options)
Mode and number of proofs:
all_libraries(+list(compound)) - one

---

all_libraries/0

Creates a diagram for all loaded libraries using default options.

Compilation flags:
static

Mode and number of proofs:
all_libraries - one

---

rlibrary/2

Creates a diagram for a library and its sub-libraries using the specified options.

Compilation flags:
static

Template:

rlibrary(Library,Options)
Mode and number of proofs:
rlibrary(+atom,+list(compound)) - one

---

rlibrary/1

Creates a diagram for a library and its sub-libraries using default options.

Compilation flags:
static

Template:
rlibrary(Library)
Mode and number of proofs:
rlibrary(+atom) - one

---

library/2

Creates a diagram for a library using the specified options.

Compilation flags:
static

Template:
library(Library,Options)
Mode and number of proofs:
library(+atom,+list(compound)) - one

---

library/1

Creates a diagram for a library using default options.

Compilation flags:
    static

Template:
    library(Library)
Mode and number of proofs:
    library(+atom) - one

---

directories/3

Creates a diagram for a set of directories using the specified options. The Project argument is used as a prefix for the diagram file name.

Compilation flags:
    static

Template:
    directories(Project,Directories,Options)
Mode and number of proofs:
    directories(+atom,+list(atom),+list(compound)) - one

---

directories/2

Creates a diagram for a set of directories using the default options. The Project argument is used as a prefix for the diagram file name.

Compilation flags:
    static

Template:
    directories(Project,Directories)
Mode and number of proofs:
    directories(+atom,+list(atom)) - one

---

rdirectory/3

Creates a diagram for a directory and its sub-directories using the specified options. The Project argument is used as a prefix for the diagram file name.

Compilation flags:
    static

Template:
    rdirectory(Project,Directory,Options)
Mode and number of proofs:
    rdirectory(+atom,+atom,+list(compound)) - one

---

rdirectory/2

Creates a diagram for a directory and its sub-directories using default options. The Project argument is used as a prefix for the diagram file name.

Compilation flags:
    static

Template:
    rdirectory(Project,Directory)
Mode and number of proofs:
    rdirectory(+atom,+atom) - one

---

rdirectory/1

Creates a diagram for a directory and its sub-directories using default options. The name of the directory is used as a prefix for the diagram file name.

Compilation flags:
    static

Template:

rdirectory(Directory)
Mode and number of proofs:
rdirectory(+atom) - one

---

directory/3

Creates a diagram for a directory using the specified options. The Project argument is used as a prefix for the diagram file name.

Compilation flags:
static

Template:
directory(Project,Directory,Options)
Mode and number of proofs:
directory(+atom,+atom,+list(compound)) - one

---

directory/2

Creates a diagram for a directory using default options. The Project argument is used as a prefix for the diagram file name.

Compilation flags:
static

Template:
directory(Project,Directory)
Mode and number of proofs:
directory(+atom,+atom) - one

---

directory/1

Creates a diagram for a directory using default options. The name of the directory is used as a prefix for the diagram file name.

Compilation flags:
    static

Template:
    directory(Directory)
Mode and number of proofs:
    directory(+atom) - one

---

files/3

Creates a diagram for a set of files using the specified options. The file can be specified by name, basename, full path, or using library notation. The Project argument is used as a prefix for the diagram file name.

Compilation flags:
    static

Template:
    files(Project,Files,Options)
Mode and number of proofs:
    files(+atom,+list(atom),+list(compound)) - one

---

files/2

Creates a diagram for a set of files using the default options. The file can be specified by name, basename, full path, or using library notation. The Project argument is used as a prefix for the diagram file name.

Compilation flags:
    static

Template:
    files(Project,Files)
Mode and number of proofs:

files(+atom,+list(atom)) - one

---

files/1

Creates a diagram for a set of files using the default options. The file can be specified by name, basename, full path, or using library notation. The prefix files is used for the diagram file name.

Compilation flags:
static

Template:
files(Files)
Mode and number of proofs:
files(+list(atom)) - one

---

all_files/1

Creates a diagram for all loaded files using the specified options.

Compilation flags:
static

Template:
all_files(Options)
Mode and number of proofs:
all_files(+list(compound)) - one

---

all_files/0

Creates a diagram for all loaded files using default options.

Compilation flags:
static

Mode and number of proofs:
all_files - one

---

format_object/1

Returns the identifier of the object implementing the graph language currently being used. Fails if none is specified.

Compilation flags:
static

Template:
format_object(Object)
Mode and number of proofs:
format_object(-object_identifier) - zero_or_one

---

diagram_description/1

Returns the diagram description.

Compilation flags:
static

Template:
diagram_description(Description)
Mode and number of proofs:
diagram_description(-atom) - one

---

diagram_name_suffix/1

Returns the diagram name suffix.

Compilation flags:
    static

Template:
    diagram_name_suffix(Suffix)
Mode and number of proofs:
    diagram_name_suffix(-atom) - one

Protected predicates

diagram_caption/3

Creates a diagram caption from the diagram description and the subject and its kind.

Compilation flags:
    static

Template:
    diagram_caption(Kind,Subject,Description)
Mode and number of proofs:
    diagram_caption(+atom,+callable,-atom) - one

---

output_rlibrary/3

Generates diagram output for a library and its sub-libraries using the specified options.

Compilation flags:
    static

Template:
    output_rlibrary(Library,Path,Options)
Mode and number of proofs:
    output_rlibrary(+atom,+atom,+list(compound)) - one

---

output_library/3

Generates diagram output for a library using the specified options.

Compilation flags:
    static

Template:
    output_library(Library,Path,Options)
Mode and number of proofs:
    output_library(+atom,+atom,+list(compound)) - one

---

output_rdirectory/3

Generates diagram output for a directory and its sub-directories using the specified options.

Compilation flags:
    static

Template:
    output_rdirectory(Project,Path,Options)
Mode and number of proofs:
    output_rdirectory(+atom,+atom,+list(compound)) - one

---

output_externals/1

Output external nodes using the specified options depending on the value of the boolean option externals/1.

Compilation flags:
    static

Template:
    output_externals(Options)
Mode and number of proofs:

output_externals(+list(compound)) - one

---

output_files/2

Generates diagram output for a list of files using the specified options.

Compilation flags:
static

Template:
output_files(Files,Options)
Mode and number of proofs:
output_files(+list,+list(compound)) - one

---

output_file/4

Generates diagram output for a file using the specified options.

Compilation flags:
static

Template:
output_file(Path,Basename,Directory,Options)
Mode and number of proofs:
output_file(+atom,+atom,+atom,+list(compound)) - one

---

output_sub_diagrams/1

Outputs sub-diagrams using the specified options.

Compilation flags:
static

Template:

output_sub_diagrams(Options)
Mode and number of proofs:
output_sub_diagrams(+list(compound)) - one

---

reset/0

Resets all temporary information used when generating a diagram.

Compilation flags:
static

Mode and number of proofs:
reset - one

---

output_node/6

Outputs a graph node.

Compilation flags:
static

Template:
output_node(Identifier,Label,Caption,Contents,Kind,Options)
Mode and number of proofs:
output_node(+nonvar,+nonvar,+nonvar,+list(nonvar),+atom,+list(compound)) - one

---

node/6

Enumerates, by backtracking, all saved nodes.

Compilation flags:
static

Template:

node(Identifier,Label,Caption,Contents,Kind,Options)
Mode and number of proofs:
node(?nonvar,?nonvar,?nonvar,?list(compound),?atom,?list(compound)) - zero_or_more

---

edge/5

Enumerates, by backtracking, all saved edges.

Compilation flags:
static

Template:
edge(From,To,Labels,Kind,Options)
Mode and number of proofs:
edge(?nonvar,?nonvar,?list(nonvar),?atom,?list(compound)) - zero_or_more

---

output_edges/1

Outputs all edges.

Compilation flags:
static

Template:
output_edges(Options)
Mode and number of proofs:
output_edges(+list(compound)) - one

---

save_edge/5

Saves a graph edge.

Compilation flags:
    static

Template:
    save_edge(From,To,Labels,Kind,Options)
Mode and number of proofs:
    save_edge(+nonvar,+nonvar,+list(nonvar),+atom,+list(compound)) - one

---

output_missing_externals/1

Outputs missing external nodes (usually due to unloaded resources) that are referenced from edges.

Compilation flags:
    static

Template:
    output_missing_externals(Options)
Mode and number of proofs:
    output_missing_externals(+list(compound)) - one

---

not_excluded_file/4

True when the given file is not excluded from the generated output. Excluded files may be specified by full path or by basename and with or without extension. Excluded directories may be listed by full or relative path.

Compilation flags:
    static

Template:
    not_excluded_file(Path,Basename,ExcludedDirectories,ExcludedFiles)
Mode and number of proofs:
    not_excluded_file(+atom,+atom,+list(atom),+list(atom)) - zero_or_one

---

output_file_path/4

Returns the output file path.

Compilation flags:
    static

Template:
    output_file_path(Name,Options,Format,Path)
Mode and number of proofs:
    output_file_path(+atom,+list(atom),+object_identifier,-atom) - one

---

locate_library/2

Locates a library given its name.

Compilation flags:
    static

Template:
    locate_library(Library,Path)
Mode and number of proofs:
    locate_library(+atom,-atom) - one

---

locate_directory/2

Locates a directory given its name or full path.

Compilation flags:
    static

Template:
    locate_directory(Directory,Path)
Mode and number of proofs:

locate_directory(+atom,-atom) - one

---

locate_file/5

Locates a file given its name, basename, full path, or library notation representation.

Compilation flags:
static

Template:
locate_file(File,Basename,Extension,Directory,Path)
Mode and number of proofs:
locate_file(+atom,+atom,+atom,+atom,-atom) - one

---

ground_entity_identifier/3

Converts an entity identifier to a ground term.

Compilation flags:
static

Template:
ground_entity_identifier(Kind,Identifier,GroundIdentifier)
Mode and number of proofs:
ground_entity_identifier(+atom,+callable,-callable) - one

---

filter_file_extension/3

Filters the file name extension depending on the file_extensions/1 option.

Compilation flags:
static

Template:

filter_file_extension(Basename,Options,Name)
Mode and number of proofs:
filter_file_extension(+atom,+list(compound),-atom) - one

---

filter_external_file_extension/3

Filters the external file name extension depending on the file_extensions/1 option.

Compilation flags:
static

Template:
filter_external_file_extension(Path,Options,Name)
Mode and number of proofs:
filter_external_file_extension(+atom,+list(compound),-atom) - one

---

add_link_options/3

Adds url/1, urls/2, and tooltip/1 link options (for use by the graph language) based on the specified path to the list of options.

Compilation flags:
static

Template:
add_link_options(Path,Options,LinkingOptions)
Mode and number of proofs:
add_link_options(+atom,+list(compound),-list(compound)) - one

---

supported_editor_url_scheme_prefix/1

Table of prefixes for text editors that supports a URL scheme to open diagram links.

Compilation flags:
    static

Template:
    supported_editor_url_scheme_prefix(Prefix)
Mode and number of proofs:
    supported_editor_url_scheme_prefix(?atom) - zero_or_more

---

omit_path_prefix/3

Removes a prefix from a path, returning the relative path, when using the option omit_path_prefixes/1. Used mainly for constructing directory and file node identifiers and captions.

Compilation flags:
    static

Template:
    omit_path_prefix(Path,Options,Relative)
Mode and number of proofs:
    omit_path_prefix(+atom,+list(compound),-atom) - one

---

add_node_zoom_option/4

Adds node zoom options when using the zoom option.

Compilation flags:
    static

Template:
    add_node_zoom_option(Identifier,Suffix,Options,NodeOptions)
Mode and number of proofs:
    add_node_zoom_option(+atom,+atom,+list(compound),-list(compound)) - one

---

message_diagram_description/1

Diagram description for progress messages.

Compilation flags:
    static

Template:
    message_diagram_description(Description)
Mode and number of proofs:
    message_diagram_description(?atom) - one

Private predicates

node_/6

Table of saved nodes.

Compilation flags:
    dynamic

Template:
    node_(Identifier,Label,Caption,Contents,Kind,Options)
Mode and number of proofs:
    node_(?nonvar,?nonvar,?nonvar,?list(compound),?atom,?list(compound)) - zero_or_more

---

node_path_/2

Table of node paths.

Compilation flags:
    dynamic

Template:

node_path_(Node,Path)
Mode and number of proofs:
node_path_(?ground,?list(ground)) - zero_or_more

---

edge_/5

Table of saved edges.

Compilation flags:
dynamic

Template:
edge_(From,To,Labels,Kind,Options)
Mode and number of proofs:
edge_(?nonvar,?nonvar,?list(nonvar),?atom,?list(compound)) - zero_or_more

---

cycle_edge_/2

Table of edges that are part of a cycle.

Compilation flags:
dynamic

Template:
cycle_edge_(From,To)
Mode and number of proofs:
cycle_edge_(?nonvar,?nonvar) - zero_or_more

Operators

(none)

object

1.55.6 diagrams

Predicates for generating all supported diagrams for libraries, directories, and files in one step using the DOT format.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 2:1:0
Date: 2019-04-07

Compilation flags:
static, context_switching_calls

Extends:
public diagrams(dot)

Remarks:
(none)

Inherited public predicates:
all_files/0 all_files/1 all_libraries/0 all_libraries/1 directories/2 directories/3 directory/1 directory/2 directory/3 files/1 files/2 files/3 libraries/1 libraries/2 libraries/3 library/1 library/2 rdirectory/1 rdirectory/2 rdirectory/3 rlibrary/1 rlibrary/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.55.7 diagrams(Format)

- Format - Graph language file format.

Predicates for generating all supported diagrams for libraries, directories, or files in one step using the specified format.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 2:8:0
Date: 2019-06-13

Compilation flags:
static, context_switching_calls

Uses:
list
os

Remarks:

- Common options: title/1, date/1, output_directory/1, relation_labels/1, node_type_captions/1, exclude_files/1, exclude_libraries/1, url_prefixes/1, omit_path_prefix/1, entity_url_suffix_target/2, and layout/1.

- Limitations: Some of the provided predicates only make sense for some types of diagrams. Also, fine tuning may require generating individual diagrams directly instead of as a batch using this utility object.

Inherited public predicates:

(none)

- Public predicates
  - libraries/3
  - libraries/2
  - libraries/1
  - all_libraries/1
  - all_libraries/0
  - rlibrary/2
  - rlibrary/1
  - library/2
  - library/1
  - directories/3
  - directories/2
  - rdirectory/3
  - rdirectory/2
  - rdirectory/1
  - directory/3
  - directory/2
  - directory/1
  - files/3
  - files/2
  - files/1
  - all_files/1
  - all_files/0
- Protected predicates
- Private predicates
- Operators

Public predicates

libraries/3

Creates all supported diagrams for a set of libraries using the specified options. The Project argument is used as a prefix for the diagram file names.

Compilation flags:
    static

Template:
    libraries(Project,Libraries,Options)
Mode and number of proofs:
    libraries(+atom,+list(atom),+list(compound)) - one

---

libraries/2

Creates all supported diagrams for a set of libraries using the default options. The Project argument is used as a prefix for the diagram file names.

Compilation flags:
    static

Template:
    libraries(Project,Libraries)
Mode and number of proofs:
    libraries(+atom,+list(atom)) - one

---

libraries/1

Creates all supported diagrams for a set of libraries using the default options. The prefix libraries is used for the diagram file names.

Compilation flags:
    static

Template:

libraries(Libraries)
Mode and number of proofs:
libraries(+list(atom)) - one

---

all_libraries/1

Creates all supported diagrams for all loaded libraries using the specified options.

Compilation flags:
static

Template:
all_libraries(Options)
Mode and number of proofs:
all_libraries(+list(compound)) - one

---

all_libraries/0

Creates all supported diagrams for all loaded libraries using default options.

Compilation flags:
static

Mode and number of proofs:
all_libraries - one

---

rlibrary/2

Creates all supported diagrams for a library and its sub-libraries using the specified options.

Compilation flags:
static

Template:

rlibrary(Library,Options)
Mode and number of proofs:
rlibrary(+atom,+list(compound)) - one

---

rlibrary/1

Creates all supported diagrams for a library and its sub-libraries using default options.

Compilation flags:
static

Template:
rlibrary(Library)
Mode and number of proofs:
rlibrary(+atom) - one

---

library/2

Creates all supported diagrams for a library using the specified options.

Compilation flags:
static

Template:
library(Library,Options)
Mode and number of proofs:
library(+atom,+list(compound)) - one

---

library/1

Creates all supported diagrams for a library using default options.

Compilation flags:
    static

Template:
    library(Library)
Mode and number of proofs:
    library(+atom) - one

---

directories/3

Creates all supported diagrams for a set of directories using the specified options. The Project argument is used as a prefix for the diagram file names.

Compilation flags:
    static

Template:
    directories(Project,Directories,Options)
Mode and number of proofs:
    directories(+atom,+list(atom),+list(compound)) - one

---

directories/2

Creates all supported diagrams for a directory using default options. The Project argument is used as a prefix for the diagram file names.

Compilation flags:
    static

Template:
    directories(Project,Directories)
Mode and number of proofs:
    directories(+atom,+list(atom)) - one

---

rdirectory/3

Creates all supported diagrams for a directory and its sub-directories using the specified options. The Project argument is used as a prefix for the diagram file name.

Compilation flags:
    static

Template:
    rdirectory(Project,Directory,Options)
Mode and number of proofs:
    rdirectory(+atom,+atom,+list(compound)) - one

---

rdirectory/2

Creates all supported diagrams for a directory and its sub-directories using default options. The Project argument is used as a prefix for the diagram file name.

Compilation flags:
    static

Template:
    rdirectory(Project,Directory)
Mode and number of proofs:
    rdirectory(+atom,+atom) - one

---

rdirectory/1

Creates all supported diagrams for a directory and its sub-directories using default options. The name of the directory is used as a prefix for the diagram file name.

Compilation flags:
    static

Template:

rdirectory(Directory)
Mode and number of proofs:
rdirectory(+atom) - one

---

directory/3

Creates all supported diagrams for a directory using the specified options. The Project argument is used as a prefix for the diagram file names.

Compilation flags:
static

Template:
directory(Project,Directory,Options)
Mode and number of proofs:
directory(+atom,+atom,+list(compound)) - one

---

directory/2

Creates all supported diagrams for a directory using default options. The Project argument is used as a prefix for the diagram file names.

Compilation flags:
static

Template:
directory(Project,Directory)
Mode and number of proofs:
directory(+atom,+atom) - one

---

directory/1

Creates all supported diagrams for a directory using default options. The name of the directory is used as a prefix for the diagram file names.

Compilation flags:
    static

Template:
    directory(Directory)
Mode and number of proofs:
    directory(+atom) - one

---

files/3

Creates all supported diagrams for a set of files using the specified options. The file can be specified by name, basename, full path, or using library notation. The Project argument is used as a prefix for the diagram file names.

Compilation flags:
    static

Template:
    files(Project,Files,Options)
Mode and number of proofs:
    files(+atom,+list(atom),+list(compound)) - one

---

files/2

Creates all supported diagrams for a set of files using the default options. The file can be specified by name, basename, full path, or using library notation. The Project argument is used as a prefix for the diagram file names.

Compilation flags:
    static

Template:

files(Project,Files)
Mode and number of proofs:
files(+atom,+list(atom)) - one

---

files/1

Creates all supported diagrams for a set of files using the default options. The file can be specified by name, basename, full path, or using library notation. The prefix "files" is used for the diagram file names.

Compilation flags:
static

Template:
files(Files)
Mode and number of proofs:
files(+list(atom)) - one

---

all_files/1

Creates all supported diagrams for all loaded files using the specified options.

Compilation flags:
static

Template:
all_files(Options)
Mode and number of proofs:
all_files(+list(compound)) - one

---

all_files/0

Creates all supported diagrams for all loaded files using default options.

Compilation flags:
    static

Mode and number of proofs:
    all_files - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.55.8 directory_dependency_diagram

Predicates for generating directory dependency diagrams in DOT format.

Availability:
    logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2019-04-07

Compilation flags:
    static, context_switching_calls

Extends:
    public directory_dependency_diagram(dot)

Remarks:
(none)

Inherited public predicates:
all_files/0 all_files/1 all_libraries/0 all_libraries/1 check_option/1 check_options/1 default_option/1 default_options/1 diagram_description/1 diagram_name_suffix/1 directories/2 directories/3 directory/1 directory/2 directory/3 files/1 files/2 files/3 format_object/1 libraries/1 libraries/2 libraries/3 library/1 library/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory/3 rlibrary/1 rlibrary/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

directory_load_diagram, file_load_diagram

object

1.55.9 directory_dependency_diagram(Format)

- Format - Graph language file format.

Predicates for generating directory dependency diagrams. A dependency exists when an entity in one directory makes a reference to an entity in another directory.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 3:2:0
Date: 2026-03-13

Compilation flags:
static, context_switching_calls

Imports:
public directory_diagram(Format)
Uses:
file_dependency_diagram(Format)
list
logtalk
modules_diagram_support

Remarks:
(none)

Inherited public predicates:
all_files/0 all_files/1 all_libraries/0 all_libraries/1 check_option/1 check_options/1 default_option/1 default_options/1 diagram_description/1 diagram_name_suffix/1 directories/2 directories/3 directory/1 directory/2 directory/3 files/1 files/2 files/3 format_object/1 libraries/1 libraries/2 libraries/3 library/1 library/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory/3 rlibrary/1 rlibrary/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
  - sub_diagram_/2
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

sub_diagram_/2

Table of directory sub-diagrams to support their generation.

Compilation flags:
    dynamic

Template:
    sub_diagram_(Project,Directory)
Mode and number of proofs:
    sub_diagram_(?atom,?atom) - zero_or_more

Operators

(none)

See also

directory_load_diagram(Format), file_load_diagram(Format), library_load_diagram(Format)

category

1.55.10 directory_diagram(Format)

- Format - Graph language file format.

Common predicates for generating directory diagrams.

Availability:
    logtalk_load(diagrams(loader))

Author: Paulo Moura

Version: 1:13:0
Date: 2024-12-04

Compilation flags:
static

Extends:
public diagram(Format)
Uses:
list

Remarks:
(none)

Inherited public predicates:
all_files/0 all_files/1 all_libraries/0 all_libraries/1 check_option/1 check_options/1 default_option/1 default_options/1 diagram_description/1 diagram_name_suffix/1 directories/2 directories/3 directory/1 directory/2 directory/3 files/1 files/2 files/3 format_object/1 libraries/1 libraries/2 libraries/3 library/1 library/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory/3 rlibrary/1 rlibrary/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
  - remember_included_directory/1
  - remember_referenced_logtalk_directory/1
  - remember_referenced_prolog_directory/1
- Private predicates
  - included_directory_/1
  - referenced_logtalk_directory_/1
  - referenced_prolog_directory_/1
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

remember_included_directory/1

Remember included Logtalk directory in the diagram.

Compilation flags:
    static

Template:
    remember_included_directory(Path)
Mode and number of proofs:
    remember_included_directory(+atom) - one

---

remember_referenced_logtalk_directory/1

Remember referenced Logtalk directory in the diagram.

Compilation flags:
    static

Template:
    remember_referenced_logtalk_directory(Path)
Mode and number of proofs:
    remember_referenced_logtalk_directory(+atom) - one

---

remember_referenced_prolog_directory/1

Remember referenced Prolog directory in the diagram.

Compilation flags:
    static

Template:
remember_referenced_prolog_directory(Path)
Mode and number of proofs:
remember_referenced_prolog_directory(+atom) - one

Private predicates

included_directory_/1

Table of Logtalk directories already included in the diagram.

Compilation flags:
dynamic

Template:
included_directory_(Path)
Mode and number of proofs:
included_directory_(?atom) - zero_or_more

---

referenced_logtalk_directory_/1

Table of referenced Logtalk directories in the diagram.

Compilation flags:
dynamic

Template:
referenced_logtalk_directory_(Path)
Mode and number of proofs:
referenced_logtalk_directory_(?atom) - zero_or_more

---

referenced_prolog_directory_/1

Table of referenced Prolog directories in the diagram.

Compilation flags:
dynamic

Template:
referenced_prolog_directory_(Path)
Mode and number of proofs:
referenced_prolog_directory_(?atom) - zero_or_more

Operators

(none)

object

1.55.11 directory_load_diagram

Predicates for generating directory loading dependency diagrams in DOT format.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2019-04-07

Compilation flags:
static, context_switching_calls

Extends:
public directory_load_diagram(dot)

Remarks:
(none)

Inherited public predicates:

all_files/0 all_files/1 all_libraries/0 all_libraries/1 check_option/1 check_options/1 default_option/1 default_options/1 diagram_description/1 diagram_name_suffix/1 directories/2 directories/3 directory/1 directory/2 directory/3 files/1 files/2 files/3 format_object/1 libraries/1 libraries/2 libraries/3 library/1 library/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory/3 rlibrary/1 rlibrary/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

directory_dependency_diagram, file_dependency_diagram

object

1.55.12 directory_load_diagram(Format)

- Format - Graph language file format.

Predicates for generating directory loading dependency diagrams.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 3:0:1
Date: 2024-04-01

Compilation flags:
static, context_switching_calls

Imports:
public directory_diagram(Format)
Uses:
file_dependency_diagram(Format)
file_load_diagram(Format)
list
logtalk
modules_diagram_support

Remarks:
(none)

Inherited public predicates:
all_files/0 all_files/1 all_libraries/0 all_libraries/1 check_option/1 check_options/1 default_option/1 default_options/1 diagram_description/1 diagram_name_suffix/1 directories/2 directories/3 directory/1 directory/2 directory/3 files/1 files/2 files/3 format_object/1 libraries/1 libraries/2 libraries/3 library/1 library/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory/3 rlibrary/1 rlibrary/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
  - sub_diagram_/2
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

sub_diagram_/2

Table of directory sub-diagrams to support their generation.

Compilation flags:
dynamic

Template:
sub_diagram_(Project,Directory)
Mode and number of proofs:
sub_diagram_(?atom,?atom) - zero_or_more

Operators

(none)

See also

directory_dependency_diagram(Format), file_dependency_diagram(Format), library_dependency_diagram(Format)

object

1.55.13 dot_graph_language

Predicates for generating graph files in the DOT language (version 2.36.0 or later).

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 3:14:0
Date: 2026-03-14

Compilation flags:

static, context_switching_calls

Implements:
public graph_language_protocol
Imports:
public options
Provides:
graph_language_registry::language_object/2
Uses:
list
os
term_io
user

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 default_option/1 default_options/1 edge/6 file_footer/3 file_header/3 graph_footer/5 graph_header/5 node/7 option/2 option/3 output_file_name/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.55.14 entity_diagram

Predicates for generating entity diagrams in DOT format with both inheritance and cross-referencing relation edges.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 2:1:0
Date: 2026-03-12

Compilation flags:
static, context_switching_calls

Extends:
public entity_diagram(dot)

Remarks:
(none)

Inherited public predicates:
all_files/0 all_files/1 all_libraries/0 all_libraries/1 check_option/1 check_options/1 default_option/1 default_options/1 diagram_description/1 diagram_name_suffix/1 directories/2 directories/3 directory/1 directory/2 directory/3 file/1 file/2 files/1 files/2 files/3 format_object/1 libraries/1 libraries/2 libraries/3 library/1 library/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory/3 rlibrary/1 rlibrary/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates

- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

inheritance_diagram, uses_diagram, caller_diagram, xref_diagram

object

1.55.15 entity_diagram(Format)

- Format - Graph language file format.

Predicates for generating entity diagrams in the specified format with both inheritance and cross-referencing relation edges.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 2:63:0
Date: 2026-03-14

Compilation flags:
static, context_switching_calls

Imports:
public diagram(Format)

Uses:
coupling_metric
list
logtalk
modules_diagram_support
user

Remarks:
(none)

Inherited public predicates:
all_files/0 all_files/1 all_libraries/0 all_libraries/1 check_option/1 check_options/1 default_option/1 default_options/1 diagram_description/1 diagram_name_suffix/1 directories/2 directories/3 directory/1 directory/2 directory/3 files/1 files/2 files/3 format_object/1 libraries/1 libraries/2 libraries/3 library/1 library/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory/3 rlibrary/1 rlibrary/2 valid_option/1 valid_options/1

- Public predicates
  - file/2
  - file/1
- Protected predicates
- Private predicates
  - included_entity_/1
  - included_module_/1
  - referenced_entity_/2
  - referenced_module_/2
- Operators

Public predicates

file/2

Creates a diagram for all entities in a loaded source file using the specified options. The file can be specified by name, basename, full path, or using library notation.

Compilation flags:
static

Template:

file(File,Options)
Mode and number of proofs:
file(+atom,+list(compound)) - one

---

file/1

Creates a diagram for all entities in a loaded source file using default options. The file can be specified by name, basename, full path, or using library notation.

Compilation flags:
static

Template:
file(File)
Mode and number of proofs:
file(+atom) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

included_entity_/1

Table of Logtalk entities already included in the diagram.

Compilation flags:
dynamic

Template:
included_entity_(Entity)
Mode and number of proofs:
included_entity_(?entity_identifier) - zero_or_more

---

included_module_/1

Table of Prolog modules already included in the diagram.

Compilation flags:
    dynamic

Template:
    included_module_(Module)
Mode and number of proofs:
    included_module_(?module_identifier) - zero_or_more

---

referenced_entity_/2

Table of referenced Logtalk entities in the diagram.

Compilation flags:
    dynamic

Template:
    referenced_entity_(Referencer,Entity)
Mode and number of proofs:
    referenced_entity_(?entity_identifier,?entity_identifier) - zero_or_more

---

referenced_module_/2

Table of referenced Logtalk entities in the diagram.

Compilation flags:
    dynamic

Template:
    referenced_module_(Referencer,Entity)
Mode and number of proofs:
    referenced_module_(?entity_identifier,?module_identifier) - zero_or_more

---

Operators

(none)

See also

inheritance_diagram(Format), uses_diagram(Format), caller_diagram(Format), xref_diagram(Format), library_diagram(Format)

object

1.55.16 file_dependency_diagram

Predicates for generating file contents dependency diagrams in DOT format. A dependency exists when an entity in one file makes a reference to an entity in another file.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 2:1:0
Date: 2019-06-13

Compilation flags:
static, context_switching_calls

Extends:
public file_dependency_diagram(dot)

Remarks:
(none)

Inherited public predicates:
all_files/0 all_files/1 all_libraries/0 all_libraries/1 check_option/1 check_options/1 default_option/1 default_options/1 diagram_description/1 diagram_name_suffix/1 directories/2 directories/3 directory/1 directory/2 directory/3 files/1 files/2 files/3 format_object/1 libraries/1 libraries/2 libraries/3 library/1 library/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory/3 rlibrary/1 rlibrary/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates

- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

file_load_diagram, directory_load_diagram, library_load_diagram

object

1.55.17 file_dependency_diagram(Format)

- Format - Graph language file format.

Predicates for generating file contents dependency diagrams. A dependency exists when an entity in one file makes a reference to an entity in another file.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 2:28:3
Date: 2024-04-01

Compilation flags:
static, context_switching_calls

Imports:

public file_diagram(Format)

Uses:

entity_diagram(Format)
list
logtalk
modules_diagram_support
os

Remarks:

(none)

Inherited public predicates:

all_files/0 all_files/1 all_libraries/0 all_libraries/1 check_option/1 check_options/1 default_option/1 default_options/1 diagram_description/1 diagram_name_suffix/1 directories/2 directories/3 directory/1 directory/2 directory/3 files/1 files/2 files/3 format_object/1 libraries/1 libraries/2 libraries/3 library/1 library/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory/3 rlibrary/1 rlibrary/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
  - sub_diagram_/1
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

sub_diagram_/1

Table of file sub-diagrams to support their generation.

Compilation flags:
dynamic

Template:
sub_diagram_(File)
Mode and number of proofs:
sub_diagram_(?atom) - zero_or_more

Operators

(none)

See also

file_load_diagram(Format), directory_load_diagram(Format), library_load_diagram(Format)

category

1.55.18 file_diagram(Format)

- Format - Graph language file format.

Common predicates for generating file diagrams.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 2:14:0
Date: 2024-12-04

Compilation flags:
static

Extends:
public diagram(Format)
Uses:
list

Remarks:
(none)

Inherited public predicates:
all_files/0 all_files/1 all_libraries/0 all_libraries/1 check_option/1 check_options/1 default_option/1 default_options/1 diagram_description/1 diagram_name_suffix/1 directories/2 directories/3 directory/1 directory/2 directory/3 files/1 files/2 files/3 format_object/1 libraries/1 libraries/2 libraries/3 library/1 library/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory/3 rlibrary/1 rlibrary/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
  - remember_included_file/1
  - remember_referenced_logtalk_file/1
  - remember_referenced_prolog_file/1
- Private predicates
  - included_file_/1
  - referenced_logtalk_file_/1
  - referenced_prolog_file_/1
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

remember_included_file/1

Remember included Logtalk file in the diagram.

Compilation flags:
    static

Template:
    remember_included_file(Path)
Mode and number of proofs:
    remember_included_file(+atom) - one

---

remember_referenced_logtalk_file/1

Remember referenced Logtalk file in the diagram.

Compilation flags:
    static

Template:
    remember_referenced_logtalk_file(Path)
Mode and number of proofs:
    remember_referenced_logtalk_file(+atom) - one

---

remember_referenced_prolog_file/1

Remember referenced Prolog file in the diagram.

Compilation flags:
    static

Template:
remember_referenced_prolog_file(Path)
Mode and number of proofs:
remember_referenced_prolog_file(+atom) - one

Private predicates

included_file_/1

Table of Logtalk files already included in the diagram.

Compilation flags:
dynamic

Template:
included_file_(Path)
Mode and number of proofs:
included_file_(?atom) - zero_or_more

---

referenced_logtalk_file_/1

Table of referenced Logtalk files in the diagram.

Compilation flags:
dynamic

Template:
referenced_logtalk_file_(Path)
Mode and number of proofs:
referenced_logtalk_file_(?atom) - zero_or_more

---

referenced_prolog_file_/1

Table of referenced Prolog files in the diagram.

Compilation flags:
dynamic

Template:
referenced_prolog_file_(Path)
Mode and number of proofs:
referenced_prolog_file_(?atom) - zero_or_more

Operators

(none)

object

1.55.19 file_load_diagram

Predicates for generating file loading dependency diagrams in DOT format. A dependency exists when a file loads or includes another file.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 2:1:0
Date: 2019-06-13

Compilation flags:
static, context_switching_calls

Extends:
public file_load_diagram(dot)

Remarks:
(none)

Inherited public predicates:

all_files/0 all_files/1 all_libraries/0 all_libraries/1 check_option/1 check_options/1 default_option/1 default_options/1 diagram_description/1 diagram_name_suffix/1 directories/2 directories/3 directory/1 directory/2 directory/3 files/1 files/2 files/3 format_object/1 libraries/1 libraries/2 libraries/3 library/1 library/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory/3 rlibrary/1 rlibrary/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

file_dependency_diagram, directory_dependency_diagram, library_dependency_diagram

object

1.55.20 file_load_diagram(Format)

- Format - Graph language file format.

Predicates for generating file loading dependency diagrams. A dependency exists when a file loads or includes another file.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 2:30:3
Date: 2024-12-05

Compilation flags:
static, context_switching_calls

Imports:
public file_diagram(Format)
Uses:
entity_diagram(Format)
list
logtalk
modules_diagram_support
os

Remarks:
(none)

Inherited public predicates:
all_files/0 all_files/1 all_libraries/0 all_libraries/1 check_option/1 check_options/1 default_option/1 default_options/1 diagram_description/1 diagram_name_suffix/1 directories/2 directories/3 directory/1 directory/2 directory/3 files/1 files/2 files/3 format_object/1 libraries/1 libraries/2 libraries/3 library/1 library/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory/3 rlibrary/1 rlibrary/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
  - sub_diagram_/1
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

sub_diagram_/1

Table of file sub-diagrams to support their generation.

Compilation flags:
dynamic

Template:
sub_diagram_(File)
Mode and number of proofs:
sub_diagram_(?atom) - zero_or_more

Operators

(none)

See also

file_dependency_diagram(Format), directory_dependency_diagram(Format), library_dependency_diagram(Format)

protocol

1.55.21 graph_language_protocol

Predicates for generating graph files.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 2:0:0
Date: 2014-12-30

Compilation flags:

static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - output_file_name/2
  - file_header/3
  - file_footer/3
  - graph_header/5
  - graph_footer/5
  - node/7
  - edge/6
- Protected predicates
- Private predicates
- Operators

Public predicates

output_file_name/2

Constructs the diagram file basename by adding a graph language dependent extension to the given name.

Compilation flags:
static

Template:
output_file_name(Name,Basename)
Mode and number of proofs:
output_file_name(+atom,-atom) - one

file_header/3

Writes the output file header using the specified options.

Compilation flags:
    static

Template:
    file_header(Stream,Identifier,Options)
Mode and number of proofs:
    file_header(+stream_or_alias,+atom,+list(compound)) - one

---

file_footer/3

Writes the output file footer using the specified options.

Compilation flags:
    static

Template:
    file_footer(Stream,Identifier,Options)
Mode and number of proofs:
    file_footer(+stream_or_alias,+atom,+list(compound)) - one

---

graph_header/5

Writes a graph header using the specified options.

Compilation flags:
    static

Template:
    graph_header(Stream,Identifier,Label,Kind,Options)
Mode and number of proofs:
    graph_header(+stream_or_alias,+atom,+atom,+atom,+list(compound)) - one

---

graph_footer/5

Writes a graph footer using the specified options.

Compilation flags:
    static

Template:
    graph_footer(Stream,Identifier,Label,Kind,Options)
Mode and number of proofs:
    graph_footer(+stream_or_alias,+atom,+atom,+atom,+list(compound)) - one

---

node/7

Writes a node using the specified options.

Compilation flags:
    static

Template:
    node(Stream,Identifier,Label,Caption,Lines,Kind,Options)
Mode and number of proofs:
    node(+stream_or_alias,+nonvar,+nonvar,+nonvar,+list(nonvar),+atom,+list(compound)) - one

---

edge/6

Writes an edge between two nodes using the specified options.

Compilation flags:
    static

Template:
    edge(Stream,Start,End,Labels,Kind,Options)
Mode and number of proofs:
    edge(+stream_or_alias,+nonvar,+nonvar,+list(nonvar),+atom,+list(compound)) - one

---

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

object

1.55.22 graph_language_registry

Registry of implemented graph languages.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 1:0:1
Date: 2020-03-25

Compilation flags:
static, context_switching_calls

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - language_object/2
- Protected predicates

- Private predicates
- Operators

Public predicates

language_object/2

Table of defined graph languages and their implementation objects.

Compilation flags:
    static, multifile

Template:
    language_object(Language,Object)
Mode and number of proofs:
    language_object(?atom,?object_identifier) - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

object

1.55.23 inheritance_diagram

Predicates for generating entity diagrams in DOT format with inheritance relation edges but no cross-referencing relation edges.

Availability:
    logtalk_load(diagrams(loader))

Author: Paulo Moura

Version: 2:0:0
Date: 2014-01-15

Compilation flags:
static, context_switching_calls

Extends:
public inheritance_diagram(dot)

Remarks:
(none)

Inherited public predicates:
all_files/0 all_files/1 all_libraries/0 all_libraries/1 check_option/1 check_options/1 default_option/1 default_options/1 diagram_description/1 diagram_name_suffix/1 directories/2 directories/3 directory/1 directory/2 directory/3 file/1 file/2 files/1 files/2 files/3 format_object/1 libraries/1 libraries/2 libraries/3 library/1 library/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory/3 rlibrary/1 rlibrary/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

entity_diagram, uses_diagram, xref_diagram

object

1.55.24 inheritance_diagram(Format)

- Format - Graph language file format.

Predicates for generating entity diagrams in the specified format with inheritance relation edges but no cross-referencing relation edges.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 2:20:0
Date: 2024-03-20

Compilation flags:
static, context_switching_calls

Extends:
public entity_diagram(Format)
Uses:
logtalk

Remarks:
(none)

Inherited public predicates:
all_files/0 all_files/1 all_libraries/0 all_libraries/1 check_option/1 check_options/1 default_option/1 default_options/1 diagram_description/1 diagram_name_suffix/1 directories/2 directories/3 directory/1 directory/2 directory/3 file/1 file/2 files/1 files/2 files/3 format_object/1 libraries/1 libraries/2 libraries/3 library/1 library/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory/3 rlibrary/1 rlibrary/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

entity_diagram(Format), uses_diagram(Format), xref_diagram(Format)

object

1.55.25 library_dependency_diagram

Predicates for generating library dependency diagrams in DOT format.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 2:1:0
Date: 2019-06-13

Compilation flags:
static, context_switching_calls

Extends:
public library_dependency_diagram(dot)

Remarks:
(none)

Inherited public predicates:
all_files/0 all_files/1 all_libraries/0 all_libraries/1 check_option/1 check_options/1 default_option/1 default_options/1 diagram_description/1 diagram_name_suffix/1 directories/2 directories/3 directory/1 directory/2 directory/3 files/1 files/2 files/3 format_object/1 libraries/1 libraries/2 libraries/3 library/1 library/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory/3 rlibrary/1 rlibrary/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

library_load_diagram, file_load_diagram, entity_diagram

object

1.55.26 library_dependency_diagram(Format)

- Format - Graph language file format.

Predicates for generating library dependency diagrams. A dependency exists when an entity in one library makes a reference to an entity in another library.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 2:35:0
Date: 2026-03-13

Compilation flags:
static, context_switching_calls

Imports:
public library_diagram(Format)
Uses:
entity_diagram(Format)
list
logtalk
modules_diagram_support

Remarks:
(none)

Inherited public predicates:
all_files/0 all_files/1 all_libraries/0 all_libraries/1 check_option/1 check_options/1 default_option/1 default_options/1 diagram_description/1 diagram_name_suffix/1 directories/2 directories/3 directory/1 directory/2 directory/3 files/1 files/2 files/3 format_object/1 libraries/1 libraries/2 libraries/3 library/1 library/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory/3 rlibrary/1 rlibrary/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
  - sub_diagram_/1
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

sub_diagram_/1

Table of library sub-diagrams to support their generation.

Compilation flags:
    dynamic

Template:
    sub_diagram_(Library)
Mode and number of proofs:
    sub_diagram_(?atom) - zero_or_more

Operators

(none)

See also

library_load_diagram(Format), directory_load_diagram(Format), file_load_diagram(Format), entity_diagram(Format)

category

1.55.27 library_diagram(Format)

- Format - Graph language file format.

Common predicates for generating library diagrams.

Availability:
    logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 2:17:0
Date: 2024-12-04

Compilation flags:
static

Extends:
public diagram(Format)
Uses:
list
user

Remarks:
(none)

Inherited public predicates:
all_files/0 all_files/1 all_libraries/0 all_libraries/1 check_option/1 check_options/1 default_option/1 default_options/1 diagram_description/1 diagram_name_suffix/1 directories/2 directories/3 directory/1 directory/2 directory/3 files/1 files/2 files/3 format_object/1 libraries/1 libraries/2 libraries/3 library/1 library/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory/3 rlibrary/1 rlibrary/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
  - add_library_documentation_url/4
  - remember_included_library/2
  - remember_referenced_logtalk_library/2
  - remember_referenced_prolog_library/2
- Private predicates
  - included_library_/2
  - referenced_logtalk_library_/2
  - referenced_prolog_library_/2
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

add_library_documentation_url/4

Adds a documentation URL when using the option url_prefixes/2.

Compilation flags:
    static

Template:
    add_library_documentation_url(Kind,Options,Library,NodeOptions)
Mode and number of proofs:
    add_library_documentation_url(+atom,+list(compound),+atom,-list(compound)) - one

---

remember_included_library/2

Remember included Logtalk library in the diagram.

Compilation flags:
    static

Template:
    remember_included_library(Library,Path)
Mode and number of proofs:
    remember_included_library(+atom,+atom) - one

---

remember_referenced_logtalk_library/2

Remember referenced Logtalk library in the diagram.

Compilation flags:
    static

Template:
remember_referenced_logtalk_library(Library,Path)
Mode and number of proofs:
remember_referenced_logtalk_library(+atom,+atom) - one

---

remember_referenced_prolog_library/2

Remember referenced Prolog library in the diagram.

Compilation flags:
static

Template:
remember_referenced_prolog_library(Library,Path)
Mode and number of proofs:
remember_referenced_prolog_library(+atom,+atom) - one

Private predicates

included_library_/2

Table of Logtalk libraries already included in the diagram.

Compilation flags:
dynamic

Template:
included_library_(Library,Path)
Mode and number of proofs:
included_library_(?atom,?atom) - zero_or_more

---

referenced_logtalk_library_/2

Table of referenced Logtalk libraries in the diagram.

Compilation flags:
    dynamic

Template:
    referenced_logtalk_library_(Library,Path)
Mode and number of proofs:
    referenced_logtalk_library_(?atom,?atom) - zero_or_more

---

referenced_prolog_library_/2

Table of referenced Prolog libraries in the diagram.

Compilation flags:
    dynamic

Template:
    referenced_prolog_library_(Library,Path)
Mode and number of proofs:
    referenced_prolog_library_(?atom,?atom) - zero_or_more

Operators

(none)

See also

inheritance_diagram(Format), uses_diagram(Format), xref_diagram(Format), entity_diagram(Format)

object

1.55.28 library_load_diagram

Predicates for generating library loading dependency diagrams in DOT format.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 2:1:0
Date: 2019-06-13

Compilation flags:
static, context_switching_calls

Extends:
public library_load_diagram(dot)

Remarks:
(none)

Inherited public predicates:
all_files/0 all_files/1 all_libraries/0 all_libraries/1 check_option/1 check_options/1 default_option/1 default_options/1 diagram_description/1 diagram_name_suffix/1 directories/2 directories/3 directory/1 directory/2 directory/3 files/1 files/2 files/3 format_object/1 libraries/1 libraries/2 libraries/3 library/1 library/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory/3 rlibrary/1 rlibrary/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

library_dependency_diagram, file_dependency_diagram, entity_diagram

object

1.55.29 library_load_diagram(Format)

- Format - Graph language file format.

Predicates for generating library loading dependency diagrams.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 2:33:1
Date: 2024-04-01

Compilation flags:
static, context_switching_calls

Imports:
public library_diagram(Format)
Uses:
entity_diagram(Format)
list
logtalk
modules_diagram_support

Remarks:
(none)

Inherited public predicates:
all_files/0 all_files/1 all_libraries/0 all_libraries/1 check_option/1 check_options/1 default_option/1 default_options/1 diagram_description/1 diagram_name_suffix/1 directories/2 directories/3 directory/1 directory/2 directory/3 files/1 files/2 files/3 format_object/1 libraries/1 libraries/2 libraries/3 library/1 library/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory/3 rlibrary/1 rlibrary/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
  - sub_diagram_/1
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

sub_diagram_/1

Table of library sub-diagrams to support their generation.

Compilation flags:
dynamic

Template:
sub_diagram_(Library)
Mode and number of proofs:
sub_diagram_(?atom) - zero_or_more

Operators

(none)

See also

library_dependency_diagram(Format), directory_dependency_diagram(Format), file_dependency_diagram(Format), entity_diagram(Format)

object

1.55.30 mermaid_graph_language

Predicates for generating graph files using Mermaid.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 0:5:0
Date: 2026-03-14

Compilation flags:
static, context_switching_calls

Implements:
public graph_language_protocol
Imports:
public options
Provides:
graph_language_registry::language_object/2
Uses:
list
os
term_io
user

Remarks:
(none)

Inherited public predicates:

check_option/1 check_options/1 default_option/1 default_options/1 edge/6 file_footer/3 file_header/3 graph_footer/5 graph_header/5 node/7 option/2 option/3 output_file_name/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
  - edge_index_/1
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

edge_index_/1

Current edge index counter.

Compilation flags:
dynamic

Template:
edge_index_(Index)
Mode and number of proofs:
edge_index_(?integer) - zero_or_one

Operators

(none)

object

1.55.31 modules_diagram_support

Utility predicates for supporting Prolog modules in diagrams.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 0:19:5
Date: 2022-07-08

Compilation flags:
static, context_switching_calls

Dependencies:
(none)

Remarks:

- Supported backend Prolog systems: ECLiPSe, SICStus Prolog, SWI-Prolog, and YAP.

Inherited public predicates:
(none)

- Public predicates
  - module_property/2
  - loaded_file_property/2
  - source_file_extension/1
- Protected predicates
- Private predicates
- Operators

Public predicates

module_property/2

Access to module properties, at least exports/1, file/1, and file/2 but also declares/2, defines/2, calls/2, and provides/3 when possible.

Compilation flags:
    static

Template:
    module_property(Module,Property)
Mode and number of proofs:
    module_property(?atom,?callable) - zero_or_more

---

loaded_file_property/2

Access to loaded source file properties, at least basename/1, directory/1 but also parent/1 when possible.

Compilation flags:
    static

Template:
    loaded_file_property(File,Property)
Mode and number of proofs:
    loaded_file_property(?atom,?callable) - zero_or_more

---

source_file_extension/1

Valid source file extension for Prolog source files.

Compilation flags:
    static

Template:
    source_file_extension(Extension)
Mode and number of proofs:

source_file_extension(?atom) - one_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

object

1.55.32 plantuml_graph_language

Predicates for generating diagrams in PlantUML format.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-03-14

Compilation flags:
static, context_switching_calls

Implements:
public graph_language_protocol
Imports:
public options
Provides:
graph_language_registry::language_object/2
Uses:
list
os
term_io
user

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 default_option/1 default_options/1 edge/6 file_footer/3 file_header/3 graph_footer/5 graph_header/5 node/7 option/2 option/3 output_file_name/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.55.33 uses_diagram

Predicates for generating entity diagrams in DOT format with only uses/2 and use_module/2 relation edges.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 2:0:1
Date: 2020-03-27

Compilation flags:
static, context_switching_calls

Extends:
public uses_diagram(dot)

Remarks:
(none)

Inherited public predicates:
all_files/0 all_files/1 all_libraries/0 all_libraries/1 check_option/1 check_options/1 default_option/1 default_options/1 diagram_description/1 diagram_name_suffix/1 directories/2 directories/3 directory/1 directory/2 directory/3 file/1 file/2 files/1 files/2 files/3 format_object/1 libraries/1 libraries/2 libraries/3 library/1 library/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory/3 rlibrary/1 rlibrary/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

entity_diagram, inheritance_diagram, xref_diagram

object

1.55.34 uses_diagram(Format)

- Format - Graph language file format.

Predicates for generating entity diagrams with only uses/2 and use_module/2 relation edges.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 2:21:0
Date: 2024-03-20

Compilation flags:
static, context_switching_calls

Extends:
public entity_diagram(Format)
Uses:
logtalk

Remarks:
(none)

Inherited public predicates:
all_files/0 all_files/1 all_libraries/0 all_libraries/1 check_option/1 check_options/1 default_option/1 default_options/1 diagram_description/1 diagram_name_suffix/1 directories/2 directories/3 directory/1 directory/2 directory/3 file/1 file/2 files/1 files/2 files/3 format_object/1 libraries/1 libraries/2 libraries/3 library/1 library/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory/3 rlibrary/1 rlibrary/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

entity_diagram(Format), inheritance_diagram(Format), xref_diagram(Format)

object

1.55.35 xref_diagram

Predicates for generating predicate call cross-referencing diagrams in DOT format.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 2:1:0
Date: 2026-03-12

Compilation flags:
static, context_switching_calls

Extends:
public xref_diagram(dot)

Remarks:
(none)

Inherited public predicates:
all_files/0 all_files/1 all_libraries/0 all_libraries/1 check_option/1 check_options/1 default_option/1 default_options/1 diagram_description/1 diagram_name_suffix/1 directories/2 directories/3 directory/1 directory/2 directory/3 entity/1 entity/2 file/1 file/2 files/1 files/2 files/3 format_object/1 libraries/1 libraries/2 libraries/3 library/1 library/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory/3 rlibrary/1 rlibrary/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

caller_diagram, entity_diagram, inheritance_diagram, uses_diagram

object

1.55.36 xref_diagram(Format)

- Format - Graph language file format.

Predicates for generating predicate call cross-referencing diagrams.

Availability:
logtalk_load(diagrams(loader))

Author: Paulo Moura
Version: 2:86:0
Date: 2026-03-12

Compilation flags:
static, context_switching_calls

Extends:
public entity_diagram(Format)
Uses:
atom
list
logtalk
modules_diagram_support
os
user

Remarks:
(none)

Inherited public predicates:
all_files/0 all_files/1 all_libraries/0 all_libraries/1 check_option/1 check_options/1 default_option/1 default_options/1 diagram_description/1 diagram_name_suffix/1 directories/2 directories/3 directory/1 directory/2 directory/3 file/1 file/2 files/1 files/2 files/3 format_object/1 libraries/1 libraries/2 libraries/3 library/1 library/2 option/2 option/3 rdirectory/1 rdirectory/2 rdirectory/3 rlibrary/1 rlibrary/2 valid_option/1 valid_options/1

- Public predicates
  - entity/2
  - entity/1
- Protected predicates
- Private predicates

    - included_predicate_/1
    - referenced_predicate_/1
    - external_predicate_/1
- Operators

Public predicates

entity/2

Creates a diagram for a single entity using the specified options.

Compilation flags:
    static

Template:
    entity(Entity,Options)
Mode and number of proofs:
    entity(+entity_identifier,+list(compound)) - one

---

entity/1

Creates a diagram for a single entity using default options.

Compilation flags:
    static

Template:
    entity(Entity)
Mode and number of proofs:
    entity(+entity_identifier) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

included_predicate_/1

Table of predicates already included in the diagram for the entity under processing.

Compilation flags:
    dynamic

Template:
    included_predicate_(Predicate)
Mode and number of proofs:
    included_predicate_(?predicate_indicator) - zero_or_more

---

referenced_predicate_/1

Table of referenced predicates for the entity under processing.

Compilation flags:
    dynamic

Template:
    referenced_predicate_(Predicate)
Mode and number of proofs:
    referenced_predicate_(?predicate_indicator) - zero_or_more

---

external_predicate_/1

Table of external predicate references for all the entities under processing.

Compilation flags:
    dynamic

Template:
external_predicate_(Reference)
Mode and number of proofs:
external_predicate_(?compound) - zero_or_more

Operators

(none)

See also

caller_diagram(Format), entity_diagram(Format), inheritance_diagram(Format), uses_diagram(Format)

1.56 dictionaries

object

1.56.1 avltree

AVL tree implementation of the dictionary protocol. Uses standard order to compare keys.

Availability:
logtalk_load(dictionaries(loader))

Author: R.A.O'Keefe, L.Damas, V.S.Costa, Glenn Burgess, Jiri Spitz, and Jan Wielemaker; Logtalk port and additional predicates by Paulo Moura
Version: 1:6:0
Date: 2026-02-10

Compilation flags:
static, context_switching_calls

Implements:
public dictionaryp
Extends:
public term
Uses:
list

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 apply/4 as_curly_bracketed/2 as_dictionary/2 as_list/2 check/1 clone/3 clone/4 delete/4 delete_max/4 delete_min/4 depth/2 empty/1 ground/1 insert/4 intersection/2 intersection/3 keys/2 lookup/2 lookup/3 lookup/4 map/2 map/3 max/3 min/3 new/1 next/4 numbervars/1 numbervars/3 occurs/2 previous/4 singletons/2 size/2 subsumes/2 subterm/2 update/3 update/4 update/5 valid/1 values/2 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

bintree, rbtree

object

1.56.2 bintree

Simple binary tree implementation of the dictionary protocol. Uses standard order to compare keys.

Availability:
logtalk_load(dictionaries(loader))

Author: Paulo Moura and Paul Fodor
Version: 2:13:0
Date: 2026-02-10

Compilation flags:
static, context_switching_calls

Implements:
public dictionaryp
Extends:
public term
Uses:
list

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 apply/4 as_curly_bracketed/2 as_dictionary/2 as_list/2 check/1 clone/3 clone/4 delete/4 delete_max/4 delete_min/4 depth/2 empty/1 ground/1 insert/4 intersection/2 intersection/3 keys/2 lookup/2 lookup/3 lookup/4 map/2 map/3 max/3 min/3 new/1 next/4 numbervars/1 numbervars/3 occurs/2 previous/4 singletons/2 size/2 subsumes/2 subterm/2 update/3 update/4 update/5 valid/1 values/2 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
  - preorder/2
  - inorder/2
  - postorder/2
- Protected predicates
- Private predicates
- Operators

Public predicates

preorder/2

Preorder tree traversal.

Compilation flags:
    static

Template:
    preorder(Tree,List)
Mode and number of proofs:
    preorder(@tree,-list) - one

---

inorder/2

Inorder tree traversal.

Compilation flags:
    static

Template:
    inorder(Tree,List)
Mode and number of proofs:
    inorder(@tree,-list) - one

---

postorder/2

Postorder tree traversal.

Compilation flags:
    static

Template:
    postorder(Tree,List)
Mode and number of proofs:
    postorder(@tree,-list) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

avltree, rbtree, splaytree

protocol

1.56.3 dictionaryp

Dictionary protocol.

Availability:
logtalk_load(dictionaries(loader))

Author: Paulo Moura
Version: 2:6:0
Date: 2026-02-10

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - as_dictionary/2
  - as_list/2
  - as_curly_bracketed/2
  - clone/3
  - clone/4
  - insert/4
  - delete/4
  - update/4
  - update/5
  - update/3
  - empty/1
  - lookup/4
  - lookup/3
  - lookup/2
  - intersection/2
  - intersection/3
  - previous/4
  - next/4
  - min/3
  - max/3
  - delete_min/4
  - delete_max/4
  - keys/2
  - values/2
  - map/2
  - map/3
  - apply/4
  - size/2
- Protected predicates
- Private predicates
- Operators

Public predicates

as_dictionary/2

Converts a list of key-value pairs to a dictionary.

Compilation flags:
    static

Template:
    as_dictionary(Pairs,Dictionary)
Mode and number of proofs:
    as_dictionary(@list(pairs),-dictionary) - one

---

as_list/2

Converts a dictionary to an ordered list (as per standard order) of key-value pairs.

Compilation flags:
    static

Template:
    as_list(Dictionary,Pairs)
Mode and number of proofs:
    as_list(@dictionary,-list(pairs)) - one

---

as_curly_bracketed/2

Creates a curly-bracketed term representation of a dictionary.

Compilation flags:
    static

Template:
    as_curly_bracketed(Dictionary,Term)
Mode and number of proofs:
    as_curly_bracketed(+dictionary,--term) - one

---

clone/3

Clones a dictionary using the same keys but with all values unbound and returning a list of all the pairs in the new clone.

Compilation flags:
    static

Template:
    clone(Dictionary,Clone,ClonePairs)
Mode and number of proofs:
    clone(+dictionary,-dictionary,-list(pairs)) - one

---

clone/4

Clones a dictionary using the same keys but with all values unbound and returning the list of all pairs in the dictionary and in the clone.

Compilation flags:
    static

Template:
    clone(Dictionary,Pairs,Clone,ClonePairs)
Mode and number of proofs:
    clone(+dictionary,-list(pairs),-dictionary,-list(pairs)) - one

---

insert/4

Inserts a key-value pair into a dictionary, returning the updated dictionary. When the key already exists, the associated value is updated.

Compilation flags:
    static

Template:

insert(OldDictionary,Key,Value,NewDictionary)
Mode and number of proofs:
insert(+dictionary,+ground,@term,-dictionary) - one

---

delete/4

Deletes a matching key-value pair from a dictionary, returning the updated dictionary. Fails if it cannot find the key or if the key exists but the value does not unify.

Compilation flags:
static

Template:
delete(OldDictionary,Key,Value,NewDictionary)
Mode and number of proofs:
delete(+dictionary,@ground,?term,-dictionary) - zero_or_one

---

update/4

Updates the value associated with Key in a dictionary, returning the updated dictionary. Fails if it cannot find the key.

Compilation flags:
static

Template:
update(OldDictionary,Key,NewValue,NewDictionary)
Mode and number of proofs:
update(+dictionary,@ground,+term,-dictionary) - zero_or_one

---

update/5

Updates the value associated with a key in a dictionary, returning the updated dictionary. Fails if it cannot find the key or if the existing value does not unify.

Compilation flags:
    static

Template:
    update(OldDictionary,Key,OldValue,NewValue,NewDictionary)
Mode and number of proofs:
    update(+dictionary,@ground,?term,+term,-dictionary) - zero_or_one

---

update/3

Updates the key-value pairs in a dictionary, returning the updated dictionary. Fails if it cannot find one of the keys.

Compilation flags:
    static

Template:
    update(OldDictionary,Pairs,NewDictionary)
Mode and number of proofs:
    update(+dictionary,@list(pair),-dictionary) - zero_or_one

---

empty/1

True iff the dictionary is empty.

Compilation flags:
    static

Template:
    empty(Dictionary)
Mode and number of proofs:
    empty(@dictionary) - zero_or_one

---

lookup/4

Lookups a matching key-value pair from a dictionary and returns the splayed dictionary with the key at the root. Fails if the key is not found. In implementations that do not update the dictionary on lookup, the same dictionary is returned.

Compilation flags:
    static

Template:
    lookup(Key,Value,Dictionary,SplayedDictionary)
Mode and number of proofs:
    lookup(+ground,?term,+tree,-tree) - zero_or_one

---

lookup/3

Lookups a matching key-value pair from a dictionary. Fails if no match is found.

Compilation flags:
    static

Template:
    lookup(Key,Value,Dictionary)
Mode and number of proofs:
    lookup(+ground,?term,@dictionary) - zero_or_one
    lookup(-ground,?term,@dictionary) - zero_or_more

---

lookup/2

Lookups all matching key-value pairs from a dictionary. Fails if it cannot find one of the keys or if a value for a key does not unify.

Compilation flags:
    static

Template:
lookup(Pairs,Dictionary)
Mode and number of proofs:
lookup(+list(pair),@dictionary) - zero_or_one

---

intersection/2

True iff the values of the dictionaries common keys unify. Trivially true when there are no common keys.

Compilation flags:
static

Template:
intersection(Dictionary1,Dictionary2)
Mode and number of proofs:
intersection(+dictionary,+dictionary) - zero_or_one

---

intersection/3

Returns the (possibly empty) intersection between two dictionaries when the values of their common keys unify.

Compilation flags:
static

Template:
intersection(Dictionary1,Dictionary2,Intersection)
Mode and number of proofs:
intersection(+dictionary,+dictionary,-dictionary) - zero_or_one

---

previous/4

Returns the previous pair in a dictionary given a key. Fails if there is no previous pair.

Compilation flags:
    static

Template:
    previous(Dictionary,Key,Previous,Value)
Mode and number of proofs:
    previous(+dictionary,+key,-key,-value) - zero_or_one

---

next/4

Returns the next pair in a dictionary given a key. Fails if there is no next pair.

Compilation flags:
    static

Template:
    next(Dictionary,Key,Next,Value)
Mode and number of proofs:
    next(+dictionary,+key,-key,-value) - zero_or_one

---

min/3

Returns the pair with the minimum key (as per standard order) in a dictionary. Fails if the dictionary is empty.

Compilation flags:
    static

Template:
    min(Dictionary,Key,Value)
Mode and number of proofs:
    min(+dictionary,-key,-value) - zero_or_one

---

max/3

Returns the pair with the maximum key (as per standard order) in a dictionary. Fails if the dictionary is empty.

Compilation flags:
    static

Template:
    max(Dictionary,Key,Value)
Mode and number of proofs:
    max(+dictionary,-key,-value) - zero_or_one

---

delete_min/4

Deletes the pair with the minimum key (as per standard order) from a dictionary, returning the deleted pair and the updated dictionary. Fails if the dictionary is empty.

Compilation flags:
    static

Template:
    delete_min(OldDictionary,Key,Value,NewDictionary)
Mode and number of proofs:
    delete_min(+dictionary,-key,-value,-dictionary) - zero_or_one

---

delete_max/4

Deletes the pair with the maximum key (as per standard order) from a dictionary, returning the deleted pair and the updated dictionary. Fails if the dictionary is empty.

Compilation flags:
    static

Template:

delete_max(OldDictionary,Key,Value,NewDictionary)
Mode and number of proofs:
delete_max(+dictionary,-key,-value,-dictionary) - zero_or_one

---

keys/2

Returns a list with all the dictionary keys in ascending order (as per standard order).

Compilation flags:
static

Template:
keys(Dictionary,Keys)
Mode and number of proofs:
keys(@dictionary,-list) - one

---

values/2

Returns a list with all the dictionary values in ascending order of the keys (as per standard order).

Compilation flags:
static

Template:
values(Dictionary,Values)
Mode and number of proofs:
values(@dictionary,-list) - one

---

map/2

Maps a closure over each dictionary key-value pair. Fails if the mapped closure attempts to modify the keys.

Compilation flags:
    static

Template:
    map(Closure,Dictionary)
Meta-predicate template:
    map(1,*)
Mode and number of proofs:
    map(@callable,+dictionary) - zero_or_more

---

map/3

Maps a closure over each dictionary key-value pair, returning the new dictionary. Fails if the mapped closure attempts to modify the keys.

Compilation flags:
    static

Template:
    map(Closure,OldDictionary,NewDictionary)
Meta-predicate template:
    map(2,*,*)
Mode and number of proofs:
    map(@callable,+dictionary,-dictionary) - zero_or_more

---

apply/4

Applies a closure to a specific key-value pair, returning the new dictionary. Fails if the key cannot be found or if the mapped closure attempts to modify the key.

Compilation flags:
    static

Template:
    apply(Closure,OldDictionary,Key,NewDictionary)
Meta-predicate template:
    apply(2,*,*,*)
Mode and number of proofs:
    apply(+callable,+dictionary,+key,-dictionary) - zero_or_one

---

size/2

Number of dictionary entries.

Compilation flags:
    static

Template:
    size(Dictionary,Size)
Mode and number of proofs:
    size(@dictionary,?integer) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

avltree, bintree, rbtree, splaytree

object

1.56.4 rbtree

Red-Black tree implementation of the dictionary protocol. Uses standard order to compare keys.

Availability:
logtalk_load(dictionaries(loader))

Author: Vitor Santos Costa; Logtalk port and additional predicates by Paulo Moura.
Version: 1:11:0
Date: 2026-02-10

Compilation flags:
static, context_switching_calls

Implements:
public dictionaryp
Extends:
public term

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 apply/4 as_curly_bracketed/2 as_dictionary/2 as_list/2 check/1 clone/3 clone/4 delete/4 delete_max/4 delete_min/4 depth/2 empty/1 ground/1 insert/4 intersection/2 intersection/3 keys/2 lookup/2 lookup/3 lookup/4 map/2 map/3 max/3 min/3 new/1 next/4 numbervars/1 numbervars/3 occurs/2 previous/4 singletons/2 size/2 subsumes/2 subterm/2 update/3 update/4 update/5 valid/1 values/2 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
  - partial_map/4
- Protected predicates
- Private predicates
- Operators

Public predicates

partial_map/4

Applies a closure to the tree pairs identified by a set of keys.

Compilation flags:
    static

Template:
    partial_map(Tree,Keys,Closure,NewTree)
Meta-predicate template:
    partial_map(*,*,2,*)
Mode and number of proofs:
    partial_map(+tree,+list,@closure,-tree) - zero_or_one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

avltree, splaytree, bintree

object

1.56.5 splaytree

Splay tree implementation of the dictionary protocol. A splay tree is a self-adjusting binary search tree with the property that recently accessed elements are quick to access again. Uses standard order to compare keys.

Availability:
    logtalk_load(dictionaries(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-10

Compilation flags:
static, context_switching_calls

Implements:
public dictionaryp
Extends:
public term
Uses:
list

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 apply/4 as_curly_bracketed/2 as_dictionary/2 as_list/2 check/1 clone/3 clone/4 delete/4 delete_max/4 delete_min/4 depth/2 empty/1 ground/1 insert/4 intersection/2 intersection/3 keys/2 lookup/2 lookup/3 lookup/4 map/2 map/3 max/3 min/3 new/1 next/4 numbervars/1 numbervars/3 occurs/2 previous/4 singletons/2 size/2 subsumes/2 subterm/2 update/3 update/4 update/5 valid/1 values/2 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

avltree, bintree, rbtree

object

1.56.6 two3tree

2-3 tree implementation.

Availability:
logtalk_load(dictionaries(loader))

Author: Michael T. Richter
Version: 1:0:0
Date: 2026-04-04

Compilation flags:
static, context_switching_calls

Implements:
public dictionaryp
Extends:
public term
Uses:
list

Remarks:

(none)

Inherited public predicates:

(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 apply/4 as_curly_bracketed/2 as_dictionary/2 as_list/2 check/1 clone/3 clone/4 delete/4 delete_max/4 delete_min/4 depth/2 empty/1 ground/1 insert/4 intersection/2 intersection/3 keys/2 lookup/2 lookup/3 lookup/4 map/2 map/3 max/3 min/3 new/1 next/4 numbervars/1 numbervars/3 occurs/2 previous/4 singletons/2 size/2 subsumes/2 subterm/2 update/3 update/4 update/5 valid/1 values/2 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
  - union/3
- Protected predicates
- Private predicates
  - union_impl/3
  - union_impl_list/3
  - clone_impl/3
  - clone_impl2/4
  - handle_root_underflow/2
  - delete_impl/5
  - fix_node2_left_underflow/6
  - fix_node2_right_underflow/6
  - fix_node3_left_underflow/9
  - fix_node3_middle_underflow/9
  - fix_node3_right_underflow/9
  - as_curly_bracketed_impl/2
  - as_curly_bracketed_impl/3
  - insert_impl/4
  - insert_empty/4
  - insert_node2_2/4
  - insert_node2_4/4
  - insert_node3_4/4
  - insert_node3_7/4
  - intersection_impl/4
  - is_node4/1
  - map_impl/2

  - map_impl/3
  - next_impl/5
  - previous_impl/5
  - node2_from_node4/2
  - update_impl/3
  - collect/4
  - select/4
- Operators

Public predicates

union/3

Computes the union of two dictionaries. If a key appears in both, the value from the larger dictionary is kept.

Compilation flags:
    static

Template:
    union(Tree1,Tree2,Union)
Mode and number of proofs:
    union(+two3tree,+two3tree,-two3tree) - zero_or_one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

union_impl/3

Inserts all key-value pairs of the smaller tree into the larger tree to compute the union.

Compilation flags:
    static

Mode and number of proofs:

union_impl(+two3tree,+two3tree,-two3tree) - zero_or_one

---

union_impl_list/3

Inserts a list of key-value pairs into a dictionary one by one.

Compilation flags:
static

Mode and number of proofs:
union_impl_list(+list(pair),+two3tree,-two3tree) - zero_or_one

---

clone_impl/3

Recursively clones a tree, leaving all values unbound and collecting the pairs of the clone.

Compilation flags:
static

Mode and number of proofs:
clone_impl(+two3tree,-two3tree,-list(pair)) - one

---

clone_impl2/4

Recursively clones a tree, returning both the original pairs and the clone pairs.

Compilation flags:
static

Mode and number of proofs:
clone_impl2(+two3tree,-two3tree,-list(pair),-list(pair)) - one

---

handle_root_underflow/2

After a deletion, if the root has become a 2-node with a single child, replace it by that child; otherwise keep the root.

Compilation flags:
static

Mode and number of proofs:
handle_root_underflow(+term,-term) - one

---

delete_impl/5

Recursive deletion that returns the resulting tree and a flag indicating whether an underflow has occurred at the current node.

Compilation flags:
static

Mode and number of proofs:
delete_impl(+term,+term,?term,-term,-boolean) - zero_or_one

---

fix_node2_left_underflow/6

Repairs an underflow after a deletion in the left subtree of a node2.

Compilation flags:
static

Mode and number of proofs:
fix_node2_left_underflow(+term,+term,+term,+term,-term,-boolean) - one

---

fix_node2_right_underflow/6

Repairs an underflow after a deletion in the right subtree of a node2.

Compilation flags:
static

Mode and number of proofs:
fix_node2_right_underflow(+term,+term,+term,+term,-term,-boolean) - one

---

fix_node3_left_underflow/9

Repairs an underflow after a deletion in the left subtree of a node3.

Compilation flags:
static

Mode and number of proofs:
fix_node3_left_underflow(+term,+term,+term,+term,+term,+term,+term,-term,-boolean) - one

---

fix_node3_middle_underflow/9

Repairs an underflow after a deletion in the middle subtree of a node3.

Compilation flags:
static

Mode and number of proofs:
fix_node3_middle_underflow(+term,+term,+term,+term,+term,+term,+term,-term,-boolean) - one

---

fix_node3_right_underflow/9

Repairs an underflow after a deletion in the right subtree of a node3.

Compilation flags:
    static

Mode and number of proofs:
    fix_node3_right_underflow(+term,+term,+term,+term,+term,+term,+term,-term,-boolean) - one

---

as_curly_bracketed_impl/2

Builds a curly-bracketed term from a list of key-value pairs.

Compilation flags:
    static

Mode and number of proofs:
    as_curly_bracketed_impl(+list(pairs),-term) - one

---

as_curly_bracketed_impl/3

Helper that accumulates a comma-separated list inside curly braces.

Compilation flags:
    static

Mode and number of proofs:
    as_curly_bracketed_impl(+list(pairs),+pair,-term) - one

---

insert_impl/4

Recursive insertion that may return a node4 on the way up.

Compilation flags:
    static

Mode and number of proofs:
    insert_impl(+term,+term,+term,-term) - one

---

insert_empty/4

Inserts into an empty tree.

Compilation flags:
    static

Mode and number of proofs:
    insert_empty(+empty,+term,+term,-term) - one

---

insert_node2_2/4

Inserts a key-value pair into a leaf 2-node, creating a leaf 3-node.

Compilation flags:
    static

Mode and number of proofs:
    insert_node2_2(+term,+term,+term,-term) - one

---

insert_node2_4/4

Inserts into a non-leaf 2-node, possibly splitting a child 4-node.

Compilation flags:
    static

Mode and number of proofs:
    insert_node2_4(+term,+term,+term,-term) - one

---

insert_node3_4/4

Inserts a key-value pair into a leaf 3-node, creating a leaf 4-node.

Compilation flags:
    static

Mode and number of proofs:
    insert_node3_4(+term,+term,+term,-term) - one

---

insert_node3_7/4

Inserts into a non-leaf 3-node, possibly splitting a child 4-node.

Compilation flags:
    static

Mode and number of proofs:
    insert_node3_7(+term,+term,+term,-term) - one

---

intersection_impl/4

Computes the intersection of a list of key-value pairs with a tree, inserting common pairs into an accumulator.

Compilation flags:
    static

Mode and number of proofs:
    intersection_impl(+list(pair),+two3tree,+list(pair),-list(pair)) - zero_or_one

---

is_node4/1

True if the term is a 4-node (temporary representation).

Compilation flags:
    static

Mode and number of proofs:
    is_node4(+term) - zero_or_one

---

map_impl/2

Traverses the tree and applies a closure to each key-value pair.

Compilation flags:
    static

Meta-predicate template:
    map_impl(*,1)
Mode and number of proofs:
    map_impl(+two3tree,@callable) - zero_or_more

---

map_impl/3

Traverses the tree, applies a closure to each key-value pair, and builds a new tree with the results.

Compilation flags:
    static

Meta-predicate template:
    map_impl(*,2,*)
Mode and number of proofs:
    map_impl(+two3tree,@callable,-two3tree) - zero_or_more

---

next_impl/5

Recursively finds the next key-value pair after a given key.

Compilation flags:
    static

Mode and number of proofs:
    next_impl(+two3tree,+key,-key,-value,+term) - zero_or_one

---

previous_impl/5

Recursively finds the previous key-value pair before a given key.

Compilation flags:
    static

Mode and number of proofs:
    previous_impl(+two3tree,+key,-key,-value,+term) - zero_or_one

---

node2_from_node4/2

Converts a 4-node (temporary) into a balanced 2-node with two 2-node children.

Compilation flags:
    static

Mode and number of proofs:
    node2_from_node4(+term,-term) - one

---

update_impl/3

Updates a dictionary with a list of key-value pairs sequentially.

Compilation flags:
    static

Mode and number of proofs:
    update_impl(@list(pair),+two3tree,-two3tree) - zero_or_one

---

collect/4

In-order traversal accumulating selected projections of nodes.

Compilation flags:
    static

Template:
    collect(Tree,Selector,Accumulator,ReversedResult)
Mode and number of proofs:
    collect(+two3tree,+selector,+list(term),-list(term)) - one

---

select/4

Projects a key-value pair according to the selector.

Compilation flags:
    static

Template:
    select(Selector,Key,Value,Element)
Mode and number of proofs:
    select(+selector,+key,+value,-term) - one

Operators

(none)

See also

avltree, bintree, rbtree, splaytree, dictionaryp

1.57 dif

object

1.57.1 dif

Provides dif/2 and derived predicates.

Availability:
    logtalk_load(dif(loader))

Author: Paulo Moura
Version: 1:3:0
Date: 2023-10-02

Compilation flags:
    static, context_switching_calls

Dependencies:

(none)

Remarks:

- Supported backend Prolog systems: B-Prolog, ECLiPSe, SICStus Prolog, SWI-Prolog, Trealla Prolog, and YAP.

Inherited public predicates:
(none)

- Public predicates
  - dif/2
  - dif/1
- Protected predicates
- Private predicates
- Operators

Public predicates

dif/2

Sets a constraint that is true iff the two terms are different.

Compilation flags:
static

Template:
dif(Term1,Term2)
Mode and number of proofs:
dif(+term,+term) - zero_or_one

dif/1

Sets a set of constraints that are true iff all terms in a list are different.

Compilation flags:
    static

Template:
    dif(Terms)
Mode and number of proofs:
    dif(+list(term)) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.58 dimension_reduction_protocols

category

1.58.1 dimension_reducer_common

Shared predicates for dimension reducer learning defaults, dataset helpers, transformation, export, and printing.

Availability:
    logtalk_load(dimension_reduction_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-07

Compilation flags:
static

Implements:
public dimension_reducer_protocol
Extends:
public options
Uses:
format
linear_algebra
list
numberlist
population
type
user

Remarks:
(none)

Inherited public predicates:
check_dimension_reducer/1 check_option/1 check_options/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 dimension_reducer_options/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 print_dimension_reducer/1 transform/3 valid_dimension_reducer/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
  - check_component_count/3
  - dimension_reducer_data/3
  - dimension_reducer_diagnostics_data/2
  - print_dimension_reducer_properties/1
  - example_attribute_values/2
  - dataset_attributes/2
  - check_continuous_attributes/1
  - check_examples_non_empty/2
  - check_example_values/2
  - check_example_attributes/2
  - attribute_value/3
  - build_encoders/4

  - base_dimension_reducer_diagnostics/6
  - preprocessing_diagnostics/3
  - iterative_dimension_reducer_diagnostics/11
  - component_iteration_diagnostics/4
  - zero_vector_like/2
  - basis_vector/3
  - initial_vectors/2
  - basis_initial_vectors/3
  - extract_components/5
  - known_attribute_values/3
  - examples_to_rows/3
  - encode_instance/3
  - encoder_attribute_names/2
  - project_components/4
  - valid_linear_encoders/1
  - valid_projection_components/2
  - valid_dimension_reducer_metadata/1
  - print_dimension_reducer_details/3
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

check_component_count/3

Checks that a requested component count does not exceed the supported maximum and returns the accepted count.

Compilation flags:
    static

Template:
    check_component_count(RequestedComponentCount,MaxComponentCount,ComponentCount)
Mode and number of proofs:

check_component_count(+integer,+integer,-integer) - one

---

dimension_reducer_data/3

Default hook predicate for exposing the learned encoders and projection components from a reducer term. Importing implementations may override it when using a non-standard reducer representation.

Compilation flags:
static

Template:
dimension_reducer_data(DimensionReducer,Encoders,Components)
Mode and number of proofs:
dimension_reducer_data(+compound,-list,-list) - one

---

dimension_reducer_diagnostics_data/2

Default hook predicate for exposing diagnostics metadata from a reducer term. Importing implementations may override it when using a non-standard reducer representation.

Compilation flags:
static

Template:
dimension_reducer_diagnostics_data(DimensionReducer,Diagnostics)
Mode and number of proofs:
dimension_reducer_diagnostics_data(+compound,-list(compound)) - one

---

print_dimension_reducer_properties/1

Hook predicate that importing dimension reducer implementations must define in order to print the learned reducer in a human-readable form.

Compilation flags:
    static

Template:
    print_dimension_reducer_properties(DimensionReducer)
Mode and number of proofs:
    print_dimension_reducer_properties(+compound) - one

---

example_attribute_values/2

Hook predicate that importing dimension reducer implementations must define in order to extract example attribute values from their local training example representation.

Compilation flags:
    static

Template:
    example_attribute_values(Example,AttributeValues)
Mode and number of proofs:
    example_attribute_values(+compound,-list(pair)) - one

---

dataset_attributes/2

Collects the dataset attribute declarations as Attribute-Values pairs.

Compilation flags:
    static

Template:
    dataset_attributes(Dataset,Attributes)
Mode and number of proofs:
    dataset_attributes(+object_identifier,-list(pair)) - one

---

check_continuous_attributes/1

Checks that all declared dataset attributes are continuous.

Compilation flags:
    static

Template:
    check_continuous_attributes(Attributes)
Mode and number of proofs:
    check_continuous_attributes(+list(pair)) - one

---

check_examples_non_empty/2

Checks that a training example collection is not empty.

Compilation flags:
    static

Template:
    check_examples_non_empty(Dataset,Examples)
Mode and number of proofs:
    check_examples_non_empty(+object_identifier,+list) - one

---

check_example_values/2

Checks that all example attribute values are present and numeric for the declared attributes.

Compilation flags:
    static

Template:
    check_example_values(Examples,AttributeNames)
Mode and number of proofs:

check_example_values(+list,+list(atom)) - one

---

check_example_attributes/2

Checks that a single example contains numeric values for all declared attributes.

Compilation flags:
static

Template:
check_example_attributes(AttributeNames,AttributeValues)
Mode and number of proofs:
check_example_attributes(+list(atom),+list(pair)) - one

---

attribute_value/3

Looks up an attribute value in a list of Attribute-Value pairs.

Compilation flags:
static

Template:
attribute_value(Attribute,AttributeValues,Value)
Mode and number of proofs:
attribute_value(+atom,+list(pair),-term) - one

---

build_encoders/4

Builds continuous feature encoders by computing per-attribute centering and optional scaling statistics.

Compilation flags:
static

Template:

build_encoders(AttributeNames,Examples,Options,Encoders)
Mode and number of proofs:
build_encoders(+list(atom),+list,+list(compound),-list(compound)) - one

---

base_dimension_reducer_diagnostics/6

Builds common diagnostics metadata terms for a learned reducer and appends reducer-specific diagnostics terms.

Compilation flags:
static

Template:
base_dimension_reducer_diagnostics(Model,AttributeNames,Components,Options,
ExtraDiagnostics,Diagnostics)
Mode and number of proofs:
base_dimension_reducer_diagnostics(+atom,+list(atom),+list,+list(compound),+list(compound),
-list(compound)) - one

---

preprocessing_diagnostics/3

Builds shared preprocessing diagnostics metadata from an explicit centering flag and the effective training options.

Compilation flags:
static

Template:
preprocessing_diagnostics(Center,Options,Preprocessing)
Mode and number of proofs:
preprocessing_diagnostics(+boolean,+list(compound),-list(compound)) - one

---

iterative_dimension_reducer_diagnostics/11

Builds diagnostics metadata for reducers that report sample counts, iterative convergence terms, and optional leading or trailing reducer-specific diagnostics.

Compilation flags:
static

Template:
iterative_dimension_reducer_diagnostics(Model,AttributeNames,Components,SampleCount,Options,LeadingDiagnostics,Convergence,Iterations,FinalDelta,TrailingDiagnostics,Diagnostics)
Mode and number of proofs:
iterative_dimension_reducer_diagnostics(+atom,+list(atom),+list,+integer,+list(compound),+list(compound),+term,+term,+term,+list(compound),-list(compound)) - one

---

component_iteration_diagnostics/4

Extracts per-component convergence, iteration, and final-delta lists from component diagnostics records.

Compilation flags:
static

Template:
component_iteration_diagnostics(ComponentDiagnostics,Convergences,IterationCounts,FinalDeltas)
Mode and number of proofs:
component_iteration_diagnostics(+list(compound),-list(atom),-list(integer),-list(number)) - one

---

zero_vector_like/2

Constructs a zero vector matching the length of the first vector in a list of vectors, or returns the empty list when the input is empty.

Compilation flags:
static

Template:
zero_vector_like(Vectors,ZeroVector)

Mode and number of proofs:
zero_vector_like(+list(list(number)),-list(number)) - one

---

basis_vector/3

Constructs a canonical basis vector for the requested size and one-based index.

Compilation flags:
static

Template:
basis_vector(Size,Index,Vector)
Mode and number of proofs:
basis_vector(+integer,+integer,-list(number)) - one

---

initial_vectors/2

Constructs the default all-ones initial vector followed by canonical basis vectors for the requested size.

Compilation flags:
static

Template:
initial_vectors(Size,Vectors)
Mode and number of proofs:
initial_vectors(+integer,-list(list(number))) - one

---

basis_initial_vectors/3

Constructs canonical basis vectors from the given one-based index up to the requested size.

Compilation flags:
static

Template:
basis_initial_vectors(Index,Size,Vectors)
Mode and number of proofs:
basis_initial_vectors(+integer,+integer,-list(list(number))) - one

---

extract_components/5

Extracts leading positive eigen-components from a numeric matrix using the shared symmetric eigensolver until the requested count or the configured tolerance is reached.

Compilation flags:
static

Template:
extract_components(Matrix,Requested,Options,Components,Eigenvalues)
Mode and number of proofs:
extract_components(+list(list(number)),+integer,+list(compound),-list(list(number)),-list(number)) - one

---

known_attribute_values/3

Collects the known numeric values for a given attribute across the training examples.

Compilation flags:
static

Template:
known_attribute_values(Examples,Attribute,Values)
Mode and number of proofs:
known_attribute_values(+list,+atom,-list(number)) - one

---

examples_to_rows/3

Encodes a list of training examples into numeric feature rows using the importing reducer example hook and learned encoders.

Compilation flags:
    static

Template:
    examples_to_rows(Examples,Encoders,Rows)
Mode and number of proofs:
    examples_to_rows(+list,+list(compound),-list(list(number))) - one

---

encode_instance/3

Encodes an instance using the learned continuous attribute encoders.

Compilation flags:
    static

Template:
    encode_instance(Encoders,AttributeValues,Features)
Mode and number of proofs:
    encode_instance(+list(compound),+list(pair),-list(number)) - one

---

encoder_attribute_names/2

Collects encoder attribute names preserving encoder order.

Compilation flags:
    static

Template:
    encoder_attribute_names(Encoders,AttributeNames)
Mode and number of proofs:
    encoder_attribute_names(+list(compound),-list(atom)) - one

---

project_components/4

Projects encoded features onto the learned components and returns component_N-Score pairs.

Compilation flags:
    static

Template:
    project_components(Components,Features,Index,ReducedInstance)
Mode and number of proofs:
    project_components(+list(list(number)),+list(number),+integer,-list(pair)) - one

---

valid_linear_encoders/1

True when a list of encoders only contains valid continuous/3 encoder terms with distinct attributes.

Compilation flags:
    static

Template:
    valid_linear_encoders(Encoders)
Mode and number of proofs:
    valid_linear_encoders(+list(compound)) - zero_or_one

---

valid_projection_components/2

True when projection components are numeric vectors compatible with the encoder feature dimension.

Compilation flags:
    static

Template:
    valid_projection_components(Encoders,Components)
Mode and number of proofs:

valid_projection_components(+list(compound),+list(list(number))) - zero_or_one

---

valid_dimension_reducer_metadata/1

True when diagnostics metadata records the reducer model and effective training options.

Compilation flags:
static

Template:
valid_dimension_reducer_metadata(Diagnostics)
Mode and number of proofs:
valid_dimension_reducer_metadata(+list(compound)) - zero_or_one

---

print_dimension_reducer_details/3

Prints the common diagnostics, encoders, and component-count lines used by reducer-specific pretty printers.

Compilation flags:
static

Template:
print_dimension_reducer_details(Diagnostics,Encoders,Components)
Mode and number of proofs:
print_dimension_reducer_details(+list(compound),+list(compound),+list) - one

---

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.58.2 dimension_reducer_protocol

Protocol for machine learning dimension reducers.

Availability:
logtalk_load(dimension_reduction_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-30

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - learn/3
  - learn/2
  - transform/3
  - check_dimension_reducer/1
  - valid_dimension_reducer/1
  - diagnostics/2
  - diagnostic/2
  - dimension_reducer_options/2
  - export_to_clauses/4

  - export_to_file/4
  - print_dimension_reducer/1
- Protected predicates
- Private predicates
- Operators

Public predicates

learn/3

Learns a dimension reducer from the given dataset object using the given options.

Compilation flags:
    static

Template:
    learn(Dataset,DimensionReducer,Options)
Mode and number of proofs:
    learn(+object_identifier,-compound,++list(compound)) - one

---

learn/2

Learns a dimension reducer from the given dataset object using default options.

Compilation flags:
    static

Template:
    learn(Dataset,DimensionReducer)
Mode and number of proofs:
    learn(+object_identifier,-compound) - one

---

transform/3

Transforms a new instance into a reduced representation using the learned dimension reducer. The instance is a list of Attribute-Value pairs and the reduced representation is a list of Component-Value pairs.

Compilation flags:
    static

Template:
    transform(DimensionReducer,Instance,ReducedInstance)
Mode and number of proofs:
    transform(+compound,+list,-list(pair)) - one

---

check_dimension_reducer/1

Checks that a learned dimension reducer term is structurally valid for the receiving implementation. Throws an exception when the term is not a valid dimension reducer representation.

Compilation flags:
    static

Template:
    check_dimension_reducer(DimensionReducer)
Mode and number of proofs:
    check_dimension_reducer(@compound) - one_or_error

Exceptions:
    DimensionReducer is a variable:
        instantiation_error
    DimensionReducer is neither a variable nor a valid dimension reducer:
        domain_error(dimension_reducer,DimensionReducer)

---

valid_dimension_reducer/1

True when a learned dimension reducer term is structurally valid for the receiving implementation. Succeeds iff check_dimension_reducer/1 succeeds without throwing an exception.

Compilation flags:
    static

Template:
    valid_dimension_reducer(DimensionReducer)
Mode and number of proofs:
    valid_dimension_reducer(@compound) - zero_or_one

---

diagnostics/2

Returns diagnostics and metadata associated with a learned dimension reducer in a representation-independent way.

Compilation flags:
    static

Template:
    diagnostics(DimensionReducer,Diagnostics)
Mode and number of proofs:
    diagnostics(+compound,-list(compound)) - one

---

diagnostic/2

Tests or enumerates individual diagnostics metadata terms for a learned dimension reducer.

Compilation flags:
    static

Template:
    diagnostic(DimensionReducer,Diagnostic)
Mode and number of proofs:
    diagnostic(+compound,?compound) - zero_or_more

---

dimension_reducer_options/2

Returns the effective training options recorded in a learned dimension reducer diagnostics list.

Compilation flags:
    static

Template:
    dimension_reducer_options(DimensionReducer,Options)
Mode and number of proofs:
    dimension_reducer_options(+compound,-list(compound)) - zero_or_one

---

export_to_clauses/4

Converts a dimension reducer into a list of predicate clauses. Functor is the functor for the generated single-argument predicate clauses whose argument is the serialized dimension reducer term.

Compilation flags:
    static

Template:
    export_to_clauses(Dataset,DimensionReducer,Functor,Clauses)
Mode and number of proofs:
    export_to_clauses(+object_identifier,+compound,+callable,-list(clause)) - one

---

export_to_file/4

Exports a dimension reducer to a file. Functor is the functor for the generated single-argument predicate clauses whose argument is the serialized dimension reducer term.

Compilation flags:
    static

Template:

export_to_file(Dataset,DimensionReducer,Functor,File)
Mode and number of proofs:
export_to_file(+object_identifier,+compound,+callable,+atom) - one

---

print_dimension_reducer/1

Prints a dimension reducer to the current output stream in a human-readable format.

Compilation flags:
static

Template:
print_dimension_reducer(DimensionReducer)
Mode and number of proofs:
print_dimension_reducer(+compound) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

dimension_reduction_dataset_protocol, supervised_dimension_reduction_dataset_protocol, target_supervised_dimension_reduction_dataset_protocol

protocol

1.58.3 dimension_reduction_dataset_protocol

Protocol for datasets used with unsupervised dimension reduction algorithms.

Availability:
logtalk_load(dimension_reduction_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-20

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - attribute_values/2
  - example/2
- Protected predicates
- Private predicates
- Operators

Public predicates

attribute_values/2

Enumerates by backtracking the attributes and their possible values. For discrete attributes, Values is a list of possible values. For continuous (numeric) attributes, Values is the atom continuous.

Compilation flags:
static

Template:
attribute_values(Attribute,Values)
Mode and number of proofs:
attribute_values(?atom,-list(atom)) - zero_or_more
attribute_values(?atom,-atom) - zero_or_more

---

example/2

Enumerates by backtracking the examples in the dataset. Each example has an Id and a list of Attribute-Value pairs.

Compilation flags:
static

Template:
example(Id,AttributeValues)
Mode and number of proofs:
example(-integer,-list(pair)) - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

protocol

1.58.4 supervised_dimension_reduction_dataset_protocol

Protocol for labeled datasets used with supervised dimension reduction algorithms such as LDA.

Availability:
logtalk_load(dimension_reduction_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-18

Compilation flags:
static

Extends:
public dimension_reduction_dataset_protocol

Remarks:
(none)

Inherited public predicates:
attribute_values/2 example/2

- Public predicates
  - class/1
  - class_values/1
  - example/3
- Protected predicates
- Private predicates
- Operators

Public predicates

class/1

Returns the name of the target class attribute.

Compilation flags:
    static

Template:
    class(Class)
Mode and number of proofs:
    class(-atom) - one

---

class_values/1

Returns the list of possible values for the target class attribute.

Compilation flags:
    static

Template:
    class_values(Values)
Mode and number of proofs:
    class_values(-list(atom)) - one

---

example/3

Enumerates by backtracking the labeled examples in the dataset. Each example has an Id, a Class value, and a list of Attribute-Value pairs.

Compilation flags:
    static

Template:
    example(Id,Class,AttributeValues)
Mode and number of proofs:

example(-integer,-atom,-list(pair)) - zero_or_more

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.58.5 target_supervised_dimension_reduction_dataset_protocol

Protocol for target-valued datasets used with supervised dimension reduction algorithms such as PLS projections.

Availability:
logtalk_load(dimension_reduction_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-28

Compilation flags:
static

Extends:
public dimension_reduction_dataset_protocol

Remarks:
(none)

Inherited public predicates:
attribute_values/2 example/2

- Public predicates
  - target/1
  - example/3
- Protected predicates
- Private predicates
- Operators

Public predicates

target/1

Returns the name of the numeric target attribute.

Compilation flags:
    static

Template:
    target(Target)
Mode and number of proofs:
    target(-atom) - one

---

example/3

Enumerates by backtracking the target-valued examples in the dataset. Each example has an Id, a numeric Target value, and a list of Attribute-Value pairs.

Compilation flags:
    static

Template:
    example(Id,Target,AttributeValues)
Mode and number of proofs:
    example(-integer,-number,-list(pair)) - zero_or_more

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.59 doclet

object

1.59.1 doclet

Utility object to help automate (re)generating documentation for a project.

Availability:
logtalk_load(doclet(loader))

Author: Paulo Moura
Version: 0:5:0
Date: 2017-01-05

Compilation flags:
static, context_switching_calls

Provides:
logtalk::message_tokens//2
Uses:
logtalk
os

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - update/0
  - doc_goal/1
  - shell_command/1
- Protected predicates
- Private predicates
- Operators

Public predicates

update/0

Updates the project documentation, first by calling a sequence of goals and second by executing a sequence of shell commands. Fails if any goal or shell command fails.

Compilation flags:
    static

Mode and number of proofs:
    update - zero_or_one

---

doc_goal/1

Table of goals, typically using the diagrams and the lgtdoc tools, used to generate the documentation. Goals are called in the order they are defined and in the context of the user pseudo-object.

Compilation flags:
    static

Template:
    doc_goal(Goal)
Mode and number of proofs:
    doc_goal(?callable) - one_or_more

---

shell_command/1

Table of shell commands to convert intermediate documentation files into user-friendly documentation. Commands are executed in the order they are defined.

Compilation flags:
    static

Template:
    shell_command(Command)
Mode and number of proofs:
    shell_command(?atom) - one_or_more

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

lgtdocp, diagram(Format)

1.60 eclat_pattern_miner

object

1.60.1 eclat_pattern_miner

Eclat frequent itemset miner for transaction datasets using portable vertical tidsets and depth-first recursive prefix extension.

Availability:
logtalk_load(eclat_pattern_miner(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public frequent_pattern_mining_common
Uses:
avltree
format
list

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_pattern_miner/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 mine/2 mine/3 option/2 option/3 pattern_miner_options/2 print_pattern_miner/1 valid_option/1 valid_options/1 valid_pattern_miner/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

pattern_miner_protocol, transaction_dataset_protocol, apriori_pattern_miner, fp_growth_pattern_miner

1.61 edcg

object

1.61.1 edcg

Multiple hidden parameters: an extension to Prolog's DCG notation. Ported to Logtalk as a hook object.

Availability:
logtalk_load(edcg(loader))

Author: Peter Van Roy; adapted to Logtalk by Paulo Moura.
Version: 1:4:2
Date: 2020-04-08

Copyright: Copyright (C) 1992 Peter Van Roy
License: MIT

Compilation flags:
static, context_switching_calls

Implements:
public expanding
Provides:
logtalk::message_tokens//2
Uses:
list
logtalk

Remarks:

- Usage: Compile source files with objects (or categories) defining EDCGs using the compiler option hook(edcg).

Inherited public predicates:
goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
  - pred_info/3
  - acc_info/7
  - acc_info/5
  - pass_info/2
  - pass_info/1
- Operators
  - op(1200,xfx,-->>)

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

pred_info/3

Declares predicates that have the listed hidden parameters.

Compilation flags:
    dynamic

Template:
    pred_info(Name,Arity,HiddenParameters)
Mode and number of proofs:
    pred_info(?atom,?integer,?list(atom)) - zero_or_more

---

acc_info/7

Long form for declaring accumulators.

Compilation flags:
    dynamic

Template:
    acc_info(Accumulator,Term,Left,Right,Joiner,LStart,RStart)
Mode and number of proofs:
    acc_info(?atom,?term,?term,?term,?callable,?term,?term) - zero_or_more

---

acc_info/5

Short form for declaring accumulators.

Compilation flags:
    dynamic

Template:
    acc_info(Accumulator,Term,Left,Right,Joiner)
Mode and number of proofs:
    acc_info(?atom,?term,?term,?term,?callable) - zero_or_more

---

pass_info/2

Long form for declaring passed arguments. Passed arguments are conceptually the same as accumulators with =/2 as the joiner function.

Compilation flags:
    dynamic

Template:
    pass_info(Argument,PStart)
Mode and number of proofs:
    pass_info(?atom,?term) - zero_or_more

---

pass_info/1

Short form for declaring passed arguments. Passed arguments are conceptually the same as accumulators with =/2 as the joiner function.

Compilation flags:
    dynamic

Template:
    pass_info(Argument)
Mode and number of proofs:
    pass_info(?atom) - zero_or_more

Operators

op(1200,xfx,-->>)

Scope:
public

1.62 elastic_net_regression

object

1.62.1 elastic_net_regression

Elastic net regression regressor supporting continuous and mixed-feature datasets using coordinate descent with coefficient-wise L1 shrinkage plus L2 stabilization. Learns from a dataset object implementing the regression_dataset_protocol protocol and returns a regressor term that can be used for prediction and exported as predicate clauses.

Availability:
logtalk_load(elastic_net_regression(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-04

Compilation flags:
static, context_switching_calls

Imports:
public regressor_common
Uses:
format
list
numberlist
type

Remarks:
(none)

Inherited public predicates:

check_option/1 check_options/1 check_regressor/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 print_regressor/1 regressor_options/2 valid_option/1 valid_options/1 valid_regressor/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

linear_regression, ridge_regression, lasso_regression, knn_regression, regression_tree, random_forest_regression, gradient_boosting_regression

1.63 elo_ranker

object

1.63.1 elo_ranker

Elo pairwise preference ranker. Learns one deterministic rating per item from a dataset object implementing the pairwise_ranking_dataset_protocol protocol by replaying the observed preference stream using a deterministic batch Elo update rule, and returns a self-describing ranker term with diagnostics that can be used for ranking and export.

Availability:
logtalk_load(elo_ranker(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public ranking_dataset_common
public score_ranker_model_common
Uses:
avltree

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_ranker/1 default_option/1 default_options/1
diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 grouped_dataset_groups/2
grouped_dataset_items/2 grouped_dataset_items/3 grouped_dataset_summary/2 learn/2
learn/3 option/2 option/3 pairwise_dataset_connected_components/2
pairwise_dataset_declared_items/2 pairwise_dataset_items/2 pairwise_dataset_matchups/2
pairwise_dataset_preferences/2 pairwise_dataset_summary/2 pairwise_dataset_win_totals/2
pairwise_measurement_dataset_connected_components/2
pairwise_measurement_dataset_declared_items/2 pairwise_measurement_dataset_items/2
pairwise_measurement_dataset_measurements/2 pairwise_measurement_dataset_summary/2
print_ranker/1 rank/3 ranker_options/2 scores/2
temporal_pairwise_dataset_connected_components/2
temporal_pairwise_dataset_declared_items/2 temporal_pairwise_dataset_games/2
temporal_pairwise_dataset_games/3 temporal_pairwise_dataset_items/2
temporal_pairwise_dataset_periods/2 temporal_pairwise_dataset_summary/2 valid_option/1
valid_options/1 valid_ranker/1 validate_grouped_dataset/1 validate_grouped_dataset/2
validate_pairwise_dataset/1 validate_pairwise_dataset/2
validate_pairwise_measurement_dataset/1 validate_pairwise_measurement_dataset/2
validate_temporal_pairwise_dataset/1 validate_temporal_pairwise_dataset/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

pairwise_ranking_dataset_protocol, ranker_protocol, colley_ranker, rank_centrality

1.64 events

object

1.64.1 after_event_registry

After events registry predicates.

Availability:
logtalk_load(events(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2009-10-08

Compilation flags:
static, context_switching_calls, events

Implements:
public event_registryp

Remarks:
(none)

Inherited public predicates:
del_monitors/0 del_monitors/4 monitor/1 monitor/4 monitored/1 monitors/1 set_monitor/4

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

before_event_registry, monitorp

object

1.64.2 before_event_registry

Before events registry predicates.

Availability:
logtalk_load(events(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2009-10-08

Compilation flags:
static, context_switching_calls, events

Implements:
public event_registryp

Remarks:
(none)

Inherited public predicates:
del_monitors/0 del_monitors/4 monitor/1 monitor/4 monitored/1 monitors/1 set_monitor/4

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

after_event_registry, monitorp

object

1.64.3 event_registry

Before and after events registry predicates.

Availability:
logtalk_load(events(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2009-10-08

Compilation flags:
static, context_switching_calls, events

Implements:
public event_registryp

Remarks:
(none)

Inherited public predicates:
del_monitors/0 del_monitors/4 monitor/1 monitor/4 monitored/1 monitors/1 set_monitor/4

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.64.4 event_registryp

Event registry protocol.

Availability:
logtalk_load(events(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2009-10-08

Compilation flags:
static

Dependencies:
(none)

Remarks:

(none)

Inherited public predicates:
(none)

- Public predicates
  - monitors/1
  - monitor/1
  - monitored/1
  - monitor/4
  - set_monitor/4
  - del_monitors/4
  - del_monitors/0
- Protected predicates
- Private predicates
- Operators

Public predicates

monitors/1

Returns a list of all current monitors.

Compilation flags:
static

Template:
monitors(Monitors)
Mode and number of proofs:
monitors(-list(object_identifier)) - one

monitor/1

Monitor is an object playing the role of a monitor.

Compilation flags:
    static

Template:
    monitor(Monitor)
Mode and number of proofs:
    monitor(-object_identifier) - zero_or_more
    monitor(+object_identifier) - zero_or_one

---

monitored/1

Returns a list of all currently monitored objects.

Compilation flags:
    static

Template:
    monitored(Objects)
Mode and number of proofs:
    monitored(-list(object_identifier)) - one

---

monitor/4

True if the arguments describe a currently defined monitored event.

Compilation flags:
    static

Template:
    monitor(Object,Message,Sender,Monitor)
Mode and number of proofs:
    monitor(?object_identifier,?nonvar,?object_identifier,?object_identifier) - zero_or_more

---

set_monitor/4

Sets a monitor for the set of matching events.

Compilation flags:
    static

Template:
    set_monitor(Object,Message,Sender,Monitor)
Mode and number of proofs:
    set_monitor(?object_identifier,?nonvar,?object_identifier,+object_identifier) - zero_or_one

---

del_monitors/4

Deletes all matching monitored events.

Compilation flags:
    static

Template:
    del_monitors(Object,Message,Sender,Monitor)
Mode and number of proofs:
    del_monitors(?object_identifier,?nonvar,?object_identifier,?object_identifier) - one

---

del_monitors/0

Deletes all monitored events.

Compilation flags:
    static

Mode and number of proofs:
    del_monitors - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

event_registry, monitorp

category

1.64.5 monitor

Monitor predicates.

Availability:
logtalk_load(events(loader))

Author: Paulo Moura
Version: 1:3:0
Date: 2019-03-08

Compilation flags:
static, events

Implements:
public monitorp

Remarks:
(none)

Inherited public predicates:
activate_monitor/0 del_spy_points/4 monitor_activated/0 reset_monitor/0 set_spy_point/4 spy_point/4 suspend_monitor/0

- Public predicates
- Protected predicates
- Private predicates
  - spy_point_/4
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

spy_point_/4

Stores current spy points.

Compilation flags:
    dynamic

Template:
    spy_point_(Event,Object,Message,Sender)
Mode and number of proofs:
    spy_point_(?event,?object,?callable,?object) - zero_or_more

---

Operators

(none)

protocol

1.64.6 monitorp

Monitor protocol.

Availability:
logtalk_load(events(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2000-07-24

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - monitor_activated/0
  - activate_monitor/0
  - suspend_monitor/0
  - reset_monitor/0
  - spy_point/4
  - set_spy_point/4
  - del_spy_points/4
- Protected predicates
- Private predicates
- Operators

Public predicates

monitor_activated/0

True if monitor is currently active.

Compilation flags:
    static

Mode and number of proofs:
    monitor_activated - zero_or_one

---

activate_monitor/0

Activates all spy points and start monitoring.

Compilation flags:
    static

Mode and number of proofs:
    activate_monitor - one

---

suspend_monitor/0

Suspends monitoring, deactivating all spy points.

Compilation flags:
    static

Mode and number of proofs:
    suspend_monitor - one

---

reset_monitor/0

Resets monitor, deactivating and deleting all spy points.

Compilation flags:
    static

Mode and number of proofs:
    reset_monitor - one

---

spy_point/4

Current spy point.

Compilation flags:
    static

Template:
    spy_point(Event,Object,Message,Sender)
Mode and number of proofs:
    spy_point(?event,?object,?callable,?object) - zero_or_more

---

set_spy_point/4

Sets a spy point.

Compilation flags:
    static

Template:
    set_spy_point(Event,Object,Message,Sender)
Mode and number of proofs:
    set_spy_point(?event,?object,?callable,?object) - one

---

del_spy_points/4

Deletes all matching spy points.

Compilation flags:
    static

Template:
    del_spy_points(Event,Object,Message,Sender)
Mode and number of proofs:
    del_spy_points(@event,@object,@callable,@object) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

monitor, event_registryp

1.65 ewma_anomaly_detector

object

1.65.1 ewma_anomaly_detector

EWMA (Exponentially Weighted Moving Average) anomaly detector for continuous sequence-like datasets. Learns per-step population mean and standard deviation from baseline training examples selected from a dataset object implementing the anomaly_dataset_protocol protocol and returns a detector term that can be used for scoring, prediction, and export.

Availability:

```
logtalk_load(ewma_anomaly_detector(loader))
```

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public anomaly_detector_common
Uses:
format
list
pairs
population
type

Remarks:
(none)

Inherited public predicates:
anomaly_detector_options/2 check_anomaly_detector/1 check_option/1 check_options/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 predict/4 print_anomaly_detector/1 score/3 score_all/3 valid_anomaly_detector/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

anomaly_dataset_protocol, anomaly_detector_protocol, cusum_anomaly_detector, isolation_forest_anomaly_detector, knn_distance_anomaly_detector, lof_anomaly_detector, modified_z_score_anomaly_detector, z_score_anomaly_detector

1.66 expand_library_alias_paths

object

1.66.1 expand_library_alias_paths

Hook object for expanding library alias paths in logtalk_library_path/2 facts when compiling a source file.

Availability:
logtalk_load(expand_library_alias_paths(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2018-04-12

Compilation flags:
static, context_switching_calls

Implements:
public expanding
Uses:

logtalk
os

Remarks:
(none)

Inherited public predicates:
goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.67 expecteds

object

1.67.1 either

Types and predicates for extended type-checking and handling of expected terms.

Availability:
logtalk_load(expecteds(loader))

Author: Paulo Moura
Version: 0:9:0
Date: 2025-06-19

Compilation flags:
static, context_switching_calls

Provides:
type::type/1
type::check/2
arbitrary::arbitrary/1
arbitrary::arbitrary/2
arbitrary::shrinker/1
arbitrary::shrink/3
arbitrary::edge_case/2
Uses:
expected
expected(Expected)
random
type

Remarks:

- Type-checking support: Defines a either(ValueType, ErrorType) type for checking expected terms where the value and error terms must be of the given types.
- QuickCheck support: Defines clauses for the type::arbitrary/1-2, arbitrary::shrinker/1, arbitrary::shrink/3, and arbitrary::edge_case/2 predicates to allow generating random values for the either(ValueType, ErrorType) type.

Inherited public predicates:
(none)

- Public predicates

  - expecteds/2
  - unexpecteds/2
  - partition/3
  - sequence/2
  - traverse/3
- Protected predicates
- Private predicates
- Operators

Public predicates

expecteds/2

Returns the values stored in the expected terms that hold a value.

Compilation flags:
    static

Template:
    expecteds(Expecteds,Values)
Mode and number of proofs:
    expecteds(+list(expected),-list) - one

---

unexpecteds/2

Returns the errors stored in the expected terms that hold an error.

Compilation flags:
    static

Template:
    unexpecteds(Expecteds,Errors)
Mode and number of proofs:
    unexpecteds(+list(expected),-list) - one

---

partition/3

Retrieves and partitions the values and errors hold by the expected terms.

Compilation flags:
    static

Template:
    partition(Expecteds,Values,Errors)
Mode and number of proofs:
    partition(+list(expected),-list,-list) - one

---

sequence/2

Returns an expected term with a list of all values when all expected terms hold values. Otherwise returns the first expected term holding an error.

Compilation flags:
    static

Template:
    sequence(Expecteds,Expected)
Mode and number of proofs:
    sequence(+list(expected),--nonvar) - one

---

traverse/3

Applies a closure to each list element to generate expected terms and then sequences them into a single expected term holding all values or the first error.

Compilation flags:
    static

Template:
    traverse(Closure,Terms,Expected)
Meta-predicate template:
    traverse(2,*,*)

Mode and number of proofs:
traverse(+callable,+list,--nonvar) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

expected, expected(Expected), type, arbitrary

object

1.67.2 expected

Constructors for expected terms. An expected term contains either a value or an error. Expected terms should be regarded as opaque terms and always used with the expected/1 object by passing the expected term as a parameter.

Availability:
logtalk_load(expecteds(loader))

Author: Paulo Moura
Version: 2:2:0
Date: 2026-02-21

Compilation flags:
static, context_switching_calls

Provides:
type::type/1
type::check/2

Remarks:

- Type-checking support: This object also defines a type expected for use with the type library object.

Inherited public predicates:
(none)

- Public predicates
  - of_unexpected/2
  - of_expected/2
  - from_goal/4
  - from_goal/3
  - from_goal/2
  - from_generator/4
  - from_generator/3
  - from_generator/2
  - from_optional/3
- Protected predicates
- Private predicates
- Operators

Public predicates

of_unexpected/2

Constructs an expected term from an error that represent that the expected value is missing.

Compilation flags:
static

Template:
of_unexpected(Error,Expected)
Mode and number of proofs:
of_unexpected(@term,--nonvar) - one

of_expected/2

Constructs an expected term from an expected value.

Compilation flags:
    static

Template:
    of_expected(Value,Expected)
Mode and number of proofs:
    of_expected(@term,--nonvar) - one

---

from_goal/4

Constructs an expected term holding a value bound by calling the given goal. Otherwise returns an expected term with the unexpected goal error or failure represented by the Error argument.

Compilation flags:
    static

Template:
    from_goal(Goal,Value,Error,Expected)
Meta-predicate template:
    from_goal(0,*,*,*)
Mode and number of proofs:
    from_goal(+callable,--term,@term,--nonvar) - one

---

from_goal/3

Constructs an expected term holding a value bound by calling the given goal. Otherwise returns an expected term with the unexpected goal error or the atom fail representing the unexpected failure.

Compilation flags:
    static

Template:
    from_goal(Goal,Value,Expected)

Meta-predicate template:
    from_goal(0,*,*)
Mode and number of proofs:
    from_goal(+callable,--term,--nonvar) - one

---

from_goal/2

Constructs an expected term holding a value bound by calling the given closure. Otherwise returns an expected term holding the unexpected closure error or the atom fail representing the unexpected failure.

Compilation flags:
    static

Template:
    from_goal(Closure,Expected)
Meta-predicate template:
    from_goal(1,*)
Mode and number of proofs:
    from_goal(+callable,--nonvar) - one

---

from_generator/4

Constructs expected terms with the values generated by calling the given goal. On goal error or failure, returns an expected term with the unexpected goal error or failure represented by the Error argument.

Compilation flags:
    static

Template:
    from_generator(Goal,Value,Error,Expected)
Meta-predicate template:
    from_generator(0,*,*,*)
Mode and number of proofs:
    from_generator(+callable,--term,@term,--nonvar) - one_or_more

---

from_generator/3

Constructs expected terms with the values generated by calling the given goal. On goal error or failure, returns an expected term with, respectively, the unexpected goal error or the atom fail representing the unexpected goal failure.

Compilation flags:
    static

Template:
    from_generator(Goal,Value,Expected)
Meta-predicate template:
    from_generator(0,*,*)
Mode and number of proofs:
    from_generator(+callable,--term,--nonvar) - one_or_more

---

from_generator/2

Constructs expected terms with the values generated by calling the given closure. On closure error or failure, returns an expected term with, respectively, the unexpected closure error or the atom fail representing the unexpected closure failure.

Compilation flags:
    static

Template:
    from_generator(Closure,Expected)
Meta-predicate template:
    from_generator(1,*)
Mode and number of proofs:
    from_generator(+callable,--nonvar) - one_or_more

---

from_optional/3

Converts an optional term to an expected term. Returns an expected term holding the value if the optional term is not empty. Returns an expected term with the given error otherwise.

Compilation flags:
    static

Template:
    from_optional(Optional,Error,Expected)
Mode and number of proofs:
    from_optional(+nonvar,@term,--nonvar) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

expected(Expected), type

object

1.67.3 expected(Expected)

Expected term predicates. Requires passing an expected term (constructed using the expected object predicates) as a parameter.

Availability:
    logtalk_load(expecteds(loader))

Author: Paulo Moura
Version: 1:6:0

Date: 2026-02-21

Compilation flags:
static, context_switching_calls

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - is_expected/0
  - is_unexpected/0
  - if_expected/1
  - if_unexpected/1
  - if_expected_or_else/2
  - unexpected/1
  - expected/1
  - map/2
  - flat_map/2
  - either/3
  - or_else/2
  - or_else_get/2
  - or_else_call/2
  - or_else_throw/1
  - or_else_fail/1
  - filter/3
  - map_unexpected/2
  - map_catching/2
  - map_both/3
  - swap/1

  - map_or_else/3
  - or/2
  - or_else_throw/2
  - zip/3
  - to_optional/1
  - flatten/1
- Protected predicates
- Private predicates
- Operators

Public predicates

is_expected/0

True if the expected term holds a value. See also the if_expected/1 predicate.

Compilation flags:
    static

Mode and number of proofs:
    is_expected - zero_or_one

---

is_unexpected/0

True if the expected term holds an error. See also the if_unexpected/1 predicate.

Compilation flags:
    static

Mode and number of proofs:
    is_unexpected - zero_or_one

---

if_expected/1

Applies a closure when the expected term holds a value using the value as argument. Succeeds otherwise.

Compilation flags:
    static

Template:
    if_expected(Closure)
Meta-predicate template:
    if_expected(1)
Mode and number of proofs:
    if_expected(+callable) - zero_or_more

---

if_unexpected/1

Applies a closure when the expected term holds an error using the error as argument. Succeeds otherwise. Can be used to throw the exception hold by the expected term by calling it the atom throw.

Compilation flags:
    static

Template:
    if_unexpected(Closure)
Meta-predicate template:
    if_unexpected(1)
Mode and number of proofs:
    if_unexpected(+callable) - zero_or_more

---

if_expected_or_else/2

Applies either ExpectedClosure or UnexpectedClosure depending on the expected term holding a value or an error.

Compilation flags:
    static

Template:
    if_expected_or_else(ExpectedClosure,UnexpectedClosure)
Meta-predicate template:
    if_expected_or_else(1,1)
Mode and number of proofs:
    if_expected_or_else(+callable,+callable) - zero_or_more

---

unexpected/1

Returns the error hold by the expected term. Throws an error otherwise.

Compilation flags:
    static

Template:
    unexpected(Error)
Mode and number of proofs:
    unexpected(--term) - one_or_error

Exceptions:
    Expected term holds a value:
        existence_error(unexpected_error,Expected)

---

expected/1

Returns the value hold by the expected term. Throws an error otherwise.

Compilation flags:
    static

Template:
    expected(Value)
Mode and number of proofs:
    expected(--term) - one_or_error

Exceptions:
    Expected term holds an error:

existence_error(expected_value,Expected)

---

map/2

When the expected term does not hold an error and mapping a closure with the expected value and the new value as additional arguments is successful, returns an expected term with the new value. Otherwise returns the same expected term.

Compilation flags:
    static

Template:
    map(Closure,NewExpected)
Meta-predicate template:
    map(2,*)
Mode and number of proofs:
    map(+callable,--nonvar) - one

---

flat_map/2

When the expected term does not hold an error and mapping a closure with the expected value and the new expected term as additional arguments is successful, returns the new expected term. Otherwise returns the same expected term.

Compilation flags:
    static

Template:
    flat_map(Closure,NewExpected)
Meta-predicate template:
    flat_map(2,*)
Mode and number of proofs:
    flat_map(+callable,--nonvar) - one

---

either/3

Applies either ExpectedClosure if the expected term holds a value or UnexpectedClosure if the expected term holds an error. Returns a new expected term if the applied closure is successful. Otherwise returns the same expected term.

Compilation flags:
    static

Template:
    either(ExpectedClosure,UnexpectedClosure,NewExpected)
Meta-predicate template:
    either(2,2,*)
Mode and number of proofs:
    either(+callable,+callable,--nonvar) - one

---

or_else/2

Returns the value hold by the expected term if it does not hold an error or the given default term if the expected term holds an error.

Compilation flags:
    static

Template:
    or_else(Value,Default)
Mode and number of proofs:
    or_else(--term,@term) - one

---

or_else_get/2

Returns the value hold by the expected term if it does not hold an error. Otherwise applies a closure to compute the expected value. Throws an error when the expected term holds an error and a value cannot be computed.

Compilation flags:
    static

Template:
    or_else_get(Value,Closure)
Meta-predicate template:
    or_else_get(*,1)
Mode and number of proofs:
    or_else_get(--term,+callable) - one_or_error

Exceptions:
    Expected term holds an unexpected error and an expected value cannot be computed:
        existence_error(expected_value,Expected)

---

or_else_call/2

Returns the value hold by the expected term if it does not hold an error. Calls a goal deterministically otherwise.

Compilation flags:
    static

Template:
    or_else_call(Value,Goal)
Meta-predicate template:
    or_else_call(*,0)
Mode and number of proofs:
    or_else_call(--term,+callable) - zero_or_one

---

or_else_throw/1

Returns the value hold by the expected term if present. Throws the error hold by the expected term as an exception otherwise.

Compilation flags:
    static

Template:
    or_else_throw(Value)
Mode and number of proofs:
    or_else_throw(--term) - one_or_error

---

or_else_fail/1

Returns the value hold by the expected term if it does not hold an error. Fails otherwise. Usually called to skip over expected terms holding errors.

Compilation flags:
    static

Template:
    or_else_fail(Value)
Mode and number of proofs:
    or_else_fail(--term) - zero_or_one

---

filter/3

When the expected term holds a value and the value satisfies the closure, returns the same expected term. When the expected term holds a value that does not satisfy the closure, returns an expected term with the given error. When the expected term holds an error, returns the same expected term.

Compilation flags:
    static

Template:
    filter(Closure,Error,NewExpected)
Meta-predicate template:
    filter(1,*,*)
Mode and number of proofs:
    filter(+callable,@term,--nonvar) - one

---

map_unexpected/2

When the expected term holds an error and mapping a closure with the error and the new error as additional arguments is successful, returns an expected term with the new error. Otherwise returns the same expected term.

Compilation flags:
    static

Template:
    map_unexpected(Closure,NewExpected)
Meta-predicate template:
    map_unexpected(2,*)
Mode and number of proofs:
    map_unexpected(+callable,--nonvar) - one

---

map_catching/2

When the expected term holds a value, applies a closure to it. Returns an expected term with the new value if the closure succeeds. Returns an expected term with the error if the closure throws an error. Returns an expected term with the atom fail as error if the closure fails. When the expected term holds an error, returns the same expected term.

Compilation flags:
    static

Template:
    map_catching(Closure,NewExpected)
Meta-predicate template:
    map_catching(2,*)
Mode and number of proofs:
    map_catching(+callable,--nonvar) - one

---

map_both/3

When the expected term holds a value and mapping ExpectedClosure with the value is successful, returns an expected term with the new value. When the expected term holds an error and mapping UnexpectedClosure with the error is successful, returns an expected term with the new error. Otherwise returns the same expected term.

Compilation flags:
    static

Template:
    map_both(ExpectedClosure,UnexpectedClosure,NewExpected)
Meta-predicate template:
    map_both(2,2,*)
Mode and number of proofs:
    map_both(+callable,+callable,--nonvar) - one

---

swap/1

Swaps the expected and unexpected terms. If the expected term holds a value, returns an unexpected term with that value. If the expected term holds an error, returns an expected term with that error.

Compilation flags:
    static

Template:
    swap(NewExpected)
Mode and number of proofs:
    swap(--nonvar) - one

---

map_or_else/3

When the expected term holds a value and mapping a closure with the value and the new value as additional arguments is successful, returns the new value. Otherwise returns the given default value.

Compilation flags:
    static

Template:
map_or_else(Closure,Default,Value)
Meta-predicate template:
map_or_else(2,*,*)
Mode and number of proofs:
map_or_else(+callable,@term,--term) - one

---

or/2

Returns the same expected term if it holds a value. Otherwise calls closure to generate a new expected term. Fails if the expected term holds an error and calling the closure fails or throws an error.

Compilation flags:
static

Template:
or(NewExpected,Closure)
Meta-predicate template:
or(*,1)
Mode and number of proofs:
or(--term,@callable) - zero_or_one

---

or_else_throw/2

Returns the value hold by the expected term if present. Throws the given error otherwise, ignoring any error hold by the expected term.

Compilation flags:
static

Template:
or_else_throw(Value,Error)
Mode and number of proofs:
or_else_throw(--term,@nonvar) - one_or_error

---

zip/3

When both this expected and the other expected hold values and applying a closure with both values and the new value as additional arguments is successful, returns an expected term with the new value. Otherwise returns the first expected term that holds an error.

Compilation flags:
    static

Template:
    zip(Closure,OtherExpected,NewExpected)
Meta-predicate template:
    zip(3,*,*)
Mode and number of proofs:
    zip(+callable,+nonvar,--nonvar) - one

---

to_optional/1

Converts the expected term to an optional term. Returns a non-empty optional term holding the value if the expected term holds a value. Returns an empty optional term if the expected term holds an error.

Compilation flags:
    static

Template:
    to_optional(Optional)
Mode and number of proofs:
    to_optional(--nonvar) - one

---

flatten/1

Flattens a nested expected term. When the expected term holds a value that is itself an expected term, returns the inner expected term. When the expected term holds a non-expected value, returns the same expected term. When the expected term holds an error, returns the same expected term.

Compilation flags:
    static

Template:
    flatten(NewExpected)
Mode and number of proofs:
    flatten(--nonvar) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

expected

1.68 fcube

object

1.68.1 fcube

FCube: An Efficient Prover for Intuitionistic Propositional Logic.

Availability:
    logtalk_load(fcube(loader))

Author: Mauro Ferrari, Camillo Fiorentini, Guido Fiorino; ported to Logtalk by Paulo Moura.
Version: 5:0:1
Date: 2024-03-14

Copyright: Copyright 2012 Mauro Ferrari, Camillo Fiorentini, Guido Fiorino; Copyright 2020-2024 Paulo Moura
License: GPL-2.0-or-later

Compilation flags:
static, context_switching_calls

Uses:
integer
list
os
set
user

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - gnu/0
  - fcube/0
  - decide/1
  - decide/2
- Protected predicates
- Private predicates
- Operators
  - op(1200,xfy,<=>)
  - op(1110,xfy,=>)
  - op(1000,xfy,&&)
  - op(500,fy,~)
  - op(1100,xfy,v)

Public predicates

gnu/0

Prints banner with copyright and license information.

Compilation flags:
    static

Mode and number of proofs:
    gnu - one

---

fcube/0

Reads a formula and applies the prover to it, printing its counter-model.

Compilation flags:
    static

Mode and number of proofs:
    fcube - one

---

decide/1

Applies the prover to the given formula and prints its counter-model.

Compilation flags:
    static

Template:
    decide(Formula)
Mode and number of proofs:
    decide(++compound) - one

---

decide/2

Applies the prover to the given formula and returns its counter-model.

Compilation flags:
    static

Template:
    decide(Formula,CounterModel)
Mode and number of proofs:
    decide(++compound,--compound) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

op(1200,xfy,<=>)

Scope:
    public

op(1110,xfy,=>)

Scope:
    public

op(1000,xfy,&&)

Scope:
    public

op(500,fy,~)

Scope:
public

op(1100,xfy,v)

Scope:
public

1.69 flags

category

1.69.1 flags

Implementation of persistent object flags.

Availability:
logtalk_load(flags(loader))

Author: Theofrastos Mantadelis
Version: 1:0:0
Date: 2010-11-27

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - get_flag_value/2
  - set_flag_value/2
  - set_flag_value/3
  - reset_flags/0
  - reset_flags/1
  - flag_groups/1
  - flag_group_chk/1
  - print_flags/0
  - print_flags/1
  - defined_flag/6
  - built_in_flag/2
- Protected predicates
  - unsafe_set_flag_value/2
  - define_flag/1
  - define_flag/2
- Private predicates
  - defined_flag_/6
  - flag_value_/2
  - validate/3
  - validate_type/1
  - is_validator/1
- Operators

Public predicates

get_flag_value/2

Gets or tests the value of a flag.

Compilation flags:
    static

Template:
    get_flag_value(Flag,Value)
Mode and number of proofs:
    get_flag_value(+atom,?nonvar) - zero_or_one

---

set_flag_value/2

Sets the value of a flag.

Compilation flags:
    static

Template:
    set_flag_value(Flag,NewValue)
Mode and number of proofs:
    set_flag_value(+atom,@nonvar) - one

---

set_flag_value/3

Sets the value of a flag, returning the old value.

Compilation flags:
    static

Template:
    set_flag_value(Flag,OldValue,NewValue)
Mode and number of proofs:
    set_flag_value(+atom,?nonvar,@nonvar) - one

---

reset_flags/0

Resets all flags to their default values.

Compilation flags:
    static

Mode and number of proofs:
    reset_flags - one

---

reset_flags/1

Resets all flags in a group to their default values.

Compilation flags:
    static

Template:
    reset_flags(Group)
Mode and number of proofs:
    reset_flags(+atom) - one

---

flag_groups/1

Returns a list of all flag groups.

Compilation flags:
    static

Template:
    flag_groups(Groups)
Mode and number of proofs:
    flag_groups(-list(atom)) - one

---

flag_group_chk/1

Checks if a given atom is a flag group.

Compilation flags:
    static

Template:
    flag_group_chk(Group)
Mode and number of proofs:

flag_group_chk(+atom) - zero_or_one

---

print_flags/0

Prints a listing of all flags.

Compilation flags:
static

Mode and number of proofs:
print_flags - one

---

print_flags/1

Prints a listing of all flags in a group.

Compilation flags:
static

Template:
print_flags(Group)
Mode and number of proofs:
print_flags(+atom) - one

---

defined_flag/6

Gets or test the existing (visible) flag definitions.

Compilation flags:
static

Template:
defined_flag(Flag,Group,Type,DefaultValue,Description,Access)

Mode and number of proofs:
    defined_flag(?atom,?atom,?nonvar,?nonvar,?atom,?atom) - zero_or_more

---

built_in_flag/2

True if the argument is a built-in flag type with the specified default value.

Compilation flags:
    static

Template:
    built_in_flag(Type,DefaultValue)
Mode and number of proofs:
    built_in_flag(?atom,?nonvar) - zero_or_more

Protected predicates

unsafe_set_flag_value/2

Sets the value of a flag without performing any validation checks.

Compilation flags:
    static

Template:
    unsafe_set_flag_value(Flag,NewValue)
Mode and number of proofs:
    unsafe_set_flag_value(+atom,@nonvar) - one

---

define_flag/1

Defines a new flag using default options.

Compilation flags:
    static

Template:
    define_flag(Flag)
Mode and number of proofs:
    define_flag(+atom) - one

---

define_flag/2

Defines a new flag using a given set of options (for example, [group(general), type(nonvar), default(true), description(Flag), access(read_write)]).

Compilation flags:
    static

Template:
    define_flag(Flag,Options)
Mode and number of proofs:
    define_flag(+atom,@list) - one

Private predicates

defined_flag_/6

Gets or test the existing flag definitions.

Compilation flags:
    dynamic

Template:
    defined_flag_(Flag,Group,Type,DefaultValue,Description,Access)
Mode and number of proofs:

defined_flag_(?atom,?atom,?nonvar,?nonvar,?atom,?atom) - zero_or_more

---

flag_value_/2

Table of flag values.

Compilation flags:
dynamic

Template:
flag_value_(Flag,Value)
Mode and number of proofs:
flag_value_(?atom,?nonvar) - zero_or_more

---

validate/3

Compilation flags:
static

---

validate_type/1

Compilation flags:
static

---

is_validator/1

Compilation flags:
static

Operators

(none)

protocol

1.69.2 flags_validator

Flag validation protocol. Must be implemented by validator objects.

Availability:
logtalk_load(flags(loader))

Author: Theofrastos Mantadelis
Version: 1:0:0
Date: 2010-11-27

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - print_flags/0
  - validate/1
- Protected predicates
- Private predicates
- Operators

Public predicates

print_flags/0

Validates the validator object itself.

Compilation flags:
    static

Mode and number of proofs:
    print_flags - zero_or_one

---

validate/1

Validates a flag value.

Compilation flags:
    static

Template:
    validate(Value)
Mode and number of proofs:
    validate(@term) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.70 format

object

1.70.1 format

Formatted output predicates.

Availability:
logtalk_load(format(loader))

Author: Paulo Moura
Version: 1:3:0
Date: 2025-11-18

Compilation flags:
static, context_switching_calls

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - format/3
  - format/2
- Protected predicates
- Private predicates
- Operators

Public predicates

format/3

Writes a list of arguments after a format specification to the specified output stream.

Compilation flags:
    static

Template:
    format(Stream,Format,Arguments)
Mode and number of proofs:
    format(@stream_or_alias,+atom,@list) - zero_or_one
    format(@stream_or_alias,+list(character_code),@list) - zero_or_one
    format(@stream_or_alias,+list(character),@list) - zero_or_one

---

format/2

Writes a list of arguments after a format specification to the current output stream.

Compilation flags:
    static

Template:
    format(Format,Arguments)
Mode and number of proofs:
    format(+atom,@list) - zero_or_one
    format(+list(character_code),@list) - zero_or_one
    format(+list(character),@list) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.71 fp_growth_pattern_miner

object

1.71.1 fp_growth_pattern_miner

FP-growth frequent itemset miner for transaction datasets using recursive conditional pattern-base projection over an FP-tree with header-table and parent-link navigation.

Availability:
logtalk_load(fp_growth_pattern_miner(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public frequent_pattern_mining_common
Uses:
format
list

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_pattern_miner/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 mine/2 mine/3 option/2 option/3 pattern_miner_options/2 print_pattern_miner/1 valid_option/1 valid_options/1 valid_pattern_miner/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

pattern_miner_protocol, transaction_dataset_protocol, apriori_pattern_miner

1.72 frequent_pattern_mining_protocols

category

1.72.1 frequent_pattern_mining_common

Shared predicates for frequent itemset miner dataset validation, support accumulation, and itemset ordering/filtering helpers.

Availability:
logtalk_load(frequent_pattern_mining_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-29

Compilation flags:
static

Extends:
public pattern_miner_common
Uses:
list
type

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_pattern_miner/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 mine/2 mine/3 option/2 option/3 pattern_miner_options/2 print_pattern_miner/1 valid_option/1 valid_options/1 valid_pattern_miner/1

- Public predicates
- Protected predicates
  - check_transactions/4
  - sort_patterns/2
  - filter_patterns/3
  - count_items/4
  - select_frequent_item_supports/3
  - valid_itemset_patterns/2
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

check_transactions/4

Validates the collected dataset transactions and returns the maximum transaction length.

Compilation flags:
    static

Template:
    check_transactions(Dataset,ItemDomain,Transactions,MaxTransactionLength)
Mode and number of proofs:
    check_transactions(+object_identifier,+list(atom),+list(pair(integer,list(atom))),-integer) - one

---

sort_patterns/2

Sorts itemset patterns first by pattern length and then lexicographically.

Compilation flags:
    static

Template:
    sort_patterns(Patterns0,Patterns)
Mode and number of proofs:
    sort_patterns(+list(compound),-list(compound)) - one

---

filter_patterns/3

Filters itemset patterns by minimum pattern length.

Compilation flags:
    static

Template:
filter_patterns(Patterns0,MinimumPatternLength,Patterns)
Mode and number of proofs:
filter_patterns(+list(compound),+integer,-list(compound)) - one

---

count_items/4

Accumulates support counts for a list of items.

Compilation flags:
static

Template:
count_items(Items,Count,ItemCounts0,ItemCounts)
Mode and number of proofs:
count_items(+list(atom),+integer,+list(pair(atom,integer)),-list(pair(atom,integer))) - one

---

select_frequent_item_supports/3

Selects the item supports that satisfy the minimum support count.

Compilation flags:
static

Template:
select_frequent_item_supports(ItemCounts,SupportCount,FrequentItemSupports)
Mode and number of proofs:
select_frequent_item_supports(+list(pair(atom,integer)),+integer,-list(compound)) - one

---

valid_itemset_patterns/2

True when the patterns are valid itemset(Items, Support) terms over the given item domain with positive integer supports.

Compilation flags:
static

Template:
valid_itemset_patterns(ItemDomain,Patterns)
Mode and number of proofs:
valid_itemset_patterns(+list(atom),+list(compound)) - zero_or_one

---

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.72.2 transaction_dataset_protocol

Protocol for transaction datasets used with frequent itemset mining algorithms. Transaction identifiers are expected to be unique within a dataset.

Availability:
logtalk_load(frequent_pattern_mining_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-29

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - items/1
  - transaction/2
- Protected predicates
- Private predicates
- Operators

Public predicates

items/1

Returns the declared transaction item domain as a canonical sorted list of unique items.

Compilation flags:
static

Template:
items(Items)
Mode and number of proofs:
items(-list(atom)) - one

---

transaction/2

Enumerates by backtracking the transactions in the dataset. The transaction identifier is expected to be unique within the dataset. Each transaction is represented as a canonical sorted list of unique declared items.

Compilation flags:
static

Template:

transaction(Id,Transaction)
Mode and number of proofs:
transaction(-integer,-list(atom)) - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.73 gaussian_mixture_clusterer

object

1.73.1 gaussian_mixture_clusterer

Gaussian mixture model clusterer for continuous datasets. Learns from a dataset object implementing the clustering_dataset_protocol protocol and returns a clusterer term that can be used for assigning new instances to clusters and exported as predicate clauses.

Availability:
logtalk_load(gaussian_mixture_clusterer(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public clusterer_common
Uses:
format

list
numberlist
pairs
population
type

Remarks:
(none)

Inherited public predicates:
check_clusterer/1 check_option/1 check_options/1 cluster/3 clusterer_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 print_clusterer/1 valid_clusterer/1 valid_option/1 valid_options/1

- Public predicates
  - cluster_probabilities/3
- Protected predicates
- Private predicates
- Operators

Public predicates

cluster_probabilities/3

Returns posterior component probabilities for a new instance as Cluster-Probability pairs in component-id order.

Compilation flags:
static

Template:
cluster_probabilities(Clusterer,Instance,Probabilities)
Mode and number of proofs:
cluster_probabilities(+compound,+list,-list(pair)) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

clusterer_protocol, clustering_dataset_protocol, kmeans_clusterer

1.74 gaussian_process_regression

object

1.74.1 gaussian_process_regression

Gaussian process regression regressor supporting continuous and mixed-feature datasets using an exact mixed Gaussian process with posterior uncertainty estimates. Learns from a dataset object implementing the regression_dataset_protocol protocol and returns a regressor term that can be used for prediction, predictive-distribution queries, and export as predicate clauses.

Availability:
logtalk_load(gaussian_process_regression(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:
static, context_switching_calls

Imports:
public regressor_common
Uses:
format
linear_algebra
list

numberlist
population
type

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_regressor/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 print_regressor/1 regressor_options/2 valid_option/1 valid_options/1 valid_regressor/1

- Public predicates
  - predict_distribution/3
- Protected predicates
- Private predicates
- Operators

Public predicates

predict_distribution/3

Predicts the posterior predictive Gaussian distribution for a new instance using the learned regressor. The returned term has the shape gaussian(Mean, Variance) where Variance includes the learned observation noise variance.

Compilation flags:
static

Template:
predict_distribution(Regressor,Instance,Distribution)
Mode and number of proofs:
predict_distribution(+compound,+list,-compound) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

linear_regression, ridge_regression, lasso_regression, elastic_net_regression, regression_tree, random_forest_regression

1.75 genint

object

1.75.1 genint

Global object for generating increasing non-negative integers for named counters. The predicates are declared as synchronized when the library is compiled using a backend supporting threads.

Availability:
logtalk_load(genint(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2022-07-21

Compilation flags:
static, context_switching_calls

Imports:
public genint_core

Remarks:
(none)

Inherited public predicates:
genint/2 reset_genint/0 reset_genint/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.75.2 genint_core

Predicates for generating increasing non-negative integers. The predicates are declared as synchronized when the library is compiled using a backend supporting threads.

Availability:
logtalk_load(genint(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2022-07-26

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - reset_genint/0
  - reset_genint/1
  - genint/2
- Protected predicates
- Private predicates
  - counter_/2
- Operators

Public predicates

reset_genint/0

Resets all counters.

Compilation flags:
static, synchronized

Mode and number of proofs:
reset_genint - one

reset_genint/1

Resets the given counter.

Compilation flags:
    static, synchronized

Template:
    reset_genint(Counter)
Mode and number of proofs:
    reset_genint(+atom) - one

---

genint/2

Returns the next integer for a given counter.

Compilation flags:
    static, synchronized

Template:
    genint(Counter,Integer)
Mode and number of proofs:
    genint(+atom,-non_negative_integer) - one

Protected predicates

(none)

Private predicates

counter_/2

Table of current state of counters.

Compilation flags:
    dynamic

Template:
counter_(Counter,Latest)
Mode and number of proofs:
counter_(?atom,?non_negative_integer) - zero_or_more

Operators

(none)

1.76 gensym

object

1.76.1 gensym

Global object for generating unique atoms. The predicates are declared as synchronized when the library is compiled using a backend supporting threads.

Availability:
logtalk_load(gensym(loader))

Author: Paulo Moura
Version: 2:0:0
Date: 2022-07-21

Compilation flags:
static, context_switching_calls

Imports:
public gensym_core

Remarks:
(none)

Inherited public predicates:
gensym/2 reset_gensym/0 reset_gensym/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.76.2 gensym_core

Predicates for generating unique atoms. Protocol based on the gensym module of SWI-Prolog. The predicates are declared as synchronized when the library is compiled using a backend supporting threads.

Availability:
logtalk_load(gensym(loader))

Author: Paulo Moura
Version: 2:1:0
Date: 2022-07-26

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - reset_gensym/0
  - reset_gensym/1
  - gensym/2
- Protected predicates
- Private predicates
  - base_/2
- Operators

Public predicates

reset_gensym/0

Resets the generator counter for all bases.

Compilation flags:
static, synchronized

Mode and number of proofs:
reset_gensym - one

---

reset_gensym/1

Resets the generator counter for a given base.

Compilation flags:
static, synchronized

Template:
reset_gensym(Base)

Mode and number of proofs:
reset_gensym(+atom) - one

---

gensym/2

Returns a new unique atom with a given base (prefix).

Compilation flags:
static, synchronized

Template:
gensym(Base,Unique)
Mode and number of proofs:
gensym(+atom,-atom) - one

Protected predicates

(none)

Private predicates

base_/2

Table of generator bases and respective counters.

Compilation flags:
dynamic

Template:
base_(Base,Counter)
Mode and number of proofs:
base_(?atom,?integer) - zero_or_more

Operators

(none)

1.77 geohash

object

1.77.1 geohash

Geohash encoder, decoder, adjacency, covering, hierarchy, integer conversion, bit-precision, and geometry covering predicates for geographic(Latitude,Longitude) coordinates.

Availability:
logtalk_load(geohash(loader))

Author: Paulo Moura
Version: 1:2:0
Date: 2026-05-11

Compilation flags:
static, context_switching_calls

Implements:
public geohash_protocol
Imports:
public options
Uses:
geospatial
list
type

Remarks:

- Alphabet: Uses the standard geohash base-32 alphabet 0123456789bcdefghjkmnpqrstuvwxyz.
- Encoding canonicalization: Encoding canonicalizes longitudes to the [-180.0,180.0[range, treating 180.0 as -180.0 so equivalent dateline coordinates map to the same geohash.
- Bounding boxes: Decoded cells are represented using bbox(geographic(MinLatitude,MinLongitude), geographic(MaxLatitude,MaxLongitude)).
- Coverage: Bounding-box coverage accepts antimeridian-crossing boxes where the minimum longitude is greater than the maximum longitude.

- Adaptive covering: Adaptive covers use max_precision/1 together with compact/1 and min_precision/1 options.
- Integer representation: Integer geohashes are 5-bit packed base-32 values with an explicit precision argument preserving leading zero groups.
- Geometry covering: Polygon and polyline covering predicates rely on generic geospatial geometry predicates and do not support antimeridian-crossing input geometries.

Inherited public predicates:

adjacent/3 adjacent_bits/4 adjacent_int/4 bounding_box/2 bounding_box_bits/3 bounding_box_int/3 cell_dimensions/3 check_option/1 check_options/1 children/2 common_prefix/3 compress/2 covering/3 covering/4 decode/2 decode_bits/3 decode_int/3 default_option/1 default_options/1 encode/3 encode_bits/3 encode_int/3 expand/2 expand_int/3 geohash_to_int/2 int_to_geohash/3 neighbors/2 neighbors_bits/3 neighbors_int/3 option/2 option/3 parent/2 polygon_covering/4 polyline_covering/4 precision/3 valid_geohash/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

geohash_protocol, geospatial

protocol

1.77.2 geohash_protocol

Geohash predicates protocol.

Availability:
logtalk_load(geohash(loader))

Author: Paulo Moura
Version: 1:2:0
Date: 2026-05-11

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - valid_geohash/1
  - encode/3
  - decode/2
  - bounding_box/2
  - precision/3
  - cell_dimensions/3
  - adjacent/3
  - neighbors/2
  - covering/3
  - covering/4
  - compress/2
  - parent/2
  - children/2

  - common_prefix/3
  - encode_int/3
  - decode_int/3
  - bounding_box_int/3
  - adjacent_int/4
  - neighbors_int/3
  - geohash_to_int/2
  - int_to_geohash/3
  - encode_bits/3
  - decode_bits/3
  - bounding_box_bits/3
  - adjacent_bits/4
  - neighbors_bits/3
  - expand/2
  - expand_int/3
  - polygon_covering/4
  - polyline_covering/4
- Protected predicates
- Private predicates
- Operators

Public predicates

valid_geohash/1

True when the argument is a non-empty geohash atom using the standard geohash base-32 alphabet 0123456789bcdefghjkmnpqrstuvwxyz.

Compilation flags:
    static

Template:
    valid_geohash(Geohash)
Mode and number of proofs:
    valid_geohash(+atom) - zero_or_one

encode/3

Encodes a geographic coordinate represented as geographic(Latitude,Longitude) into a geohash atom with the given precision in characters. Longitude is canonicalized to the [-180.0,180.0[range, so 180.0 is treated as -180.0.

Compilation flags:
    static

Template:
    encode(Coordinate,Precision,Geohash)
Mode and number of proofs:
    encode(+compound,+positive_integer,-atom) - zero_or_one

---

decode/2

Decodes a geohash atom into the center geographic coordinate of the represented cell.

Compilation flags:
    static

Template:
    decode(Geohash,Coordinate)
Mode and number of proofs:
    decode(+atom,-compound) - zero_or_one

---

bounding_box/2

Decodes a geohash atom into its geographic bounding box represented as bbox(geographic(MinLatitude, MinLongitude),geographic(MaxLatitude,MaxLongitude)).

Compilation flags:
    static

Template:
    bounding_box(Geohash,BoundingBox)
Mode and number of proofs:

bounding_box(+atom,-compound) - zero_or_one

---

precision/3

Returns the maximum latitude and longitude errors in degrees for geohashes of the given precision in characters.

Compilation flags:
static

Template:
precision(Precision,LatitudeError,LongitudeError)
Mode and number of proofs:
precision(+positive_integer,-float,-float) - zero_or_one

---

cell_dimensions/3

Returns the latitude and longitude spans in degrees of a geohash cell for the given precision in characters.

Compilation flags:
static

Template:
cell_dimensions(Precision,LatitudeSpan,LongitudeSpan)
Mode and number of proofs:
cell_dimensions(+positive_integer,-float,-float) - zero_or_one

---

adjacent/3

Returns the adjacent geohash in the given direction. Supported directions are north, south, east, west, north_east, north_west, south_east, and south_west. Eastward and westward adjacency wraps across the antimeridian using the same longitude canonicalization as encode/3. Northward and southward adjacency fails when there is no neighboring cell beyond the poles.

Compilation flags:

static

Template:
adjacent(Geohash,Direction,AdjacentGeohash)
Mode and number of proofs:
adjacent(+atom,+atom,-atom) - zero_or_one

---

neighbors/2

Returns the available neighboring geohashes as Direction-Geohash pairs in clockwise order starting at north.

Compilation flags:
static

Template:
neighbors(Geohash,Neighbors)
Mode and number of proofs:
neighbors(+atom,-list(compound)) - zero_or_one

---

covering/3

Returns the list of geohashes at the given precision whose cells cover a geographic bounding box represented as bbox(geographic(MinLatitude,MinLongitude),geographic(MaxLatitude,MaxLongitude)). Degenerate point bounding boxes and exact cell-aligned boundaries are accepted. Antimeridian-crossing bounding boxes are accepted.

Compilation flags:
static

Template:
covering(BoundingBox,Precision,Geohashes)
Mode and number of proofs:
covering(+compound,+positive_integer,-list(atom)) - zero_or_one

---

covering/4

Returns a geohash cover for a geographic bounding box. CoverSpec is either precision(Precision) for a fixed-depth cover or max_precision(MaxPrecision) for an adaptive mixed-depth cover. Options may include compact(Boolean) and min_precision(PositiveInteger).

Compilation flags:
    static

Template:
    covering(BoundingBox,CoverSpec,Geohashes,Options)
Mode and number of proofs:
    covering(+compound,+compound,-list(atom),++list(compound)) - zero_or_one

---

compress/2

Compresses a list of geohashes by replacing complete sibling sets with their parent prefix while preserving the covered area.

Compilation flags:
    static

Template:
    compress(Geohashes,CompressedGeohashes)
Mode and number of proofs:
    compress(+list(atom),-list(atom)) - zero_or_one

---

parent/2

Returns the immediate parent prefix of a geohash. Fails for precision-1 geohashes.

Compilation flags:
    static

Template:
    parent(Geohash,Parent)
Mode and number of proofs:

parent(+atom,-atom) - zero_or_one

---

children/2

Returns the 32 child geohashes of a geohash in standard geohash base-32 order.

Compilation flags:
static

Template:
children(Geohash,Children)
Mode and number of proofs:
children(+atom,-list(atom)) - zero_or_one

---

common_prefix/3

Returns the longest common geohash prefix shared by two geohashes. The prefix may be the empty atom when the geohashes share no leading characters.

Compilation flags:
static

Template:
common_prefix(Geohash1,Geohash2,Prefix)
Mode and number of proofs:
common_prefix(+atom,+atom,-atom) - zero_or_one

---

encode_int/3

Encodes a geographic coordinate into an integer geohash representation at the given precision in characters.

Compilation flags:
static

Template:
encode_int(Coordinate,Precision,HashInteger)
Mode and number of proofs:
encode_int(+compound,+positive_integer,-integer) - zero_or_one

---

decode_int/3

Decodes an integer geohash representation at the given precision into the center geographic coordinate of the represented cell.

Compilation flags:
static

Template:
decode_int(HashInteger,Precision,Coordinate)
Mode and number of proofs:
decode_int(+integer,+positive_integer,-compound) - zero_or_one

---

bounding_box_int/3

Decodes an integer geohash representation at the given precision into its geographic bounding box.

Compilation flags:
static

Template:
bounding_box_int(HashInteger,Precision,BoundingBox)
Mode and number of proofs:
bounding_box_int(+integer,+positive_integer,-compound) - zero_or_one

---

adjacent_int/4

Returns the adjacent integer geohash at the same precision in the given direction.

Compilation flags:
    static

Template:
    adjacent_int(HashInteger,Precision,Direction,AdjacentHashInteger)
Mode and number of proofs:
    adjacent_int(+integer,+positive_integer,+atom,-integer) - zero_or_one

---

neighbors_int/3

Returns the available neighboring integer geohashes as Direction-HashInteger pairs in clockwise order starting at north.

Compilation flags:
    static

Template:
    neighbors_int(HashInteger,Precision,Neighbors)
Mode and number of proofs:
    neighbors_int(+integer,+positive_integer,-list(compound)) - zero_or_one

---

geohash_to_int/2

Converts a geohash atom to its packed integer representation using five bits per character.

Compilation flags:
    static

Template:
    geohash_to_int(Geohash,HashInteger)
Mode and number of proofs:
    geohash_to_int(+atom,-integer) - zero_or_one

---

int_to_geohash/3

Converts a packed integer geohash representation and an explicit precision in characters to a geohash atom.

Compilation flags:
    static

Template:
    int_to_geohash(HashInteger,Precision,Geohash)
Mode and number of proofs:
    int_to_geohash(+integer,+positive_integer,-atom) - zero_or_one

---

encode_bits/3

Encodes a geographic coordinate into an integer geohash representation using the given bit precision.

Compilation flags:
    static

Template:
    encode_bits(Coordinate,Bits,HashInteger)
Mode and number of proofs:
    encode_bits(+compound,+positive_integer,-integer) - zero_or_one

---

decode_bits/3

Decodes an integer geohash representation at the given bit precision into the center geographic coordinate of the represented cell.

Compilation flags:
    static

Template:
    decode_bits(HashInteger,Bits,Coordinate)

Mode and number of proofs:
decode_bits(+integer,+positive_integer,-compound) - zero_or_one

---

bounding_box_bits/3

Decodes an integer geohash representation at the given bit precision into its geographic bounding box.

Compilation flags:
static

Template:
bounding_box_bits(HashInteger,Bits,BoundingBox)
Mode and number of proofs:
bounding_box_bits(+integer,+positive_integer,-compound) - zero_or_one

---

adjacent_bits/4

Returns the adjacent integer geohash at the same bit precision in the given direction.

Compilation flags:
static

Template:
adjacent_bits(HashInteger,Bits,Direction,AdjacentHashInteger)
Mode and number of proofs:
adjacent_bits(+integer,+positive_integer,+atom,-integer) - zero_or_one

---

neighbors_bits/3

Returns the available neighboring integer geohashes as Direction-HashInteger pairs in clockwise order starting at north for the given bit precision.

Compilation flags:
static

Template:
neighbors_bits(HashInteger,Bits,Neighbors)
Mode and number of proofs:
neighbors_bits(+integer,+positive_integer,-list(compound)) - zero_or_one

---

expand/2

Returns the geohash followed by its available neighbors in clockwise order starting at north.

Compilation flags:
static

Template:
expand(Geohash,ExpandedGeohashes)
Mode and number of proofs:
expand(+atom,-list(atom)) - zero_or_one

---

expand_int/3

Returns the integer geohash followed by its available neighboring integer geohashes in clockwise order starting at north.

Compilation flags:
static

Template:
expand_int(HashInteger,Precision,ExpandedHashes)
Mode and number of proofs:
expand_int(+integer,+positive_integer,-list(integer)) - zero_or_one

---

polygon_covering/4

Returns a geohash cover for a polygon represented as a list of geographic coordinates. CoverSpec is either precision(Precision) for a fixed-depth cover or max_precision(MaxPrecision) for an adaptive mixed-depth cover. Options may include compact(Boolean) and min_precision(PositiveInteger). Antimeridian-crossing polygons are not supported.

Compilation flags:
    static

Template:
    polygon_covering(Polygon,CoverSpec,Geohashes,Options)
Mode and number of proofs:
    polygon_covering(+list(compound),+compound,-list(atom),++list(compound)) - zero_or_one

---

polyline_covering/4

Returns a geohash cover for a polyline represented as a list of two or more geographic coordinates. CoverSpec is either precision(Precision) for a fixed-depth cover or max_precision(MaxPrecision) for an adaptive mixed-depth cover. Options may include compact(Boolean), min_precision(PositiveInteger), and buffer(Distance) where the buffer distance is given in kilometers. Antimeridian-crossing polylines are not supported.

Compilation flags:
    static

Template:
    polyline_covering(Polyline,CoverSpec,Geohashes,Options)
Mode and number of proofs:
    polyline_covering(+list(compound),+compound,-list(atom),++list(compound)) - zero_or_one

---

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

geohash, geospatial

1.78 geojson

object

1.78.1 geojson

GeoJSON parser, generator, and validator using default JSON term representations.

Availability:
logtalk_load(geojson(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-08

Compilation flags:
static, context_switching_calls

Extends:
public geojson(curly,dash,atom)

Remarks:
(none)

Inherited public predicates:
generate/2 geojson_to_json/2 json_to_geojson/2 parse/2 validate/1 validate/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.78.2 geojson(ObjectRepresentation,PairRepresentation,StringRepresentation)

- ObjectRepresentation - Object representation to be used when decoding embedded JSON objects. Possible values are curly (default) and list.
- PairRepresentation - Pair representation to be used when decoding embedded JSON objects. Possible values are dash (default), equal, and colon.
- StringRepresentation - Text representation to be used when decoding JSON strings. Possible values are atom (default), chars, and codes.

GeoJSON (RFC 7946) parser, generator, and validator built on top of the json library.

Availability:
    logtalk_load(geojson(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-08

Compilation flags:
    static, context_switching_calls

Implements:
public geojson_protocol
Uses:
json(ObjectRepresentation,PairRepresentation,StringRepresentation)
list

Remarks:
(none)

Inherited public predicates:
generate/2 geojson_to_json/2 json_to_geojson/2 parse/2 validate/1 validate/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

json, geospatial

protocol

1.78.3 geojson_protocol

GeoJSON (RFC 7946) parser, generator, and validator protocol.

Availability:
logtalk_load(geojson(loader))

Author: Paulo Moura
Version: 0:1:0
Date: 2026-05-08

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - parse/2
  - generate/2
  - validate/1
  - validate/2
  - json_to_geojson/2
  - geojson_to_json/2
- Protected predicates
- Private predicates
- Operators

Public predicates

parse/2

Parses a GeoJSON document from the given source (file(Path), stream(Stream), codes(List), chars(List), or atom(Atom)) into a native GeoJSON term.

Compilation flags:
    static

Template:
    parse(Source,Term)
Mode and number of proofs:
    parse(++compound,--term) - one_or_error

Exceptions:
    Source is a variable:
        instantiation_error
    Source is neither a variable nor a valid source:
        domain_error(geojson_source,Source)
    Parsed JSON does not represent a valid GeoJSON document:
        domain_error(geojson,Document)

---

generate/2

Generates GeoJSON content using the representation specified in the first argument (file(Path), stream(Stream), codes(List), chars(List), or atom(Atom)) from a native GeoJSON term.

Compilation flags:
    static

Template:
    generate(Sink,Term)
Mode and number of proofs:
    generate(+compound,++term) - one_or_error

Exceptions:
    Sink is a variable:
        instantiation_error
    Sink is neither a variable nor a valid sink:

domain_error(geojson_sink,Sink)
Term is not a valid GeoJSON term:
domain_error(geojson,Term)

---

validate/1

True iff the argument is a valid native GeoJSON term according to RFC 7946.

Compilation flags:
static

Template:
validate(Term)
Mode and number of proofs:
validate(@term) - zero_or_one

---

validate/2

Validates a native GeoJSON term returning a list of reason terms whose final argument is the failing path.

Compilation flags:
static

Template:
validate(Term,Errors)
Mode and number of proofs:
validate(@term,-list(compound)) - one_or_error

Remarks:

- invalid_geojson_term(Path): The term is not any supported GeoJSON geometry, feature, or feature collection representation.
- invalid_geometry(Path): A geometry member is neither a valid geometry term nor @null where @null is allowed.
- invalid_properties(Path): A feature properties member is neither @null nor a valid JSON object term in the selected representation.
- invalid_options(Path): The options argument is not a list.

- option_not_allowed(Name, Path): An option with the given name is not allowed for the enclosing GeoJSON term.
- duplicate_option(Name, Path): The same option name occurs more than once in the options list.
- unknown_option(Option, Path): An option term is not recognized as one of the supported options.
- invalid_bbox(Path): A bounding box is not a list of four or six numbers.
- bbox_longitude_out_of_range(Path): A bounding box longitude value lies outside the RFC 7946 range [-180,180].
- bbox_latitude_out_of_range(Path): A bounding box latitude value lies outside the RFC 7946 range [-90,90].
- bbox_latitude_order(Path): The south latitude value is greater than the north latitude value in a bounding box.
- bbox_altitude_order(Path): The minimum altitude value is greater than the maximum altitude value in a three-dimensional bounding box.
- bbox_dimension_mismatch(Actual, Expected, Path): A bounding box length does not match the dimensionality required by the represented geometry, feature, or collection.
- invalid_id(Path): A feature identifier is neither a number nor a valid string term in the selected representation.
- invalid_foreign_members(Path): Foreign members are not represented as a valid pair list or embedded JSON object term, or contain an invalid pair term.
- prohibited_member(Key, Path): A prohibited foreign member name such as crs was used.
- reserved_foreign_member(Key, Path): A reserved GeoJSON member name was used as a foreign member key.
- invalid_position(Path): A position is not a list of at least two numeric coordinates.
- position_longitude_out_of_range(Path): A position longitude value lies outside the RFC 7946 range [-180,180].
- position_latitude_out_of_range(Path): A position latitude value lies outside the RFC 7946 range [-90,90].
- invalid_json_object_pair(Path): A JSON object member is not represented using a supported key-value pair term.
- invalid_json_key(Path): A JSON object key is not a valid string term in the selected representation.
- duplicate_json_object_key(Key, Path): The same JSON object key occurs more than once in the same object.
- invalid_json_value(Path): A JSON value is not a number, boolean, null, valid string term, array, or object term.
- insufficient_positions(MinimumLength, Path): A coordinate array contains fewer positions than required for the enclosing geometry.
- invalid_position_array(Path): A coordinate array expected to contain positions is not a list of valid positions.
- inconsistent_position_dimension(Path): Positions in the same coordinate array do not all have the same dimensionality.
- invalid_line_string_array(Path): A MultiLineString coordinate array is not a valid list of line string coordinate arrays.

- invalid_polygon(Path): A polygon coordinate structure is invalid, for example because it is not a non-empty list of linear rings.
- ring_not_closed(Path): A linear ring does not repeat its first position as its last position.
- invalid_multi_polygon(Path): A MultiPolygon coordinate array is not a valid list of polygon coordinate arrays.
- invalid_geometry_collection(Path): A GeometryCollection geometries member is not a list.
- invalid_feature_collection(Path): A FeatureCollection features member is not a list.
- invalid_feature(Path): A value inside a FeatureCollection is not a valid feature term.

---

json_to_geojson/2

Converts a JSON term, as returned by the json library, into a native GeoJSON term.

Compilation flags:
    static

Template:
    json_to_geojson(JSON,GeoJSON)
Mode and number of proofs:
    json_to_geojson(+term,-term) - one_or_error

Exceptions:
    JSON is a variable:
        instantiation_error
    JSON is not a valid GeoJSON JSON term:
        domain_error(geojson,JSON)

---

geojson_to_json/2

Converts a native GeoJSON term into a JSON term suitable for the json library.

Compilation flags:
    static

Template:
    geojson_to_json(GeoJSON,JSON)
Mode and number of proofs:

geojson_to_json(+term,-term) - one_or_error

Exceptions:
GeoJSON is a variable:
instantiation_error
GeoJSON is not a valid GeoJSON term:
domain_error(geojson,GeoJSON)

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

geojson, json, geospatial

1.79 geospatial

object

1.79.1 geospatial

Geospatial predicates over geographic coordinates represented as geographic(Latitude,Longitude).

Availability:
logtalk_load(geospatial(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-11

Compilation flags:

static, context_switching_calls

Implements:
public geospatial_protocol
Uses:
list

Remarks:

- Distance unit: Kilometers.
- Coordinate ranges: Latitude values must be in the [-90.0,90.0] range and longitude values in the [-180.0,180.0] range.

Inherited public predicates:
along_track_distance/4 bbox_contains/2 bbox_contains_polygon/2 bbox_expand/3 bbox_from_coordinates/2 bbox_intersects/2 bbox_intersects_polygon/2 bbox_intersects_polyline/2 bbox_union/3 bounding_box/3 clockwise_polygon/2 close_polygon/2 coordinates_bounding_box/2 counterclockwise_polygon/2 cross_track_distance/4 destination_point/4 equirectangular_inverse/4 equirectangular_projection/4 final_bearing/3 haversine_distance/3 initial_bearing/3 interpolate_great_circle/4 interpolate_rhumb/4 is_clockwise_polygon/1 is_valid_polygon/1 mean_center/2 midpoint/3 minimum_enclosing_circle/3 nearest_coordinate/5 nearest_point_on_polyline/4 nearest_point_on_segment/4 normalize_coordinate/2 normalize_polygon_orientation/3 point_in_polygon/2 point_to_polyline_distance/3 polygon_area/2 polygon_bounding_box/2 polygon_centroid/2 polygon_orientation/2 polygon_perimeter/2 polygon_perimeter/3 polygons_intersect/2 polyline_length/2 polyline_length/3 polyline_resample/3 polyline_simplify/3 polyline_split_at_distance/4 rhumb_bearing/3 rhumb_destination_point/4 rhumb_distance/3 rhumb_midpoint/3 route_distance/2 route_distance/3 route_distance/4 valid_coordinate/1 vincenty_distance/3 within_distance/4

- Public predicates
  - distance/4
  - distance/5
  - bbox_overlaps/2
- Protected predicates
- Private predicates
- Operators

Public predicates

distance/4

Computes the distance in kilometers between two coordinates using a selected metric. Supported metrics are haversine, vincenty, and rhumb.

Compilation flags:
    static

Template:
    distance(Coordinate1,Coordinate2,Metric,Distance)
Mode and number of proofs:
    distance(+compound,+compound,+atom,-float) - zero_or_one

---

distance/5

Computes the distance between two coordinates using a selected metric and output unit. Supported metrics are haversine, vincenty, and rhumb. Valid Unit argument values are kilometers, meters, miles, and nautical_miles.

Compilation flags:
    static

Template:
    distance(Coordinate1,Coordinate2,Metric,Unit,Distance)
Mode and number of proofs:
    distance(+compound,+compound,+atom,+atom,-float) - zero_or_one

---

bbox_overlaps/2

True when two bounding boxes have a strict positive-area overlap. Touching only at an edge or corner fails. Antimeridian-crossing bounding boxes are supported.

Compilation flags:
    static

Template:

bbox_overlaps(BoundingBox1,BoundingBox2)
Mode and number of proofs:
bbox_overlaps(+compound,+compound) - zero_or_one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

geospatial_protocol, numberlist

protocol

1.79.2 geospatial_protocol

Geospatial predicates protocol.

Availability:
logtalk_load(geospatial(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-05-11

Compilation flags:
static

Dependencies:
(none)

Remarks:

(none)

Inherited public predicates:
(none)

- Public predicates
  - valid_coordinate/1
  - normalize_coordinate/2
  - equirectangular_projection/4
  - equirectangular_inverse/4
  - haversine_distance/3
  - vincenty_distance/3
  - rhumb_distance/3
  - rhumb_bearing/3
  - rhumb_destination_point/4
  - interpolate_rhumb/4
  - rhumb_midpoint/3
  - distance/4
  - distance/5
  - initial_bearing/3
  - final_bearing/3
  - midpoint/3
  - destination_point/4
  - interpolate_great_circle/4
  - cross_track_distance/4
  - along_track_distance/4
  - within_distance/4
  - nearest_coordinate/5
  - mean_center/2
  - minimum_enclosing_circle/3
  - coordinates_bounding_box/2
  - point_in_polygon/2
  - polygon_area/2
  - polygon_centroid/2
  - polygon_bounding_box/2

  - close_polygon/2
  - polygon_orientation/2
  - is_clockwise_polygon/1
  - normalize_polygon_orientation/3
  - clockwise_polygon/2
  - counterclockwise_polygon/2
  - is_valid_polygon/1
  - bbox_contains/2
  - bbox_intersects/2
  - bbox_overlaps/2
  - bbox_union/3
  - bbox_expand/3
  - bbox_intersects_polygon/2
  - bbox_contains_polygon/2
  - bbox_from_coordinates/2
  - bbox_intersects_polyline/2
  - point_to_polyline_distance/3
  - nearest_point_on_segment/4
  - nearest_point_on_polyline/4
  - polyline_length/2
  - polyline_length/3
  - polyline_simplify/3
  - polyline_split_at_distance/4
  - polyline_resample/3
  - polygon_perimeter/2
  - polygon_perimeter/3
  - polygons_intersect/2
  - bounding_box/3
  - route_distance/2
  - route_distance/3
  - route_distance/4
- Protected predicates
- Private predicates
- Operators

Public predicates

valid_coordinate/1

True if the argument is a valid geographic coordinate represented as geographic(Latitude,Longitude) with latitude in the [-90.0,90.0] range and longitude in the [-180.0,180.0] range.

Compilation flags:
    static

Template:
    valid_coordinate(Coordinate)
Mode and number of proofs:
    valid_coordinate(@compound) - zero_or_one

---

normalize_coordinate/2

Normalizes a coordinate by wrapping longitude to the [-180.0,180.0] range and reflecting latitude over the poles when necessary.

Compilation flags:
    static

Template:
    normalize_coordinate(Coordinate,NormalizedCoordinate)
Mode and number of proofs:
    normalize_coordinate(+compound,-compound) - one

---

equirectangular_projection/4

Projects a coordinate to local equirectangular planar coordinates in kilometers using a reference latitude in degrees.

Compilation flags:
    static

Template:

equirectangular_projection(Coordinate,ReferenceLatitude,X,Y)
Mode and number of proofs:
equirectangular_projection(+compound,+float,-float,-float) - zero_or_one

---

equirectangular_inverse/4

Converts local equirectangular planar coordinates in kilometers to a geographic coordinate using a reference latitude in degrees.

Compilation flags:
static

Template:
equirectangular_inverse(X,Y,ReferenceLatitude,Coordinate)
Mode and number of proofs:
equirectangular_inverse(+float,+float,+float,-compound) - zero_or_one

---

haversine_distance/3

Computes the great-circle distance in kilometers between two coordinates using the Haversine formula.

Compilation flags:
static

Template:
haversine_distance(Coordinate1,Coordinate2,Distance)
Mode and number of proofs:
haversine_distance(+compound,+compound,-float) - zero_or_one

---

vincenty_distance/3

Computes the geodesic distance in kilometers between two coordinates using the Vincenty inverse formula over the WGS84 ellipsoid. Fails if the iterative method does not converge.

Compilation flags:
    static

Template:
    vincenty_distance(Coordinate1,Coordinate2,Distance)
Mode and number of proofs:
    vincenty_distance(+compound,+compound,-float) - zero_or_one

---

rhumb_distance/3

Computes the rhumb-line (loxodrome) distance in kilometers between two coordinates on a spherical Earth model.

Compilation flags:
    static

Template:
    rhumb_distance(Coordinate1,Coordinate2,Distance)
Mode and number of proofs:
    rhumb_distance(+compound,+compound,-float) - zero_or_one

---

rhumb_bearing/3

Computes the rhumb-line initial bearing in degrees in the [0.0,360.0[range from the first coordinate to the second coordinate.

Compilation flags:
    static

Template:
    rhumb_bearing(Coordinate1,Coordinate2,Bearing)
Mode and number of proofs:

rhumb_bearing(+compound,+compound,-float) - zero_or_one

---

rhumb_destination_point/4

Computes the destination coordinate from a start coordinate, a rhumb-line bearing in degrees, and a distance in kilometers on a spherical Earth model.

Compilation flags:
static

Template:
rhumb_destination_point(Start,Bearing,Distance,Destination)
Mode and number of proofs:
rhumb_destination_point(+compound,+float,+float,-compound) - zero_or_one

---

interpolate_rhumb/4

Computes an intermediate coordinate along the rhumb-line path between two coordinates using a fraction in the [0.0,1.0] range.

Compilation flags:
static

Template:
interpolate_rhumb(Coordinate1,Coordinate2,Fraction,Coordinate)
Mode and number of proofs:
interpolate_rhumb(+compound,+compound,+float,-compound) - zero_or_one

---

rhumb_midpoint/3

Computes the midpoint along the rhumb-line path between two coordinates.

Compilation flags:
    static

Template:
    rhumb_midpoint(Coordinate1,Coordinate2,Midpoint)
Mode and number of proofs:
    rhumb_midpoint(+compound,+compound,-compound) - zero_or_one

---

distance/4

Computes the distance in kilometers between two coordinates using a selected metric. Supported metrics are haversine, vincenty, and rhumb.

Compilation flags:
    static

Template:
    distance(Coordinate1,Coordinate2,Metric,Distance)
Mode and number of proofs:
    distance(+compound,+compound,+atom,-float) - zero_or_one

---

distance/5

Computes the distance between two coordinates using a selected metric and output unit. Supported metrics are haversine, vincenty, and rhumb. Valid Unit argument values are kilometers, meters, miles, and nautical_miles.

Compilation flags:
    static

Template:
    distance(Coordinate1,Coordinate2,Metric,Unit,Distance)
Mode and number of proofs:

distance(+compound,+compound,+atom,+atom,-float) - zero_or_one

---

initial_bearing/3

Computes the initial bearing in degrees in the [0.0,360.0[range from the first coordinate to the second coordinate.

Compilation flags:
static

Template:
initial_bearing(Coordinate1,Coordinate2,Bearing)
Mode and number of proofs:
initial_bearing(+compound,+compound,-float) - zero_or_one

---

final_bearing/3

Computes the final bearing in degrees in the [0.0,360.0[range when arriving at the second coordinate from the first coordinate.

Compilation flags:
static

Template:
final_bearing(Coordinate1,Coordinate2,Bearing)
Mode and number of proofs:
final_bearing(+compound,+compound,-float) - zero_or_one

---

midpoint/3

Computes the geographic midpoint between two coordinates using a spherical Earth model.

Compilation flags:
    static

Template:
    midpoint(Coordinate1,Coordinate2,Midpoint)
Mode and number of proofs:
    midpoint(+compound,+compound,-compound) - zero_or_one

---

destination_point/4

Computes the destination coordinate from a start coordinate, an initial bearing in degrees, and a distance in kilometers using a spherical Earth model.

Compilation flags:
    static

Template:
    destination_point(Start,Bearing,Distance,Destination)
Mode and number of proofs:
    destination_point(+compound,+float,+float,-compound) - zero_or_one

---

interpolate_great_circle/4

Computes an intermediate coordinate along the great-circle path between two coordinates using a fraction in the [0.0,1.0] range.

Compilation flags:
    static

Template:
    interpolate_great_circle(Coordinate1,Coordinate2,Fraction,Coordinate)
Mode and number of proofs:
    interpolate_great_circle(+compound,+compound,+float,-compound) - zero_or_one

---

cross_track_distance/4

Computes the signed cross-track distance in kilometers from a coordinate to the great-circle path defined by a start and end coordinate.

Compilation flags:
    static

Template:
    cross_track_distance(Coordinate,Start,End,Distance)
Mode and number of proofs:
    cross_track_distance(+compound,+compound,+compound,-float) - zero_or_one

---

along_track_distance/4

Computes the along-track distance in kilometers from the start coordinate to the closest point to a coordinate on the great-circle path defined by a start and end coordinate.

Compilation flags:
    static

Template:
    along_track_distance(Coordinate,Start,End,Distance)
Mode and number of proofs:
    along_track_distance(+compound,+compound,+compound,-float) - zero_or_one

---

within_distance/4

True when the distance between two coordinates is less than or equal to the given radius in kilometers using the selected metric.

Compilation flags:
    static

Template:

within_distance(Coordinate1,Coordinate2,Radius,Metric)
Mode and number of proofs:
within_distance(+compound,+compound,+float,+atom) - zero_or_one

---

nearest_coordinate/5

Finds the nearest coordinate to the origin coordinate in a list using the selected metric, returning the nearest coordinate and distance in kilometers.

Compilation flags:
static

Template:
nearest_coordinate(Origin,Coordinates,Metric,Nearest,Distance)
Mode and number of proofs:
nearest_coordinate(+compound,+list(compound),+atom,-compound,-float) - zero_or_one

---

mean_center/2

Computes the arithmetic mean center of a list of one or more coordinates.

Compilation flags:
static

Template:
mean_center(Coordinates,Center)
Mode and number of proofs:
mean_center(+list(compound),-compound) - zero_or_one

---

minimum_enclosing_circle/3

Computes an approximate minimum enclosing circle for a list of one or more coordinates, returning the circle center coordinate and radius in kilometers.

Compilation flags:
    static

Template:
    minimum_enclosing_circle(Coordinates,Center,Radius)
Mode and number of proofs:
    minimum_enclosing_circle(+list(compound),-compound,-float) - zero_or_one

---

coordinates_bounding_box/2

Computes the axis-aligned latitude/longitude bounding box for a list of one or more coordinates.

Compilation flags:
    static

Template:
    coordinates_bounding_box(Coordinates,BoundingBox)
Mode and number of proofs:
    coordinates_bounding_box(+list(compound),-compound) - zero_or_one

---

point_in_polygon/2

True when a coordinate is inside (or on the boundary of) a polygon represented as a list of coordinates. Uses a planar ray-casting algorithm over latitude/longitude coordinates.

Compilation flags:
    static

Template:
    point_in_polygon(Point,Polygon)
Mode and number of proofs:
    point_in_polygon(+compound,+list(compound)) - zero_or_one

---

polygon_area/2

Computes an approximate polygon area in square kilometers by projecting coordinates to a local equirectangular plane and applying the shoelace formula.

Compilation flags:
    static

Template:
    polygon_area(Polygon,Area)
Mode and number of proofs:
    polygon_area(+list(compound),-float) - zero_or_one

---

polygon_centroid/2

Computes an approximate polygon centroid by using a local equirectangular projection and planar centroid formula.

Compilation flags:
    static

Template:
    polygon_centroid(Polygon,Centroid)
Mode and number of proofs:
    polygon_centroid(+list(compound),-compound) - zero_or_one

---

polygon_bounding_box/2

Computes the axis-aligned latitude/longitude bounding box for a polygon represented as a list of coordinates.

Compilation flags:
    static

Template:

polygon_bounding_box(Polygon,BoundingBox)
Mode and number of proofs:
polygon_bounding_box(+list(compound),-compound) - zero_or_one

---

close_polygon/2

Returns a closed polygon ring by ensuring the first coordinate is repeated at the end of the list.

Compilation flags:
static

Template:
close_polygon(Polygon,ClosedPolygon)
Mode and number of proofs:
close_polygon(+list(compound),-list(compound)) - zero_or_one

---

polygon_orientation/2

Computes polygon ring orientation as clockwise or counterclockwise using a local projected signed area approximation.

Compilation flags:
static

Template:
polygon_orientation(Polygon,Orientation)
Mode and number of proofs:
polygon_orientation(+list(compound),-atom) - zero_or_one

---

is_clockwise_polygon/1

True when a polygon ring orientation is clockwise.

Compilation flags:
    static

Template:
    is_clockwise_polygon(Polygon)
Mode and number of proofs:
    is_clockwise_polygon(+list(compound)) - zero_or_one

---

normalize_polygon_orientation/3

Normalizes a polygon ring orientation to clockwise or counterclockwise.

Compilation flags:
    static

Template:
    normalize_polygon_orientation(Polygon,Orientation,OrientedPolygon)
Mode and number of proofs:
    normalize_polygon_orientation(+list(compound),+atom,-list(compound)) - zero_or_one

---

clockwise_polygon/2

Returns a polygon ring with clockwise orientation.

Compilation flags:
    static

Template:
    clockwise_polygon(Polygon,ClockwisePolygon)
Mode and number of proofs:
    clockwise_polygon(+list(compound),-list(compound)) - zero_or_one

---

counterclockwise_polygon/2

Returns a polygon ring with counterclockwise orientation.

Compilation flags:
    static

Template:
    counterclockwise_polygon(Polygon,CounterclockwisePolygon)
Mode and number of proofs:
    counterclockwise_polygon(+list(compound),-list(compound)) - zero_or_one

---

is_valid_polygon/1

True when a polygon has at least three valid coordinates after normalizing optional closure.

Compilation flags:
    static

Template:
    is_valid_polygon(Polygon)
Mode and number of proofs:
    is_valid_polygon(+list(compound)) - zero_or_one

---

bbox_contains/2

True when a coordinate or bounding box is inside or on the boundary of a bounding box term bbox(geographic(MinLatitude,MinLongitude),geographic(MaxLatitude,MaxLongitude)). Antimeridian-crossing bounding boxes are supported.

Compilation flags:
    static

Template:
    bbox_contains(BoundingBox,Contained)
Mode and number of proofs:
    bbox_contains(+compound,+compound) - zero_or_one

---

bbox_intersects/2

True when two bounding boxes intersect or touch. Antimeridian-crossing bounding boxes are supported.

Compilation flags:
    static

Template:
    bbox_intersects(BoundingBox1,BoundingBox2)
Mode and number of proofs:
    bbox_intersects(+compound,+compound) - zero_or_one

---

bbox_overlaps/2

True when two bounding boxes have a strict positive-area overlap. Touching only at an edge or corner fails. Antimeridian-crossing bounding boxes are supported.

Compilation flags:
    static

Template:
    bbox_overlaps(BoundingBox1,BoundingBox2)
Mode and number of proofs:
    bbox_overlaps(+compound,+compound) - zero_or_one

---

bbox_union/3

Computes the minimal bounding box containing two bounding boxes.

Compilation flags:
    static

Template:
    bbox_union(BoundingBox1,BoundingBox2,BoundingBox)

Mode and number of proofs:
bbox_union(+compound,+compound,-compound) - zero_or_one

---

bbox_expand/3

Expands a bounding box by a distance in kilometers on all sides using a local spherical approximation.

Compilation flags:
static

Template:
bbox_expand(BoundingBox,Distance,ExpandedBoundingBox)
Mode and number of proofs:
bbox_expand(+compound,+number,-compound) - zero_or_one

---

bbox_intersects_polygon/2

True when a bounding box intersects a polygon or either contains the other. Antimeridian-crossing bounding boxes are supported.

Compilation flags:
static

Template:
bbox_intersects_polygon(BoundingBox,Polygon)
Mode and number of proofs:
bbox_intersects_polygon(+compound,+list(compound)) - zero_or_one

---

bbox_contains_polygon/2

True when a bounding box fully contains a polygon, including polygon vertices on the boundary. Antimeridian-crossing bounding boxes are supported.

Compilation flags:
    static

Template:
    bbox_contains_polygon(BoundingBox,Polygon)
Mode and number of proofs:
    bbox_contains_polygon(+compound,+list(compound)) - zero_or_one

---

bbox_from_coordinates/2

Computes a bounding box from a list of one or more coordinates.

Compilation flags:
    static

Template:
    bbox_from_coordinates(Coordinates,BoundingBox)
Mode and number of proofs:
    bbox_from_coordinates(+list(compound),-compound) - zero_or_one

---

bbox_intersects_polyline/2

True when a bounding box intersects or touches a polyline with two or more coordinates. Antimeridian-crossing bounding boxes are supported.

Compilation flags:
    static

Template:
    bbox_intersects_polyline(BoundingBox,Polyline)
Mode and number of proofs:
    bbox_intersects_polyline(+compound,+list(compound)) - zero_or_one

---

point_to_polyline_distance/3

Computes the minimum distance in kilometers from a coordinate to a polyline with two or more coordinates.

Compilation flags:
    static

Template:
    point_to_polyline_distance(Point,Polyline,Distance)
Mode and number of proofs:
    point_to_polyline_distance(+compound,+list(compound),-float) - zero_or_one

---

nearest_point_on_segment/4

Computes the nearest coordinate on a segment to a coordinate using a local equirectangular approximation.

Compilation flags:
    static

Template:
    nearest_point_on_segment(Point,SegmentStart,SegmentEnd,NearestPoint)
Mode and number of proofs:
    nearest_point_on_segment(+compound,+compound,+compound,-compound) - zero_or_one

---

nearest_point_on_polyline/4

Computes the nearest coordinate on a polyline to a coordinate and the corresponding distance in kilometers.

Compilation flags:
    static

Template:
    nearest_point_on_polyline(Point,Polyline,NearestPoint,Distance)
Mode and number of proofs:

nearest_point_on_polyline(+compound,+list(compound),-compound,-float) - zero_or_one

---

polyline_length/2

Computes the polyline length in kilometers for a list of two or more coordinates using the default haversine metric.

Compilation flags:
static

Template:
polyline_length(Coordinates,Length)
Mode and number of proofs:
polyline_length(+list(compound),-float) - zero_or_one

---

polyline_length/3

Computes the polyline length in kilometers for a list of two or more coordinates using the selected metric.

Compilation flags:
static

Template:
polyline_length(Coordinates,Metric,Length)
Mode and number of proofs:
polyline_length(+list(compound),+atom,-float) - zero_or_one

---

polyline_simplify/3

Simplifies a polyline using the Douglas-Peucker algorithm with a tolerance in kilometers.

Compilation flags:
static

Template:
polyline_simplify(Coordinates,Tolerance,SimplifiedCoordinates)
Mode and number of proofs:
polyline_simplify(+list(compound),+number,-list(compound)) - zero_or_one

---

polyline_split_at_distance/4

Splits a polyline at a distance in kilometers from its first coordinate, returning left and right polylines that share the split coordinate.

Compilation flags:
static

Template:
polyline_split_at_distance(Coordinates,Distance,LeftCoordinates,RightCoordinates)
Mode and number of proofs:
polyline_split_at_distance(+list(compound),+number,-list(compound),-list(compound)) - zero_or_one

---

polyline_resample/3

Resamples a polyline using a fixed step in kilometers, preserving first and last coordinates.

Compilation flags:
static

Template:
polyline_resample(Coordinates,Step,ResampledCoordinates)
Mode and number of proofs:
polyline_resample(+list(compound),+number,-list(compound)) - zero_or_one

---

polygon_perimeter/2

Computes the polygon perimeter in kilometers using the default haversine metric.

Compilation flags:
    static

Template:
    polygon_perimeter(Polygon,Perimeter)
Mode and number of proofs:
    polygon_perimeter(+list(compound),-float) - zero_or_one

---

polygon_perimeter/3

Computes the polygon perimeter in kilometers using the selected metric.

Compilation flags:
    static

Template:
    polygon_perimeter(Polygon,Metric,Perimeter)
Mode and number of proofs:
    polygon_perimeter(+list(compound),+atom,-float) - zero_or_one

---

polygons_intersect/2

True when two polygons intersect or one polygon is contained in the other.

Compilation flags:
    static

Template:
    polygons_intersect(Polygon1,Polygon2)
Mode and number of proofs:
    polygons_intersect(+list(compound),+list(compound)) - zero_or_one

---

bounding_box/3

Computes a spherical bounding box around a center coordinate for a given radius in kilometers. The returned bounding box term is bbox(geographic(MinLatitude,MinLongitude),geographic(MaxLatitude, MaxLongitude)).

Compilation flags:
    static

Template:
    bounding_box(Center,Radius,BoundingBox)
Mode and number of proofs:
    bounding_box(+compound,+positive_number,-compound) - zero_or_one

---

route_distance/2

Computes the route distance in kilometers for a list of two or more coordinates using the default haversine metric.

Compilation flags:
    static

Template:
    route_distance(Coordinates,Distance)
Mode and number of proofs:
    route_distance(+list(compound),-float) - zero_or_one

---

route_distance/3

Computes the route distance in kilometers for a list of two or more coordinates using the selected metric. Supported metrics are haversine, vincenty, and rhumb.

Compilation flags:
    static

Template:
    route_distance(Coordinates,Metric,Distance)

Mode and number of proofs:
route_distance(+list(compound),+atom,-float) - zero_or_one

---

route_distance/4

Computes the route distance for a list of two or more coordinates using the selected metric and output unit. Supported metrics are haversine, vincenty, and rhumb. Valid Unit argument values are kilometers, meters, miles, and nautical_miles.

Compilation flags:
static

Template:
route_distance(Coordinates,Metric,Unit,Distance)
Mode and number of proofs:
route_distance(+list(compound),+atom,+atom,-float) - zero_or_one

---

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

geospatial, numberlistp, listp

1.80 git

object

1.80.1 git

Predicates for accessing a git project current branch and latest commit data.

Availability:
logtalk_load(git(loader))

Author: Paulo Moura
Version: 2:1:2
Date: 2024-03-11

Compilation flags:
static, context_switching_calls

Implements:
public git_protocol
Uses:
os
user

Remarks:
(none)

Inherited public predicates:
branch/2 commit_author/2 commit_date/2 commit_hash/2 commit_hash_abbreviated/2 commit_log/3 commit_message/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.80.2 git_protocol

Predicates for accessing a git project current branch and latest commit data.

Availability:
logtalk_load(git(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2022-01-21

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - branch/2
  - commit_author/2
  - commit_date/2
  - commit_hash/2
  - commit_hash_abbreviated/2
  - commit_message/2
  - commit_log/3
- Protected predicates
- Private predicates
- Operators

Public predicates

branch/2

Returns the name of the current git branch. Fails if the directory is not a git repo or a sub-directory of a git repo directory.

Compilation flags:
    static

Template:
    branch(Directory,Branch)
Mode and number of proofs:
    branch(+atom,?atom) - zero_or_one

---

commit_author/2

Returns the latest commit author. Fails if the directory is not a git repo or a sub-directory of a git repo directory.

Compilation flags:
    static

Template:

commit_author(Directory,Author)
Mode and number of proofs:
commit_author(+atom,-atom) - zero_or_one

---

commit_date/2

Returns the latest commit date (strict ISO 8601 format). Fails if the directory is not a git repo or a sub-directory of a git repo directory.

Compilation flags:
static

Template:
commit_date(Directory,Date)
Mode and number of proofs:
commit_date(+atom,-atom) - zero_or_one

---

commit_hash/2

Returns the latest commit hash. Fails if the directory is not a git repo or a sub-directory of a git repo directory.

Compilation flags:
static

Template:
commit_hash(Directory,Hash)
Mode and number of proofs:
commit_hash(+atom,-atom) - zero_or_one

---

commit_hash_abbreviated/2

Returns the latest commit abbreviated hash. Fails if the directory is not a git repo or a sub-directory of a git repo directory.

Compilation flags:
    static

Template:
    commit_hash_abbreviated(Directory,Hash)
Mode and number of proofs:
    commit_hash_abbreviated(+atom,-atom) - zero_or_one

---

commit_message/2

Returns the latest commit message. Fails if the directory is not a git repo or a sub-directory of a git repo directory.

Compilation flags:
    static

Template:
    commit_message(Directory,Message)
Mode and number of proofs:
    commit_message(+atom,-atom) - zero_or_one

---

commit_log/3

Returns the git latest commit log output for the given format (see e.g. https://git-scm.com/docs/pretty-formats). Fails if the directory is not a git repo or a sub-directory of a git repo directory.

Compilation flags:
    static

Template:
    commit_log(Directory,Format,Output)
Mode and number of proofs:

commit_log(+atom,+atom,-atom) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.81 glicko2_periodic_ranker

object

1.81.1 glicko2_periodic_ranker

Multi-period Glicko-2 ranker over temporal pairwise game datasets with explicit rating periods and per-game results.

Availability:
logtalk_load(glicko2_periodic_ranker(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public ranking_dataset_common
public score_ranker_model_common
public glicko2_common
Uses:
avltree
list

Remarks:

(none)

Inherited public predicates:

check_option/1 check_options/1 check_ranker/1 default_option/1 default_options/1
diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 grouped_dataset_groups/2
grouped_dataset_items/2 grouped_dataset_items/3 grouped_dataset_summary/2 learn/2
learn/3 option/2 option/3 pairwise_dataset_connected_components/2
pairwise_dataset_declared_items/2 pairwise_dataset_items/2 pairwise_dataset_matchups/2
pairwise_dataset_preferences/2 pairwise_dataset_summary/2 pairwise_dataset_win_totals/2
pairwise_measurement_dataset_connected_components/2
pairwise_measurement_dataset_declared_items/2 pairwise_measurement_dataset_items/2
pairwise_measurement_dataset_measurements/2 pairwise_measurement_dataset_summary/2
print_ranker/1 rank/3 ranker_options/2 scores/2
temporal_pairwise_dataset_connected_components/2
temporal_pairwise_dataset_declared_items/2 temporal_pairwise_dataset_games/2
temporal_pairwise_dataset_games/3 temporal_pairwise_dataset_items/2
temporal_pairwise_dataset_periods/2 temporal_pairwise_dataset_summary/2 valid_option/1
valid_options/1 valid_ranker/1 validate_grouped_dataset/1 validate_grouped_dataset/2
validate_pairwise_dataset/1 validate_pairwise_dataset/2
validate_pairwise_measurement_dataset/1 validate_pairwise_measurement_dataset/2
validate_temporal_pairwise_dataset/1 validate_temporal_pairwise_dataset/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

temporal_pairwise_ranking_dataset_protocol, ranker_protocol, glicko2_ranker

1.82 glicko2_ranker

object

1.82.1 glicko2_ranker

Glicko-2 pairwise preference ranker. Learns one deterministic rating per item from a dataset object implementing the pairwise_ranking_dataset_protocol protocol by applying one Glicko-2 rating-period update over the aggregated pairwise outcomes and returns a self-describing ranker term with diagnostics that can be used for ranking and export.

Availability:
logtalk_load(glicko2_ranker(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public ranking_dataset_common
public score_ranker_model_common
Uses:
avltree
list

Remarks:

(none)

Inherited public predicates:

check_option/1 check_options/1 check_ranker/1 default_option/1 default_options/1
diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 grouped_dataset_groups/2
grouped_dataset_items/2 grouped_dataset_items/3 grouped_dataset_summary/2 learn/2
learn/3 option/2 option/3 pairwise_dataset_connected_components/2
pairwise_dataset_declared_items/2 pairwise_dataset_items/2 pairwise_dataset_matchups/2
pairwise_dataset_preferences/2 pairwise_dataset_summary/2 pairwise_dataset_win_totals/2
pairwise_measurement_dataset_connected_components/2
pairwise_measurement_dataset_declared_items/2 pairwise_measurement_dataset_items/2
pairwise_measurement_dataset_measurements/2 pairwise_measurement_dataset_summary/2
print_ranker/1 rank/3 ranker_options/2 scores/2
temporal_pairwise_dataset_connected_components/2
temporal_pairwise_dataset_declared_items/2 temporal_pairwise_dataset_games/2
temporal_pairwise_dataset_games/3 temporal_pairwise_dataset_items/2
temporal_pairwise_dataset_periods/2 temporal_pairwise_dataset_summary/2 valid_option/1
valid_options/1 valid_ranker/1 validate_grouped_dataset/1 validate_grouped_dataset/2
validate_pairwise_dataset/1 validate_pairwise_dataset/2
validate_pairwise_measurement_dataset/1 validate_pairwise_measurement_dataset/2
validate_temporal_pairwise_dataset/1 validate_temporal_pairwise_dataset/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

pairwise_ranking_dataset_protocol, ranker_protocol, elo_ranker, rank_centrality

1.83 gradient_boosting_classifier

object

1.83.1 gradient_boosting_classifier

Gradient boosting classifier using multinomial additive models fitted by regression trees to softmax residuals.

Availability:
logtalk_load(gradient_boosting_classifier(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-11

Compilation flags:
static, context_switching_calls

Imports:
public probabilistic_classifier_common
Uses:
format
list
numberlist
regression_tree
type

Remarks:
(none)

Inherited public predicates:
check_classifier/1 check_option/1 check_options/1 classifier_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 predict_probabilities/3 print_classifier/1 valid_classifier/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

dataset_protocol, regression_tree, gradient_boosting_regression, adaptive_boosting_classifier, random_forest_classifier

1.84 gradient_boosting_regression

object

1.84.1 gradient_boosting_regression

Gradient boosting regression using regression trees as additive base learners fitted to successive residuals.

Availability:
logtalk_load(gradient_boosting_regression(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:
static, context_switching_calls

Imports:
public regressor_common
Uses:
format
list
population
regression_tree
type

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_regressor/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 print_regressor/1 regressor_options/2 valid_option/1 valid_options/1 valid_regressor/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

linear_regression, knn_regression, regression_tree, random_forest_regression

1.85 grammars

object

1.85.1 blank_grammars(Format)

Blank grammars.

Availability:
logtalk_load(grammars(loader))

Author: Paulo Moura
Version: 0:4:0
Date: 2025-10-06

Compilation flags:
static, context_switching_calls

Dependencies:

(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - white_space//0
  - white_spaces//0
  - space//0
  - spaces//0
  - tab//0
  - tabs//0
  - new_line//0
  - new_lines//0
  - blank//0
  - blanks//0
  - non_blank//1
  - non_blanks//1
  - control//0
  - controls//0
- Protected predicates
- Private predicates
- Operators

Public predicates

white_space//0

Consumes a single space or tab.

Compilation flags:
static

Mode and number of proofs:
    white_space - zero_or_one

---

white_spaces//0

Consumes zero or more spaces and tabs.

Compilation flags:
    static

Mode and number of proofs:
    white_spaces - one

---

space//0

Consumes a single space.

Compilation flags:
    static

Mode and number of proofs:
    space - zero_or_one

---

spaces//0

Consumes zero or more spaces.

Compilation flags:
    static

Mode and number of proofs:
    spaces - one

---

tab//0

Consumes a single tab.

Compilation flags:
    static

Mode and number of proofs:
    tab - zero_or_one

---

tabs//0

Consumes zero or more tabs.

Compilation flags:
    static

Mode and number of proofs:
    tabs - one

---

new_line//0

Consumes a single new line.

Compilation flags:
    static

Mode and number of proofs:
    new_line - zero_or_one

---

new_lines//0

Consumes zero or more new lines.

Compilation flags:
    static

Mode and number of proofs:
    new_lines - one

---

blank//0

Consumes a single space, tab, vertical tab, line feed, or new line.

Compilation flags:
    static

Mode and number of proofs:
    blank - zero_or_one

---

blanks//0

Consumes zero or more spaces, tabs, vertical tabs, line feeds, or new lines.

Compilation flags:
    static

Mode and number of proofs:
    blanks - one

---

non_blank//1

Returns a single non-blank character or character code.

Compilation flags:
    static

Template:
    non_blank(NonBlank)
Mode and number of proofs:
    non_blank(-atomic) - zero_or_one

---

non_blanks//1

Returns a (possibly empty) list of non-blank characters or character codes.

Compilation flags:
    static

Template:
    non_blanks(NonBlanks)
Mode and number of proofs:
    non_blanks(-list(atomic)) - one

---

control//0

Consumes a single control character or character code. Support for the null control character depends on the Prolog backend.

Compilation flags:
    static

Mode and number of proofs:
    control - zero_or_one

---

controls//0

Consumes zero or more control characters or character codes. Support for the null control character depends on the Prolog backend.

Compilation flags:
static

Mode and number of proofs:
controls - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

object

1.85.2 ip_grammars(Format)

IP address grammars.

Availability:
logtalk_load(grammars(loader))

Author: Paulo Moura
Version: 0:2:0
Date: 2025-10-06

Compilation flags:
static, context_switching_calls

Uses:
number_grammars(Format)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - ipv4//1
  - ipv6//1
- Protected predicates
- Private predicates
- Operators

Public predicates

ipv4//1

Parses an IPv4 network address in the format XXX.XXX.XXX.XXX where each XXX is an octet (i.e., an integer between 0 and 255).

Compilation flags:
static

Template:
ipv4(Octets)
Mode and number of proofs:
ipv4(?list(integer)) - zero_or_one

---

ipv6//1

Parses an IPv6 network address in the format XXXX.XXXX.XXXX.XXXX.XXXX.XXXX.XXXX.XXXX where each X is a hexadecimal digit.

Compilation flags:
static

Template:

ipv6(HexDigits)
Mode and number of proofs:
ipv6(?list(integer)) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.85.3 number_grammars(Format)

Number grammars.

Availability:
logtalk_load(grammars(loader))

Author: Paulo Moura
Version: 0:3:0
Date: 2025-10-06

Compilation flags:
static, context_switching_calls

Uses:
list

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - bit//1
  - bits//1
  - digit//1
  - digits//1
  - hex_digit//1
  - hex_digits//1
  - natural//1
  - integer//1
  - float//1
  - number//1
  - sign//1
  - dot//1
- Protected predicates
- Private predicates
- Operators

Public predicates

bit//1

Parses a single bit.

Compilation flags:
    static

Template:
    bit(Bit)
Mode and number of proofs:
    bit(?integer) - zero_or_one

bits//1

Parses a sequence of one or more bits.

Compilation flags:
    static

Template:
    bits(Bits)
Mode and number of proofs:
    bits(?list(integer)) - zero_or_one

---

digit//1

Parses a single decimal digit.

Compilation flags:
    static

Template:
    digit(Digit)
Mode and number of proofs:
    digit(?atomic) - zero_or_one

---

digits//1

Parses a sequence of zero of more digits.

Compilation flags:
    static

Template:
    digits(Digits)
Mode and number of proofs:
    digits(?list(atomic)) - one

---

hex_digit//1

Parses a single hexa-decimal digit.

Compilation flags:
    static

Template:
    hex_digit(HexDigit)
Mode and number of proofs:
    hex_digit(?atomic) - zero_or_one

---

hex_digits//1

Parses a sequence of zero or more hexa-decimal digits.

Compilation flags:
    static

Template:
    hex_digits(HexDigits)
Mode and number of proofs:
    hex_digits(?list(atomic)) - one

---

natural//1

Parses a natural number (a non signed integer).

Compilation flags:
    static

Template:
    natural(Natural)
Mode and number of proofs:
    natural(?non_negative_integer) - zero_or_one

---

integer//1

Parses an integer.

Compilation flags:
    static

Template:
    integer(Integer)
Mode and number of proofs:
    integer(?integer) - zero_or_one

---

float//1

Parses a float.

Compilation flags:
    static

Template:
    float(Float)
Mode and number of proofs:
    float(?float) - zero_or_one

---

number//1

Parses a number (an integer or a float).

Compilation flags:
    static

Template:
    number(Number)
Mode and number of proofs:
    number(?number) - zero_or_one

---

sign//1

Parses a number sign (plus or minus).

Compilation flags:
    static

Template:
    sign(Sign)
Mode and number of proofs:
    sign(?atomic) - zero_or_one

---

dot//1

Parses a decimal dot.

Compilation flags:
    static

Template:
    dot(Dot)
Mode and number of proofs:
    dot(?atomic) - zero_or_one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.85.4 sequence_grammars

Sequence grammars.

Availability:
logtalk_load(grammars(loader))

Author: Paulo Moura
Version: 0:4:0
Date: 2025-10-06

Compilation flags:
static, context_switching_calls

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - zero_or_more//2
  - one_or_more//2
  - zero_or_more//1
  - one_or_more//1
  - zero_or_more//0
  - one_or_more//0
  - without//2
- Protected predicates
- Private predicates

- Operators

Public predicates

zero_or_more//2

Eagerly collect zero or more terminals that satisfy the given closure.

Compilation flags:
    static

Template:
    zero_or_more(Closure,Terminals)
Meta-predicate template:
    zero_or_more(1,*)
Mode and number of proofs:
    zero_or_more(+callable,-list(atomic)) - one

---

one_or_more//2

Eagerly collect one or more terminals that satisfy the given closure.

Compilation flags:
    static

Template:
    one_or_more(Closure,Terminals)
Meta-predicate template:
    one_or_more(1,*)
Mode and number of proofs:
    one_or_more(+callable,-list(atomic)) - zero_or_one

---

zero_or_more//1

Eagerly collect zero or more terminals.

Compilation flags:
    static

Template:
    zero_or_more(Terminals)
Mode and number of proofs:
    zero_or_more(-list(atomic)) - one

---

one_or_more//1

Eagerly collect one or more terminals.

Compilation flags:
    static

Template:
    one_or_more(Terminals)
Mode and number of proofs:
    one_or_more(-list(atomic)) - zero_or_one

---

zero_or_more//0

Eagerly parse zero or more terminals.

Compilation flags:
    static

Mode and number of proofs:
    zero_or_more - one

---

one_or_more//0

Eagerly parse one or more terminals.

Compilation flags:
    static

Mode and number of proofs:
    one_or_more - zero_or_one

---

without//2

Collects input terminals until one of the stop terminals is found. The stop terminals are excluded from the collected terminals.

Compilation flags:
    static

Template:
    without(StopTerminals,Terminals)
Mode and number of proofs:
    without(+list(atomic),-list(atomic)) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.86 graphs

category

1.86.1 directed_graph_common

Common directed graph predicates shared by directed graph objects. Uses self-dispatch to call object-specific predicates and avltree for internal bookkeeping.

Availability:
logtalk_load(graphs(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-25

Compilation flags:
static

Implements:
public directed_graph_protocol
Extends:
public graph_common
Uses:
avltree
list

Remarks:
(none)

Inherited public predicates:
add_edges/3 add_vertex/3 add_vertices/3 all_pairs_min_paths/2 all_pairs_min_predecessors/2 breadth_first_order/3 cycle/2 delete_edges/3 delete_vertex/3 delete_vertices/3 depth_first_order/3 edges/2 empty/1 has_cycle/1 has_path/3 in_degree/3 is_acyclic/1 is_bipartite/1 is_complete/1 is_sparse/1 max_path/5 min_distances/3 min_path/5 min_predecessors/3 neighbors/3 new/1 new/2 new/3 number_of_edges/2 number_of_vertices/2 out_degree/3 path/3 reachable/3 strongly_connected_components/2 symmetric_closure/2 topological_sort/2 transitive_closure/2 transpose/2 vertices/2 weakly_connected_components/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.86.2 directed_graph_protocol

Protocol for directed graph predicates such as transpose, closures, topological sorting, degree queries, and strongly connected components.

Availability:
logtalk_load(graphs(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-25

Compilation flags:
static

Extends:
public graph_protocol

Remarks:
(none)

Inherited public predicates:
add_edges/3 add_vertex/3 add_vertices/3 all_pairs_min_paths/2 all_pairs_min_predecessors/2 breadth_first_order/3 delete_edges/3 delete_vertex/3 delete_vertices/3 depth_first_order/3 edges/2 empty/1 has_path/3 is_bipartite/1 is_complete/1 is_sparse/1 max_path/5 min_distances/3 min_path/5 min_predecessors/3 neighbors/3 new/1 new/2 new/3 number_of_edges/2 number_of_vertices/2 path/3 reachable/3 vertices/2

- Public predicates
  - transpose/2
  - transitive_closure/2
  - symmetric_closure/2
  - topological_sort/2
  - in_degree/3
  - out_degree/3
  - is_acyclic/1
  - has_cycle/1
  - cycle/2
  - strongly_connected_components/2
  - weakly_connected_components/2
- Protected predicates
- Private predicates
- Operators

Public predicates

transpose/2

Unifies NewGraph with a graph obtained by reversing all edges.

Compilation flags:
static

Template:
transpose(Graph,NewGraph)
Mode and number of proofs:

transpose(+graph,-graph) - one

---

transitive_closure/2

Generates the transitive closure of the graph.

Compilation flags:
static

Template:
transitive_closure(Graph,Closure)
Mode and number of proofs:
transitive_closure(+graph,-graph) - one

---

symmetric_closure/2

Generates the symmetric closure of the graph.

Compilation flags:
static

Template:
symmetric_closure(Graph,Closure)
Mode and number of proofs:
symmetric_closure(+graph,-graph) - one

---

topological_sort/2

Unifies Sorted with a topological sort of the vertices in the graph. Assumes the graph is a DAG.

Compilation flags:
static

Template:

topological_sort(Graph,Sorted)
Mode and number of proofs:
topological_sort(+graph,-list(vertex)) - one

---

in_degree/3

Unifies Degree with the number of incoming edges to Vertex. Fails if Vertex is not in the graph.

Compilation flags:
static

Template:
in_degree(Vertex,Graph,Degree)
Mode and number of proofs:
in_degree(+vertex,+graph,-integer) - zero_or_one

---

out_degree/3

Unifies Degree with the number of outgoing edges from Vertex. Fails if Vertex is not in the graph.

Compilation flags:
static

Template:
out_degree(Vertex,Graph,Degree)
Mode and number of proofs:
out_degree(+vertex,+graph,-integer) - zero_or_one

---

is_acyclic/1

True iff the graph is a directed acyclic graph (DAG).

Compilation flags:
    static

Template:
    is_acyclic(Graph)
Mode and number of proofs:
    is_acyclic(+graph) - zero_or_one

---

has_cycle/1

True iff the graph contains at least one directed cycle.

Compilation flags:
    static

Template:
    has_cycle(Graph)
Mode and number of proofs:
    has_cycle(+graph) - zero_or_one

---

cycle/2

Enumerates directed cycles as lists of vertices where the first and last vertices are the same.

Compilation flags:
    static

Template:
    cycle(Graph,Cycle)
Mode and number of proofs:
    cycle(+graph,-list(vertex)) - zero_or_more

---

strongly_connected_components/2

Computes the strongly connected components of the graph using Tarjan's algorithm. Each component is a sorted list of vertices. Components are returned in topological order.

Compilation flags:
    static

Template:
    strongly_connected_components(Graph,SCCs)
Mode and number of proofs:
    strongly_connected_components(+graph,-list(list(vertex))) - one

---

weakly_connected_components/2

Computes the weakly connected components of the graph by considering edge directions as undirected. Each component is returned as a sorted list of vertices.

Compilation flags:
    static

Template:
    weakly_connected_components(Graph,Components)
Mode and number of proofs:
    weakly_connected_components(+graph,-list(list(vertex))) - one

---

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.86.3 graph_common

Common graph predicates shared by all graph objects. Uses self-dispatch to call object-specific predicates such as neighbors/3, vertices/2, and edges/2.

Availability:
logtalk_load(graphs(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-04-17

Compilation flags:
static

Implements:
public graph_protocol
Uses:
list
set

Remarks:
(none)

Inherited public predicates:
add_edges/3 add_vertex/3 add_vertices/3 all_pairs_min_paths/2 all_pairs_min_predecessors/2 breadth_first_order/3 delete_edges/3 delete_vertex/3 delete_vertices/3 depth_first_order/3 edges/2 empty/1 has_path/3 is_bipartite/1 is_complete/1 is_sparse/1 max_path/5 min_distances/3 min_path/5 min_predecessors/3 neighbors/3 new/1 new/2 new/3 number_of_edges/2 number_of_vertices/2 path/3 reachable/3 vertices/2

- Public predicates
- Protected predicates
  - pairs_to_edges/2

  - vertex_neighbors_to_edges/4
  - wpairs_to_edges/2
  - wvertex_neighbors_to_edges/4
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

pairs_to_edges/2

Converts a list of Vertex-Neighbors pairs from a dictionary into a flat list of Vertex1-Vertex2 edges.

Compilation flags:
    static

Template:
    pairs_to_edges(Pairs,Edges)
Mode and number of proofs:
    pairs_to_edges(+list(pair),-list) - one

---

vertex_neighbors_to_edges/4

Converts a neighbor list for a vertex into edges using a difference list.

Compilation flags:
    static

Template:
    vertex_neighbors_to_edges(Neighbors,Vertex,Edges,RestEdges)
Mode and number of proofs:
    vertex_neighbors_to_edges(+list,+vertex,-list,-list) - one

---

wpairs_to_edges/2

Converts a list of Vertex-WNeighbors pairs from a dictionary into a flat list of (Vertex1-Vertex2)-Weight weighted edges.

Compilation flags:
    static

Template:
    wpairs_to_edges(Pairs,Edges)
Mode and number of proofs:
    wpairs_to_edges(+list(pair),-list) - one

---

wvertex_neighbors_to_edges/4

Converts a weighted neighbor list for a vertex into weighted edges using a difference list.

Compilation flags:
    static

Template:
    wvertex_neighbors_to_edges(WNeighbors,Vertex,Edges,RestEdges)
Mode and number of proofs:
    wvertex_neighbors_to_edges(+list,+vertex,-list,-list) - one

---

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.86.4 graph_protocol

Common protocol for all types of graphs. Graphs are represented using a dictionary where keys are vertices and values are sorted lists of neighbors (implicitly defining edges).

Availability:
logtalk_load(graphs(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-25

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - new/1
  - new/2
  - new/3
  - empty/1
  - vertices/2
  - edges/2
  - add_vertex/3
  - add_vertices/3
  - delete_vertex/3
  - delete_vertices/3
  - add_edges/3
  - delete_edges/3

    - neighbors/3
    - reachable/3
    - breadth_first_order/3
    - depth_first_order/3
    - number_of_vertices/2
    - number_of_edges/2
    - path/3
    - has_path/3
    - min_path/5
    - max_path/5
    - min_distances/3
    - min_predecessors/3
    - all_pairs_min_paths/2
    - all_pairs_min_predecessors/2
    - is_complete/1
    - is_bipartite/1
    - is_sparse/1
- Protected predicates
- Private predicates
- Operators

Public predicates

new/1

Creates a new empty graph.

Compilation flags:
    static

Template:
    new(Graph)
Mode and number of proofs:
    new(-graph) - one

new/2

Creates a new graph from a list of edges. Vertices are defined implicitly by edges.

Compilation flags:
    static

Template:
    new(Edges,Graph)
Mode and number of proofs:
    new(+list(edge),-graph) - one

---

new/3

Creates a new graph from a list of vertices and a list of edges. Vertices may also be defined implicitly by edges.

Compilation flags:
    static

Template:
    new(Vertices,Edges,Graph)
Mode and number of proofs:
    new(+list(vertex),+list(edge),-graph) - one

---

empty/1

True iff the given graph is empty.

Compilation flags:
    static

Template:
    empty(Graph)
Mode and number of proofs:
    empty(@graph) - zero_or_one

---

vertices/2

Unifies Vertices with a sorted list of all vertices in the graph.

Compilation flags:
    static

Template:
    vertices(Graph,Vertices)
Mode and number of proofs:
    vertices(+graph,-list(vertex)) - one

---

edges/2

Unifies Edges with a list of all edges in the graph.

Compilation flags:
    static

Template:
    edges(Graph,Edges)
Mode and number of proofs:
    edges(+graph,-list(edge)) - one

---

add_vertex/3

Adds a vertex to the graph. If the vertex already exists, the graph is unchanged.

Compilation flags:
    static

Template:
    add_vertex(Graph,Vertex,NewGraph)
Mode and number of proofs:

add_vertex(+graph,+vertex,-graph) - one

---

add_vertices/3

Adds a list of vertices to the graph.

Compilation flags:
static

Template:
add_vertices(Graph,Vertices,NewGraph)
Mode and number of proofs:
add_vertices(+graph,+list(vertex),-graph) - one

---

delete_vertex/3

Deletes a vertex and all edges incident to it from the graph. If the vertex does not exist, the graph is unchanged.

Compilation flags:
static

Template:
delete_vertex(Graph,Vertex,NewGraph)
Mode and number of proofs:
delete_vertex(+graph,+vertex,-graph) - one

---

delete_vertices/3

Deletes a list of vertices and all edges incident to them from the graph.

Compilation flags:
static

Template:
delete_vertices(Graph,Vertices,NewGraph)
Mode and number of proofs:
delete_vertices(+graph,+list(vertex),-graph) - one

---

add_edges/3

Adds a list of edges to the graph. Vertices referenced by edges are added implicitly.

Compilation flags:
static

Template:
add_edges(Graph,Edges,NewGraph)
Mode and number of proofs:
add_edges(+graph,+list(edge),-graph) - one

---

delete_edges/3

Deletes a list of edges from the graph. Vertices are not deleted. Non-existing edges are silently ignored.

Compilation flags:
static

Template:
delete_edges(Graph,Edges,NewGraph)
Mode and number of proofs:
delete_edges(+graph,+list(edge),-graph) - one

---

neighbors/3

Unifies Neighbors with a sorted list of the neighbors of Vertex in the graph. Fails if Vertex is not in the graph.

Compilation flags:
    static

Template:
    neighbors(Vertex,Graph,Neighbors)
Mode and number of proofs:
    neighbors(+vertex,+graph,-list(vertex)) - zero_or_one

---

reachable/3

Unifies Vertices with a sorted list of vertices reachable from Vertex (including Vertex itself). Fails if Vertex is not in the graph.

Compilation flags:
    static

Template:
    reachable(Vertex,Graph,Vertices)
Mode and number of proofs:
    reachable(+vertex,+graph,-list(vertex)) - zero_or_one

---

breadth_first_order/3

Unifies Vertices with the breadth-first traversal order rooted at Vertex. Fails if Vertex is not in the graph.

Compilation flags:
    static

Template:
    breadth_first_order(Vertex,Graph,Vertices)
Mode and number of proofs:
    breadth_first_order(+vertex,+graph,-list(vertex)) - zero_or_one

depth_first_order/3

Unifies Vertices with the depth-first traversal order rooted at Vertex. Fails if Vertex is not in the graph.

Compilation flags:
    static

Template:
    depth_first_order(Vertex,Graph,Vertices)
Mode and number of proofs:
    depth_first_order(+vertex,+graph,-list(vertex)) - zero_or_one

---

number_of_vertices/2

Unifies N with the number of vertices in the graph.

Compilation flags:
    static

Template:
    number_of_vertices(Graph,N)
Mode and number of proofs:
    number_of_vertices(+graph,-integer) - one

---

number_of_edges/2

Unifies N with the number of edges in the graph.

Compilation flags:
    static

Template:
    number_of_edges(Graph,N)
Mode and number of proofs:

number_of_edges(+graph,-integer) - one

---

path/3

Returns a maximal path (list of vertices) rooted at Vertex, enumerating different paths on backtracking. Fails if Vertex is not in the graph.

Compilation flags:
static

Template:
path(Vertex,Graph,Path)
Mode and number of proofs:
path(+vertex,+graph,-list(vertex)) - zero_or_more

---

has_path/3

True iff there is a path from Vertex1 to Vertex2 in the graph.

Compilation flags:
static

Template:
has_path(Vertex1,Vertex2,Graph)
Mode and number of proofs:
has_path(+vertex,+vertex,+graph) - zero_or_one

---

min_path/5

Finds the minimum cost path from Vertex1 to Vertex2. For unweighted graphs, cost is the number of edges. Fails if no path exists.

Compilation flags:
static

Template:
    min_path(Vertex1,Vertex2,Graph,Path,Cost)
Mode and number of proofs:
    min_path(+vertex,+vertex,+graph,-list(vertex),-number) - zero_or_one

---

max_path/5

Finds the maximum cost acyclic path from Vertex1 to Vertex2. For unweighted graphs, cost is the number of edges. Fails if no path exists.

Compilation flags:
    static

Template:
    max_path(Vertex1,Vertex2,Graph,Path,Cost)
Mode and number of proofs:
    max_path(+vertex,+vertex,+graph,-list(vertex),-number) - zero_or_one

---

min_distances/3

Computes minimum path costs from Vertex to all reachable vertices. Returns a list of Target-Cost pairs including Vertex-0.

Compilation flags:
    static

Template:
    min_distances(Vertex,Graph,Distances)
Mode and number of proofs:
    min_distances(+vertex,+graph,-list(pair)) - zero_or_one

---

min_predecessors/3

Computes predecessor links for minimum paths rooted at Vertex. Returns a list of Target-Predecessor pairs; the source predecessor is none.

Compilation flags:
    static

Template:
    min_predecessors(Vertex,Graph,Predecessors)
Mode and number of proofs:
    min_predecessors(+vertex,+graph,-list(pair)) - zero_or_one

---

all_pairs_min_paths/2

Computes minimum path costs for all ordered pairs of vertices with a path between them. Returns a list of (Vertex1-Vertex2)-Cost terms.

Compilation flags:
    static

Template:
    all_pairs_min_paths(Graph,Pairs)
Mode and number of proofs:
    all_pairs_min_paths(+graph,-list(pair)) - one

---

all_pairs_min_predecessors/2

Computes predecessor links for minimum paths for all ordered source-target pairs with a path. Returns a list of (Vertex1-Vertex2)-Predecessor terms.

Compilation flags:
    static

Template:
    all_pairs_min_predecessors(Graph,Pairs)
Mode and number of proofs:

all_pairs_min_predecessors(+graph,-list(pair)) - one

---

is_complete/1

True iff every pair of distinct vertices in the graph is connected by an edge.

Compilation flags:
static

Template:
is_complete(Graph)
Mode and number of proofs:
is_complete(+graph) - zero_or_one

---

is_bipartite/1

True iff the graph is bipartite, i.e. its vertices can be partitioned into two sets such that every edge connects a vertex in one set to a vertex in the other.

Compilation flags:
static

Template:
is_bipartite(Graph)
Mode and number of proofs:
is_bipartite(+graph) - zero_or_one

---

is_sparse/1

True iff the graph is sparse, i.e. the number of edges is at most |V| * log2(|V|). The cutoff |E| = |V| * log2(|V|) separates sparse graphs (where adjacency lists are efficient) from dense graphs (where adjacency matrix representations may be preferable).

Compilation flags:

static

Template:
is_sparse(Graph)
Mode and number of proofs:
is_sparse(+graph) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

category

1.86.5 graph_types

A set of graph related types and generators.

Availability:
logtalk_load(graphs(loader))

Author: Paulo Moura
Version: 2:0:0
Date: 2026-02-19

Compilation flags:
static

Provides:
type::type/1
type::check/2
arbitrary::arbitrary/1
arbitrary::arbitrary/2
Uses:

integer
type

Remarks:

- Provided types: This category adds vertex, edge, and weighted_edge types for type-checking when using the type library object.
- Type vertex: Any non-variable term.
- Type edge: An unweighted edge represented as a V1-V2 pair of vertices.
- Type weighted_edge: A weighted edge represented as (V1-V2)-Weight where V1 and V2 are vertices and Weight is a number.
- Generating edges: Use the edges(N,V0,V) generator for unweighted edges or weighted_edges(N,V0,V,W) for weighted edges. N is the upper limit to the number of edges. V0 and V must be positive integers and will be used for the range of vertices. W is the upper limit for edge weights (positive integers).

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.86.6 undirected_graph_common

Common predicates shared by undirected graph objects. Uses self-dispatch to call object-specific predicates such as is_connected/1, vertices/2, edges/2, and neighbors/3.

Availability:
logtalk_load(graphs(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-25

Compilation flags:
static

Extends:
public graph_common
Uses:
avltree
list
set

Remarks:
(none)

Inherited public predicates:
add_edges/3 add_vertex/3 add_vertices/3 all_pairs_min_paths/2 all_pairs_min_predecessors/2 breadth_first_order/3 delete_edges/3 delete_vertex/3 delete_vertices/3 depth_first_order/3 edges/2 empty/1 has_path/3 is_bipartite/1 is_complete/1 is_sparse/1 max_path/5 min_distances/3 min_path/5 min_predecessors/3 neighbors/3 new/1 new/2 new/3 number_of_edges/2 number_of_vertices/2 path/3 reachable/3 vertices/2

- Public predicates
  - is_tree/1
  - has_cycle/1
  - cycle/2
  - graph_coloring/3
  - articulation_points/2
  - bridges/2
  - maximal_cliques/2
  - maximum_cliques/2
- Protected predicates
- Private predicates
- Operators

Public predicates

is_tree/1

True iff the graph is a tree, i.e. it is connected and has exactly |V| - 1 edges.

Compilation flags:
    static

Template:
    is_tree(Graph)
Mode and number of proofs:
    is_tree(+graph) - zero_or_one

---

has_cycle/1

True iff the graph contains at least one cycle.

Compilation flags:
    static

Template:
    has_cycle(Graph)

Mode and number of proofs:
    has_cycle(+graph) - zero_or_one

---

cycle/2

Enumerates cycles as lists of vertices where the first and last vertices are the same.

Compilation flags:
    static

Template:
    cycle(Graph,Cycle)
Mode and number of proofs:
    cycle(+graph,-list(vertex)) - zero_or_more

---

graph_coloring/3

Computes a greedy vertex coloring of the graph. Coloring is a list of Vertex-Color pairs where colors are integers starting from 1. NumberOfColors is the total number of colors used.

Compilation flags:
    static

Template:
    graph_coloring(Graph,Coloring,NumberOfColors)
Mode and number of proofs:
    graph_coloring(+graph,-list(pair),-integer) - one

---

articulation_points/2

Computes all articulation points (cut vertices) of the graph. The result is a sorted list of vertices.

Compilation flags:
    static

Template:
    articulation_points(Graph,Points)
Mode and number of proofs:
    articulation_points(+graph,-list(vertex)) - one

---

bridges/2

Computes all bridges (cut edges) of the graph. Each bridge is returned once as Vertex1-Vertex2 with Vertex1 @< Vertex2.

Compilation flags:
    static

Template:
    bridges(Graph,Bridges)
Mode and number of proofs:
    bridges(+graph,-list(edge)) - one

---

maximal_cliques/2

Computes all maximal cliques of the graph using the Bron-Kerbosch algorithm with pivoting. A maximal clique is a clique that cannot be extended by adding another adjacent vertex. Each clique is a sorted list of vertices. The list of cliques is sorted in standard order.

Compilation flags:
    static

Template:
    maximal_cliques(Graph,Cliques)
Mode and number of proofs:

maximal_cliques(+graph,-list(list(vertex))) - one

---

maximum_cliques/2

Computes all maximum cliques of the graph, i.e. the largest maximal cliques. Each clique is a sorted list of vertices. The list of cliques is sorted in standard order. For an empty graph, returns an empty list.

Compilation flags:
static

Template:
maximum_cliques(Graph,Cliques)
Mode and number of proofs:
maximum_cliques(+graph,-list(list(vertex))) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.86.7 unweighted_directed_graph

Unweighted directed graph predicates using the AVL tree dictionary representation.

Availability:
logtalk_load(graphs(loader))

Author: Paulo Moura
Version: 1:0:0

Date: 2026-02-19

Compilation flags:
static, context_switching_calls

Extends:
public unweighted_directed_graph(avltree)

Remarks:
(none)

Inherited public predicates:
add_edge/4 add_edges/3 add_vertex/3 add_vertices/3 all_pairs_min_paths/2 all_pairs_min_predecessors/2 breadth_first_order/3 complement/2 compose/3 cycle/2 delete_edge/4 delete_edges/3 delete_vertex/3 delete_vertices/3 depth_first_order/3 edge/3 edges/2 empty/1 has_cycle/1 has_path/3 in_degree/3 is_acyclic/1 is_bipartite/1 is_complete/1 is_sparse/1 leaves/2 max_path/5 min_distances/3 min_path/5 min_predecessors/3 neighbors/3 new/1 new/2 new/3 number_of_edges/2 number_of_vertices/2 out_degree/3 path/3 reachable/3 strongly_connected_components/2 symmetric_closure/2 topological_sort/2 topological_sort/3 transitive_closure/2 transitive_reduction/2 transpose/2 union/3 vertices/2 weakly_connected_components/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.86.8 unweighted_directed_graph(Dictionary)

Unweighted directed graph predicates using a dictionary representation. The parametric object parameter is the dictionary to use for the graph representation.

Availability:
logtalk_load(graphs(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-20

Compilation flags:
static, context_switching_calls

Imports:
public unweighted_graph_common(Dictionary)
public directed_graph_common
Uses:
pairs
set

Remarks:
(none)

Inherited public predicates:
add_edge/4 add_edges/3 add_vertex/3 add_vertices/3 all_pairs_min_paths/2 all_pairs_min_predecessors/2 breadth_first_order/3 complement/2 cycle/2 delete_edge/4 delete_edges/3 delete_vertex/3 delete_vertices/3 depth_first_order/3 edge/3 edges/2 empty/1 has_cycle/1 has_path/3 in_degree/3 is_acyclic/1 is_bipartite/1 is_complete/1 is_sparse/1 max_path/5 min_distances/3 min_path/5 min_predecessors/3 neighbors/3 new/1 new/2 new/3 number_of_edges/2 number_of_vertices/2 out_degree/3 path/3 reachable/3 strongly_connected_components/2 symmetric_closure/2 topological_sort/2 transitive_closure/2 transpose/2 vertices/2 weakly_connected_components/2

- Public predicates
  - compose/3
  - union/3
  - topological_sort/3
  - leaves/2
  - transitive_reduction/2
- Protected predicates
- Private predicates
- Operators

Public predicates

compose/3

Composes NewGraph by connecting the drains of LeftGraph to the sources of RightGraph.

Compilation flags:
    static

Template:
    compose(LeftGraph,RightGraph,NewGraph)
Mode and number of proofs:
    compose(+graph,+graph,-graph) - one

---

union/3

Unifies UnionGraph with the union of Graph1 and Graph2.

Compilation flags:
    static

Template:
    union(Graph1,Graph2,UnionGraph)
Mode and number of proofs:
    union(+graph,+graph,-graph) - one

---

topological_sort/3

Difference list version of topological_sort/2 where Sorted0 is the tail of Sorted.

Compilation flags:
    static

Template:
    topological_sort(Graph,Sorted0,Sorted)
Mode and number of proofs:
    topological_sort(+graph,+list(vertex),-list(vertex)) - one

---

leaves/2

Unifies Leaves with a sorted list of vertices with no outgoing edges.

Compilation flags:
    static

Template:
    leaves(Graph,Leaves)
Mode and number of proofs:
    leaves(+graph,-list(vertex)) - one

---

transitive_reduction/2

Computes the transitive reduction of the graph. An edge Vertex1-Vertex2 is in the reduction iff it is in the graph and there is no other path of length >= 2 from Vertex1 to Vertex2.

Compilation flags:
    static

Template:
    transitive_reduction(Graph,Reduction)

Mode and number of proofs:
transitive_reduction(+graph,-graph) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.86.9 unweighted_graph_common(Dictionary)

Common unweighted graph predicates shared by both unweighted directed and unweighted undirected graph objects. Uses self-dispatch to call object-specific predicates such as add_edge/4, delete_edge/4, and neighbors/3.

Availability:
logtalk_load(graphs(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-20

Compilation flags:
static

Implements:
public unweighted_graph_protocol
Extends:
public graph_common
Uses:
avltree
list
set

Remarks:
(none)

Inherited public predicates:
add_edge/4 add_edges/3 add_vertex/3 add_vertices/3 all_pairs_min_paths/2 all_pairs_min_predecessors/2 breadth_first_order/3 complement/2 delete_edge/4 delete_edges/3 delete_vertex/3 delete_vertices/3 depth_first_order/3 edge/3 edges/2 empty/1 has_path/3 is_bipartite/1 is_complete/1 is_sparse/1 max_path/5 min_distances/3 min_path/5 min_predecessors/3 neighbors/3 new/1 new/2 new/3 number_of_edges/2 number_of_vertices/2 path/3 reachable/3 vertices/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.86.10 unweighted_graph_protocol

Protocol for unweighted graph predicates, extending the common graph protocol with unweighted edge operations.

Availability:
logtalk_load(graphs(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-19

Compilation flags:
static

Extends:
public graph_protocol

Remarks:
(none)

Inherited public predicates:
add_edges/3 add_vertex/3 add_vertices/3 all_pairs_min_paths/2 all_pairs_min_predecessors/2 breadth_first_order/3 delete_edges/3 delete_vertex/3 delete_vertices/3 depth_first_order/3 edges/2 empty/1 has_path/3 is_bipartite/1 is_complete/1 is_sparse/1 max_path/5 min_distances/3 min_path/5 min_predecessors/3 neighbors/3 new/1 new/2 new/3 number_of_edges/2 number_of_vertices/2 path/3 reachable/3 vertices/2

- Public predicates
  - edge/3
  - add_edge/4
  - delete_edge/4
  - complement/2
- Protected predicates
- Private predicates
- Operators

Public predicates

edge/3

True iff there is an edge between Vertex1 and Vertex2 in Graph.

Compilation flags:
static

Template:
edge(Vertex1,Vertex2,Graph)
Mode and number of proofs:
edge(+vertex,+vertex,+graph) - zero_or_one

---

add_edge/4

Adds an edge between Vertex1 and Vertex2 to the graph.

Compilation flags:
static

Template:
add_edge(Graph,Vertex1,Vertex2,NewGraph)
Mode and number of proofs:
add_edge(+graph,+vertex,+vertex,-graph) - one

---

delete_edge/4

Deletes the edge between Vertex1 and Vertex2 from the graph. The graph is unchanged if the edge does not exist.

Compilation flags:
static

Template:
delete_edge(Graph,Vertex1,Vertex2,NewGraph)
Mode and number of proofs:
delete_edge(+graph,+vertex,+vertex,-graph) - one

---

complement/2

Unifies NewGraph with the complement graph where there is an edge between all pairs of vertices not connected in the original graph.

Compilation flags:
    static

Template:
    complement(Graph,NewGraph)
Mode and number of proofs:
    complement(+graph,-graph) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.86.11 unweighted_undirected_graph

Unweighted undirected graph predicates using the AVL tree dictionary representation.

Availability:
    logtalk_load(graphs(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-19

Compilation flags:
    static, context_switching_calls

Extends:
public unweighted_undirected_graph(avltree)

Remarks:
(none)

Inherited public predicates:
add_edge/4 add_edges/3 add_vertex/3 add_vertices/3 all_pairs_min_paths/2 all_pairs_min_predecessors/2 articulation_points/2 breadth_first_order/3 bridges/2 complement/2 connected_components/2 cycle/2 degree/3 delete_edge/4 delete_edges/3 delete_vertex/3 delete_vertices/3 depth_first_order/3 edge/3 edges/2 empty/1 graph_coloring/3 has_cycle/1 has_path/3 is_bipartite/1 is_complete/1 is_connected/1 is_sparse/1 is_tree/1 max_path/5 maximal_cliques/2 maximum_cliques/2 min_distances/3 min_path/5 min_predecessors/3 neighbors/3 new/1 new/2 new/3 number_of_edges/2 number_of_vertices/2 path/3 reachable/3 vertices/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.86.12 unweighted_undirected_graph(Dictionary)

Unweighted undirected graph predicates using a dictionary representation. Undirected edges are stored as two directed edges. The parametric object parameter is the dictionary to use for the graph representation.

Availability:
logtalk_load(graphs(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-20

Compilation flags:
static, context_switching_calls

Imports:
public unweighted_graph_common(Dictionary)
public undirected_graph_common
Uses:
list
set

Remarks:
(none)

Inherited public predicates:
add_edge/4 add_edges/3 add_vertex/3 add_vertices/3 all_pairs_min_paths/2 all_pairs_min_predecessors/2 articulation_points/2 breadth_first_order/3 bridges/2 complement/2 cycle/2 delete_edge/4 delete_edges/3 delete_vertex/3 delete_vertices/3 depth_first_order/3 edge/3 edges/2 empty/1 graph_coloring/3 has_cycle/1 has_path/3 is_bipartite/1 is_complete/1 is_sparse/1 is_tree/1 max_path/5 maximal_cliques/2 maximum_cliques/2 min_distances/3 min_path/5 min_predecessors/3 neighbors/3 new/1 new/2 new/3 number_of_edges/2 number_of_vertices/2 path/3 reachable/3 vertices/2

- Public predicates
  - degree/3
  - is_connected/1
  - connected_components/2
- Protected predicates
- Private predicates

- Operators

Public predicates

degree/3

Unifies Degree with the number of edges incident to Vertex. Fails if Vertex is not in the graph.

Compilation flags:
    static

Template:
    degree(Vertex,Graph,Degree)
Mode and number of proofs:
    degree(+vertex,+graph,-integer) - zero_or_one

---

is_connected/1

True iff the graph is connected (every vertex is reachable from every other vertex).

Compilation flags:
    static

Template:
    is_connected(Graph)
Mode and number of proofs:
    is_connected(+graph) - zero_or_one

---

connected_components/2

Unifies Components with a list of connected components. Each component is a sorted list of vertices.

Compilation flags:
    static

Template:
connected_components(Graph,Components)
Mode and number of proofs:
connected_components(+graph,-list(list(vertex))) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.86.13 weighted_directed_graph

Weighted directed graph predicates using the AVL tree dictionary representation. Edge weights use a pair representation (Vertex-Weight in neighbor lists, (Vertex1-Vertex2)-Weight for edge lists).

Availability:
logtalk_load(graphs(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-19

Compilation flags:
static, context_switching_calls

Extends:
public weighted_directed_graph(avltree)

Remarks:
(none)

Inherited public predicates:

add_edge/5 add_edges/3 add_vertex/3 add_vertices/3 all_pairs_min_paths/2 all_pairs_min_predecessors/2 breadth_first_order/3 cycle/2 delete_edge/5 delete_edges/3 delete_vertex/3 delete_vertices/3 depth_first_order/3 edge/4 edges/2 empty/1 has_cycle/1 has_negative_cycle/1 has_path/3 in_degree/3 is_acyclic/1 is_bipartite/1 is_complete/1 is_sparse/1 max_path/5 min_distances/3 min_path/5 min_path_bellman_ford/5 min_paths/3 min_predecessors/3 neighbors/3 new/1 new/2 new/3 number_of_edges/2 number_of_vertices/2 out_degree/3 path/3 reachable/3 strongly_connected_components/2 symmetric_closure/2 topological_sort/2 transitive_closure/2 transpose/2 vertices/2 weakly_connected_components/2 wneighbors/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.86.14 weighted_directed_graph(Dictionary)

Weighted directed graph predicates using a dictionary representation. Edge weights use a pair representation (Vertex-Weight in neighbor lists, (Vertex1-Vertex2)-Weight for edge lists). The parametric object parameter is the dictionary to use for the graph representation.

Availability:
logtalk_load(graphs(loader))

Author: Paulo Moura

Version: 1:0:0
Date: 2026-02-20

Compilation flags:
static, context_switching_calls

Imports:
public weighted_graph_common(Dictionary)
public directed_graph_common
Uses:
pairs

Remarks:
(none)

Inherited public predicates:
add_edge/5 add_edges/3 add_vertex/3 add_vertices/3 all_pairs_min_paths/2 all_pairs_min_predecessors/2 breadth_first_order/3 cycle/2 delete_edge/5 delete_edges/3 delete_vertex/3 delete_vertices/3 depth_first_order/3 edge/4 edges/2 empty/1 has_cycle/1 has_negative_cycle/1 has_path/3 in_degree/3 is_acyclic/1 is_bipartite/1 is_complete/1 is_sparse/1 max_path/5 min_distances/3 min_path/5 min_path_bellman_ford/5 min_predecessors/3 neighbors/3 new/1 new/2 new/3 number_of_edges/2 number_of_vertices/2 out_degree/3 path/3 reachable/3 strongly_connected_components/2 symmetric_closure/2 topological_sort/2 transitive_closure/2 transpose/2 vertices/2 weakly_connected_components/2 wneighbors/3

- Public predicates
  - min_paths/3
- Protected predicates
- Private predicates
- Operators

Public predicates

min_paths/3

Computes shortest path tree from Vertex1 to all reachable vertices.

Compilation flags:
static

Template:
    min_paths(Vertex1,Graph,PathTree)
Mode and number of proofs:
    min_paths(+vertex,+graph,-graph) - zero_or_one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.86.15 weighted_graph_common(Dictionary)

Common weighted graph predicates shared by both weighted directed and weighted undirected graph objects. Uses self-dispatch to call object-specific predicates such as add_edge/5, delete_edge/5, and edges/2.

Availability:
    logtalk_load(graphs(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-25

Compilation flags:
    static

Implements:
    public weighted_graph_protocol
Extends:
    public graph_common
Uses:
    list
    pairs

Remarks:
(none)

Inherited public predicates:
add_edge/5 add_edges/3 add_vertex/3 add_vertices/3 all_pairs_min_paths/2 all_pairs_min_predecessors/2 breadth_first_order/3 delete_edge/5 delete_edges/3 delete_vertex/3 delete_vertices/3 depth_first_order/3 edge/4 edges/2 empty/1 has_negative_cycle/1 has_path/3 is_bipartite/1 is_complete/1 is_sparse/1 max_path/5 min_distances/3 min_path/5 min_path_bellman_ford/5 min_predecessors/3 neighbors/3 new/1 new/2 new/3 number_of_edges/2 number_of_vertices/2 path/3 reachable/3 vertices/2 wneighbors/3

- Public predicates
- Protected predicates
  - winsert_neighbor/4
  - wremove_neighbor/4
  - wfind/3
  - wremove_vertex_from_all/3
  - relax_neighbors/7
  - pq_insert/3
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

winsert_neighbor/4

Inserts a weighted neighbor into a sorted weighted neighbor list, replacing any existing entry for the same vertex.

Compilation flags:
static

Template:
winsert_neighbor(WNeighbors,Vertex,Weight,NewWNeighbors)
Mode and number of proofs:

winsert_neighbor(+list,+vertex,+number,-list) - one

---

wremove_neighbor/4

Removes a vertex from a sorted weighted neighbor list, unifying the weight. Fails if the vertex is not found.

Compilation flags:
static

Template:
wremove_neighbor(WNeighbors,Vertex,Weight,NewWNeighbors)
Mode and number of proofs:
wremove_neighbor(+list,+vertex,-number,-list) - zero_or_one

---

wfind/3

Finds the weight associated with a vertex in a weighted neighbor list.

Compilation flags:
static

Template:
wfind(WNeighbors,Vertex,Weight)
Mode and number of proofs:
wfind(+list,+vertex,-number) - zero_or_one

---

wremove_vertex_from_all/3

Removes a vertex from all weighted neighbor lists in a list of vertex-neighbors pairs.

Compilation flags:
static

Template:

wremove_vertex_from_all(Pairs,Vertex,NewPairs)
Mode and number of proofs:
wremove_vertex_from_all(+list(pair),+vertex,-list(pair)) - one

---

relax_neighbors/7

Relaxes neighbors during Dijkstra shortest path computation, updating distances and priority queue.

Compilation flags:
static

Template:
relax_neighbors(WNeighbors,Vertex,Distance,Queue,Dist,NewQueue,NewDist)
Mode and number of proofs:
relax_neighbors(+list,+vertex,+number,+list,+dictionary,-list,-dictionary) - one

---

pq_insert/3

Inserts an element into a sorted priority queue (list of Distance-Vertex pairs).

Compilation flags:
static

Template:
pq_insert(Queue,Item,NewQueue)
Mode and number of proofs:
pq_insert(+list,+pair,-list) - one

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.86.16 weighted_graph_protocol

Protocol for weighted graph predicates, extending the common graph protocol with weighted edge operations.

Availability:
logtalk_load(graphs(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-25

Compilation flags:
static

Extends:
public graph_protocol

Remarks:
(none)

Inherited public predicates:
add_edges/3 add_vertex/3 add_vertices/3 all_pairs_min_paths/2 all_pairs_min_predecessors/2 breadth_first_order/3 delete_edges/3 delete_vertex/3 delete_vertices/3 depth_first_order/3 edges/2 empty/1 has_path/3 is_bipartite/1 is_complete/1 is_sparse/1 max_path/5 min_distances/3 min_path/5 min_predecessors/3 neighbors/3 new/1 new/2 new/3 number_of_edges/2 number_of_vertices/2 path/3 reachable/3 vertices/2

- Public predicates
  - edge/4
  - add_edge/5

    - delete_edge/5
    - wneighbors/3
    - min_path_bellman_ford/5
    - has_negative_cycle/1
- Protected predicates
- Private predicates
- Operators

Public predicates

edge/4

True iff there is an edge between Vertex1 and Vertex2 with weight Weight in Graph.

Compilation flags:
    static

Template:
    edge(Vertex1,Vertex2,Weight,Graph)
Mode and number of proofs:
    edge(+vertex,+vertex,?number,+graph) - zero_or_one

---

add_edge/5

Adds a weighted edge between Vertex1 and Vertex2 with weight Weight.

Compilation flags:
    static

Template:
    add_edge(Graph,Vertex1,Vertex2,Weight,NewGraph)
Mode and number of proofs:
    add_edge(+graph,+vertex,+vertex,+number,-graph) - one

---

delete_edge/5

Deletes the weighted edge between Vertex1 and Vertex2. Unifies Weight with the weight of the deleted edge. The graph is unchanged if the edge does not exist.

Compilation flags:
    static

Template:
    delete_edge(Graph,Vertex1,Vertex2,Weight,NewGraph)
Mode and number of proofs:
    delete_edge(+graph,+vertex,+vertex,?number,-graph) - one

---

wneighbors/3

Unifies WNeighbors with a list of NeighborVertex-Weight pairs for the neighbors of Vertex. Fails if Vertex is not in the graph.

Compilation flags:
    static

Template:
    wneighbors(Vertex,Graph,WNeighbors)
Mode and number of proofs:
    wneighbors(+vertex,+graph,-list(pair)) - zero_or_one

---

min_path_bellman_ford/5

Finds the minimum cost path from Vertex1 to Vertex2 using the Bellman-Ford algorithm. Supports negative edge weights and fails if no path exists or if a reachable negative cycle makes the optimum undefined.

Compilation flags:
    static

Template:
    min_path_bellman_ford(Vertex1,Vertex2,Graph,Path,Cost)
Mode and number of proofs:

min_path_bellman_ford(+vertex,+vertex,+graph,-list(vertex),-number) - zero_or_one

---

has_negative_cycle/1

True iff Graph contains a negative-weight cycle.

Compilation flags:
static

Template:
has_negative_cycle(Graph)
Mode and number of proofs:
has_negative_cycle(+graph) - zero_or_one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.86.17 weighted_undirected_graph

Weighted undirected graph predicates using the AVL tree dictionary representation. Each edge is stored in both directions. Edge weights use a pair representation (Vertex-Weight in neighbor lists, (V1-V2)-Weight for edge lists).

Availability:
logtalk_load(graphs(loader))

Author: Paulo Moura
Version: 1:0:0

Date: 2026-02-19

Compilation flags:
static, context_switching_calls

Extends:
public weighted_undirected_graph(avltree)

Remarks:
(none)

Inherited public predicates:
add_edge/5 add_edges/3 add_vertex/3 add_vertices/3 all_pairs_min_paths/2 all_pairs_min_predecessors/2 articulation_points/2 breadth_first_order/3 bridges/2 connected_components/2 cycle/2 degree/3 delete_edge/5 delete_edges/3 delete_vertex/3 delete_vertices/3 depth_first_order/3 edge/4 edges/2 empty/1 graph_coloring/3 has_cycle/1 has_negative_cycle/1 has_path/3 is_bipartite/1 is_complete/1 is_connected/1 is_sparse/1 is_tree/1 max_path/5 max_tree/3 maximal_cliques/2 maximum_cliques/2 min_distances/3 min_path/5 min_path_bellman_ford/5 min_predecessors/3 min_tree/3 neighbors/3 new/1 new/2 new/3 number_of_edges/2 number_of_vertices/2 path/3 reachable/3 vertices/2 wneighbors/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.86.18 weighted_undirected_graph(Dictionary)

Weighted undirected graph predicates using a dictionary representation. Each edge is stored in both directions. Edge weights use a pair representation (Vertex-Weight in neighbor lists, (Vertex1-Vertex2)-Weight for edge lists). The parametric object parameter is the dictionary to use for the graph representation.

Availability:
logtalk_load(graphs(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-20

Compilation flags:
static, context_switching_calls

Imports:
public weighted_graph_common(Dictionary)
public undirected_graph_common
Uses:
list

Remarks:
(none)

Inherited public predicates:
add_edge/5 add_edges/3 add_vertex/3 add_vertices/3 all_pairs_min_paths/2 all_pairs_min_predecessors/2 articulation_points/2 breadth_first_order/3 bridges/2 cycle/2 delete_edge/5 delete_edges/3 delete_vertex/3 delete_vertices/3 depth_first_order/3 edge/4 edges/2 empty/1 graph_coloring/3 has_cycle/1 has_negative_cycle/1 has_path/3 is_bipartite/1 is_complete/1 is_sparse/1 is_tree/1 max_path/5 maximal_cliques/2 maximum_cliques/2 min_distances/3 min_path/5 min_path_bellman_ford/5 min_predecessors/3 neighbors/3 new/1 new/2 new/3 number_of_edges/2 number_of_vertices/2 path/3 reachable/3 vertices/2 wneighbors/3

- Public predicates
  - degree/3
  - is_connected/1
  - connected_components/2
  - min_tree/3
  - max_tree/3
- Protected predicates
- Private predicates
- Operators

Public predicates

degree/3

Returns the degree (number of edges incident to the vertex) of Vertex in Graph.

Compilation flags:
    static

Template:
    degree(Vertex,Graph,Degree)
Mode and number of proofs:
    degree(+vertex,+graph,-integer) - zero_or_one

---

is_connected/1

True if Graph is connected (all vertices are reachable from any vertex).

Compilation flags:
    static

Template:
    is_connected(Graph)
Mode and number of proofs:
    is_connected(+graph) - zero_or_one

---

connected_components/2

Returns the list of connected components (each a list of vertices).

Compilation flags:
    static

Template:
    connected_components(Graph,Components)
Mode and number of proofs:
    connected_components(+graph,-list(list)) - one

---

min_tree/3

Constructs a minimum spanning tree and returns its total weight.

Compilation flags:
    static

Template:
    min_tree(Graph,Tree,Cost)
Mode and number of proofs:
    min_tree(+graph,-graph,-number) - zero_or_one

---

max_tree/3

Constructs a maximum spanning tree and returns its total weight.

Compilation flags:
    static

Template:
    max_tree(Graph,Tree,Cost)
Mode and number of proofs:

max_tree(+graph,-graph,-number) - zero_or_one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.87 gsp_pattern_miner

object

1.87.1 gsp_pattern_miner

GSP sequential pattern miner for sequence datasets using level-wise candidate generation and pruning.

Availability:
logtalk_load(gsp_pattern_miner(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public sequential_pattern_mining_common
Uses:
avltree
format
list

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_pattern_miner/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 mine/2 mine/3 option/2 option/3 pattern_miner_options/2 print_pattern_miner/1 valid_option/1 valid_options/1 valid_pattern_miner/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

pattern_miner_protocol, sequence_dataset_protocol, prefix_span_pattern_miner, spade_pattern_miner, clo_span_pattern_miner

1.88 hashes

object

1.88.1 crc32_non_reflected(Polynomial,Initial,FinalXor,AppendLength)

- Polynomial - Canonical non-reflected CRC-32 polynomial.
- Initial - Initial CRC accumulator value.
- FinalXor - Final xor value.
- AppendLength - Boolean flag controlling whether the message length is appended as little-endian bytes.

Parametric non-reflected CRC-32 hash function using a canonical polynomial, configurable initial value and final xor value, and optional appended little-endian length bytes.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
static, context_switching_calls

Implements:
public hash_protocol
Uses:
hash_common_32
list

Remarks:
(none)

Inherited public predicates:
hash/2

- Public predicates
- Protected predicates
- Private predicates

- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

crc32_reflected(Polynomial), crc32b, crc32c, crc32posix, crc32mpeg2, crc32bzip2, crc32q, murmurhash3_x86_32, fnv1a_32

object

1.88.2 crc32_reflected(Polynomial)

- Polynomial - Reflected CRC-32 polynomial.

Parametric reflected CRC-32 hash function using initial value 0xFFFFFFFF and final xor 0xFFFFFFFF.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
static, context_switching_calls

Implements:

public hash_protocol

Uses:

hash_common_32

Remarks:

(none)

Inherited public predicates:

hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

crc32_non_reflected(Polynomial,Initial,FinalXor,AppendLength), crc32b, crc32c, crc32posix, crc32mpeg2, crc32bzip2, crc32q, murmurhash3_x86_32, fnv1a_32

object

1.88.3 crc32b

CRC-32/ISO-HDLC hash function using the reflected polynomial 0xEDB88320, initial value 0xFFFFFFFF, and final xor 0xFFFFFFFF.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
static, context_switching_calls

Extends:
public crc32_reflected(3988292384)

Remarks:
(none)

Inherited public predicates:
hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

crc32_reflected(Polynomial), crc32c, crc32posix, crc32mpeg2, crc32bzip2, crc32q, murmurhash3_x86_32, fnv1a_32

object

1.88.4 crc32bzip2

CRC-32/BZIP2 hash function using the canonical polynomial 0x04C11DB7, initial value 0xFFFFFFFF, and final xor 0xFFFFFFFF.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
static, context_switching_calls

Extends:
public crc32_non_reflected(79764919,4294967295,4294967295,false)

Remarks:
(none)

Inherited public predicates:
hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

crc32_non_reflected(Polynomial,Initial,FinalXor,AppendLength), crc32mpeg2, crc32posix, crc32q, crc32_reflected(Polynomial), crc32b, crc32c, murmurhash3_x86_32, fnv1a_32

object

1.88.5 crc32c

CRC-32C/Castagnoli hash function using the reflected polynomial 0x82F63B78, initial value 0xFFFFFFFF, and final xor 0xFFFFFFFF.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:

static, context_switching_calls

Extends:
public crc32_reflected(2197175160)

Remarks:
(none)

Inherited public predicates:
hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

crc32_reflected(Polynomial), crc32b, crc32posix, crc32mpeg2, crc32bzip2, crc32q, murmurhash3_x86_32, fnv1a_32

object

1.88.6 crc32mpeg2

CRC-32/MPEG-2 hash function using the canonical polynomial 0x04C11DB7, initial value 0xFFFFFFFF, and final xor 0x00000000.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
static, context_switching_calls

Extends:
public crc32_non_reflected(79764919,4294967295,0,false)

Remarks:
(none)

Inherited public predicates:
hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

crc32_non_reflected(Polynomial,Initial,FinalXor,AppendLength), crc32posix, crc32bzip2, crc32q, crc32_reflected(Polynomial), crc32b, crc32c, murmurhash3_x86_32, fnv1a_32

object

1.88.7 crc32posix

CRC-32/POSIX (cksum) hash function using the canonical polynomial 0x04C11DB7, initial value 0x00000000, appended little-endian length bytes, and final xor 0xFFFFFFFF.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:
static, context_switching_calls

Extends:
public crc32_non_reflected(79764919,0,4294967295,true)

Remarks:
(none)

Inherited public predicates:
hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

crc32_non_reflected(Polynomial,Initial,FinalXor,AppendLength), crc32_reflected(Polynomial), crc32b, crc32c, crc32mpeg2, crc32bzip2, crc32q, murmurhash3_x86_32, fnv1a_32

object

1.88.8 crc32q

CRC-32Q hash function, also used by AIXM-style formats, using the canonical polynomial 0x814141AB, initial value 0x00000000, and final xor 0x00000000.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-05

Compilation flags:

static, context_switching_calls

Extends:
public crc32_non_reflected(2168537515,0,0,false)

Remarks:
(none)

Inherited public predicates:
hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

crc32_non_reflected(Polynomial,Initial,FinalXor,AppendLength), crc32mpeg2, crc32bzip2, crc32posix, crc32_reflected(Polynomial), crc32b, crc32c, murmurhash3_x86_32, fnv1a_32

object

1.88.9 djb2_32

DJB2 32-bit hash function.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-04

Compilation flags:
static, context_switching_calls

Implements:
public hash_protocol
Uses:
hash_common_32

Remarks:
(none)

Inherited public predicates:
hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

djb2_64, sdbm_32, fnv1a_32

object

1.88.10 djb2_64

DJB2 64-bit hash function.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-04

Compilation flags:
static, context_switching_calls

Implements:
public hash_protocol
Uses:
hash_common_64

Remarks:
(none)

Inherited public predicates:
hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

djb2_32, sdbm_64, fnv1a_64

object

1.88.11 fips202_fixed_hash(RateBytes,Suffix,OutputBytes)

Common implementation of the fixed-size FIPS 202 SHA-3 variants suitable for use with HMAC.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-04-15

Compilation flags:
static, context_switching_calls

Implements:
public hash_digest_protocol
Extends:
public fips202_hash(RateBytes,Suffix,OutputBytes)

Remarks:
(none)

Inherited public predicates:
block_size/1 digest/2 digest_size/1 hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.88.12 fips202_hash(RateBytes,Suffix,OutputBytes)

Common implementation of the standardized FIPS 202 SHA-3 and SHAKE variants using the Keccak-f[1600] permutation.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-04-15

Compilation flags:
static, context_switching_calls

Implements:
public hash_protocol
Uses:
hash_common_32
hash_common_64
list

Remarks:
(none)

Inherited public predicates:
hash/2

- Public predicates
- Protected predicates
  - absorb/2
  - squeeze/3
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

absorb/2

Absorbs the input bytes into the Keccak state and returns the resulting state after the final padded block permutation.

Compilation flags:
    static

Template:
    absorb(Bytes,State)
Mode and number of proofs:
    absorb(+list(byte),--list(integer)) - one

---

squeeze/3

Squeezes the requested number of output bytes from a Keccak state.

Compilation flags:
    static

Template:
    squeeze(State,OutputBytes,DigestBytes)
Mode and number of proofs:
    squeeze(+list(integer),+integer,--list(byte)) - one

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.88.13 fnv1a_32

FNV-1a 32-bit hash function.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-04

Compilation flags:
static, context_switching_calls

Implements:
public hash_protocol
Uses:
hash_common_32

Remarks:
(none)

Inherited public predicates:
hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

fnv1a_64, djb2_32, sdbm_32

object

1.88.14 fnv1a_64

FNV-1a 64-bit hash function.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-04

Compilation flags:
static, context_switching_calls

Implements:
public hash_protocol
Uses:
hash_common_64

Remarks:
(none)

Inherited public predicates:
hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

fnv1a_32, djb2_64, sdbm_64

object

1.88.15 hash_common_32

Auxiliary predicates for the hashes library 32-bit algorithms.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-04

Compilation flags:
static, context_switching_calls

Uses:
list

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - word32_hex/2
  - bytes_hex/2
  - mask32/1
  - add32/3
  - add32/4
  - add32/5
  - mul32/3
  - rol32/3
  - ror32/3
  - little_endian_word32/2
  - big_endian_word32/2
  - integer_to_little_endian_bytes32/2
  - integer_to_big_endian_bytes32/2
  - pad_md/4
- Protected predicates
- Private predicates
- Operators

Public predicates

word32_hex/2

Converts a 32-bit word into an 8-digit lowercase hexadecimal atom.

Compilation flags:
    static

Template:
    word32_hex(Word,Hex)
Mode and number of proofs:
    word32_hex(+integer,-atom) - one

---

bytes_hex/2

Converts a list of bytes into a lowercase hexadecimal atom.

Compilation flags:
    static

Template:
    bytes_hex(Bytes,Hex)
Mode and number of proofs:
    bytes_hex(+list(integer),-atom) - one

---

mask32/1

Returns the 32-bit mask value.

Compilation flags:
    static

Template:
    mask32(Mask)
Mode and number of proofs:
    mask32(-integer) - one

---

add32/3

Adds two integers modulo 2^32.

Compilation flags:
    static

Template:
    add32(A,B,Sum)
Mode and number of proofs:
    add32(+integer,+integer,-integer) - one

---

add32/4

Adds three integers modulo 2^32.

Compilation flags:
    static

Template:
    add32(A,B,C,Sum)
Mode and number of proofs:
    add32(+integer,+integer,+integer,-integer) - one

---

add32/5

Adds four integers modulo 2^32.

Compilation flags:
    static

Template:
    add32(A,B,C,D,Sum)
Mode and number of proofs:

add32(+integer,+integer,+integer,+integer,-integer) - one

---

mul32/3

Multiplies two integers modulo 2^32.

Compilation flags:
static

Template:
mul32(A,B,Product)
Mode and number of proofs:
mul32(+integer,+integer,-integer) - one

---

rol32/3

Rotates a 32-bit word left by the given number of bits.

Compilation flags:
static

Template:
rol32(Value,Shift,Rotated)
Mode and number of proofs:
rol32(+integer,+integer,-integer) - one

---

ror32/3

Rotates a 32-bit word right by the given number of bits.

Compilation flags:
static

Template:

ror32(Value,Shift,Rotated)
Mode and number of proofs:
ror32(+integer,+integer,-integer) - one

---

little_endian_word32/2

Decodes four bytes in little-endian order into a 32-bit word.

Compilation flags:
static

Template:
little_endian_word32(Bytes,Word)
Mode and number of proofs:
little_endian_word32(+list(integer),-integer) - one

---

big_endian_word32/2

Decodes four bytes in big-endian order into a 32-bit word.

Compilation flags:
static

Template:
big_endian_word32(Bytes,Word)
Mode and number of proofs:
big_endian_word32(+list(integer),-integer) - one

---

integer_to_little_endian_bytes32/2

Encodes a 32-bit word into four bytes in little-endian order.

Compilation flags:
    static

Template:
    integer_to_little_endian_bytes32(Integer,Bytes)
Mode and number of proofs:
    integer_to_little_endian_bytes32(+integer,-list(integer)) - one

---

integer_to_big_endian_bytes32/2

Encodes a 32-bit word into four bytes in big-endian order.

Compilation flags:
    static

Template:
    integer_to_big_endian_bytes32(Integer,Bytes)
Mode and number of proofs:
    integer_to_big_endian_bytes32(+integer,-list(integer)) - one

---

pad_md/4

Pads a message using MD-style padding with a little-endian or big-endian length field.

Compilation flags:
    static

Template:
    pad_md(Endian,Bytes,LengthFieldBytes,PaddedBytes)
Mode and number of proofs:
    pad_md(+little_big,+list(integer),+integer,-list(integer)) - one

---

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.88.16 hash_common_64

Auxiliary predicates for the hashes library 64-bit algorithms.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-04

Compilation flags:
static, context_switching_calls

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - word64_hex/2
  - mask64/1

      - add64/3
      - mul64/3
      - rol64/3
      - xor64/3
      - or64/3
      - and64/3
      - not64/2
      - shl64/3
      - shr64/3
      - integer_to_big_endian_bytes64/2
  - Protected predicates
  - Private predicates
  - Operators

Public predicates

word64_hex/2

Converts a 64-bit word into a 16-digit lowercase hexadecimal atom.

Compilation flags:
    static

Template:
    word64_hex(Word,Hex)
Mode and number of proofs:
    word64_hex(+integer,-atom) - one

---

mask64/1

Returns the 64-bit mask value.

Compilation flags:
    static

Template:

mask64(Mask)
Mode and number of proofs:
mask64(-integer) - one

---

add64/3

Adds two integers modulo 2^64.

Compilation flags:
static

Template:
add64(A,B,Sum)
Mode and number of proofs:
add64(+integer,+integer,-integer) - one

---

mul64/3

Multiplies two integers modulo 2^64.

Compilation flags:
static

Template:
mul64(A,B,Product)
Mode and number of proofs:
mul64(+integer,+integer,-integer) - one

---

rol64/3

Rotates a 64-bit word left by the given number of bits.

Compilation flags:
    static

Template:
    rol64(Value,Shift,Rotated)
Mode and number of proofs:
    rol64(+integer,+integer,-integer) - one

---

xor64/3

Computes the bitwise exclusive-or of two integers modulo 2^64.

Compilation flags:
    static

Template:
    xor64(A,B,Xor)
Mode and number of proofs:
    xor64(+integer,+integer,-integer) - one

---

or64/3

Computes the bitwise disjunction of two integers modulo 2^64.

Compilation flags:
    static

Template:
    or64(A,B,Or)
Mode and number of proofs:
    or64(+integer,+integer,-integer) - one

---

and64/3

Computes the bitwise conjunction of two integers modulo 2^64.

Compilation flags:
    static

Template:
    and64(A,B,And)
Mode and number of proofs:
    and64(+integer,+integer,-integer) - one

---

not64/2

Computes the bitwise complement of an integer modulo 2^64.

Compilation flags:
    static

Template:
    not64(Value,Complement)
Mode and number of proofs:
    not64(+integer,-integer) - one

---

shl64/3

Shifts a 64-bit word left by the given number of bits and masks the result.

Compilation flags:
    static

Template:
    shl64(Value,Shift,Shifted)
Mode and number of proofs:
    shl64(+integer,+integer,-integer) - one

---

shr64/3

Shifts a 64-bit word right by the given number of bits after masking the input.

Compilation flags:
    static

Template:
    shr64(Value,Shift,Shifted)
Mode and number of proofs:
    shr64(+integer,+integer,-integer) - one

---

integer_to_big_endian_bytes64/2

Encodes a 64-bit word into eight bytes in big-endian order.

Compilation flags:
    static

Template:
    integer_to_big_endian_bytes64(Integer,Bytes)
Mode and number of proofs:
    integer_to_big_endian_bytes64(+integer,-list(integer)) - one

---

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

protocol

1.88.17 hash_digest_protocol

Protocol for fixed-size cryptographic hash functions exposing raw digests and digest metadata.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-16

Compilation flags:
static

Extends:
public hash_protocol

Remarks:
(none)

Inherited public predicates:
hash/2

- Public predicates
  - digest/2
  - digest_size/1
  - block_size/1
- Protected predicates
- Private predicates
- Operators

Public predicates

digest/2

Computes the digest for a list of bytes and returns it as a list of bytes.

Compilation flags:
    static

Template:
    digest(Bytes,Digest)
Mode and number of proofs:
    digest(+list(byte),--list(byte)) - one

---

digest_size/1

Returns the digest size in bytes.

Compilation flags:
    static

Template:
    digest_size(DigestSize)
Mode and number of proofs:
    digest_size(--integer) - one

---

block_size/1

Returns the hash block size in bytes.

Compilation flags:
    static

Template:
    block_size(BlockSize)
Mode and number of proofs:
    block_size(--integer) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.88.18 hash_protocol

Hashing protocol. Hash values are returned as lowercase hexadecimal atoms using the output width of each algorithm or extensible-output function instance.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-04

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - hash/2
- Protected predicates
- Private predicates
- Operators

Public predicates

hash/2

Computes the hash for a list of bytes.

Compilation flags:
    static

Template:
    hash(Bytes,Hash)
Mode and number of proofs:
    hash(+list(byte),--atom) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

object

1.88.19 md5

MD5 hash function.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-04-15

Compilation flags:
static, context_switching_calls

Implements:
public hash_digest_protocol
Uses:
hash_common_32
list

Remarks:
(none)

Inherited public predicates:
block_size/1 digest/2 digest_size/1 hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

sha1, sha256

object

1.88.20 murmurhash3_x64_128

MurmurHash3 x64 128-bit hash function with seed 0.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-04

Compilation flags:
static, context_switching_calls

Implements:
public hash_protocol
Uses:
hash_common_64
list

Remarks:

(none)

Inherited public predicates:
hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

murmurhash3_x86_32, murmurhash3_x86_128

object

1.88.21 murmurhash3_x86_128

MurmurHash3 x86 128-bit hash function with seed 0.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura

Version: 1:0:0
Date: 2026-04-04

Compilation flags:
static, context_switching_calls

Implements:
public hash_protocol
Uses:
hash_common_32
list

Remarks:
(none)

Inherited public predicates:
hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

murmurhash3_x86_32, murmurhash3_x64_128

object

1.88.22 murmurhash3_x86_32

MurmurHash3 x86 32-bit hash function with seed 0.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-04

Compilation flags:
static, context_switching_calls

Implements:
public hash_protocol
Uses:
hash_common_32
list

Remarks:
(none)

Inherited public predicates:
hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

murmurhash3_x86_128, murmurhash3_x64_128

object

1.88.23 sdbm_32

sdbm 32-bit hash function.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-04

Compilation flags:
static, context_switching_calls

Implements:
public hash_protocol
Uses:
hash_common_32

Remarks:
(none)

Inherited public predicates:
hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

sdbm_64, djb2_32, fnv1a_32

object

1.88.24 sdbm_64

sdbm 64-bit hash function.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura

Version: 1:0:0
Date: 2026-04-04

Compilation flags:
static, context_switching_calls

Implements:
public hash_protocol
Uses:
hash_common_64

Remarks:
(none)

Inherited public predicates:
hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

sdbm_32, djb2_64, fnv1a_64

object

1.88.25 sha1

SHA-1 hash function.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-04-15

Compilation flags:
static, context_switching_calls

Implements:
public hash_digest_protocol
Uses:
hash_common_32
list

Remarks:
(none)

Inherited public predicates:
block_size/1 digest/2 digest_size/1 hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

md5, sha256

object

1.88.26 sha256

SHA-256 hash function.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-04-15

Compilation flags:
static, context_switching_calls

Implements:
public hash_digest_protocol
Uses:
hash_common_32

list

Remarks:
(none)

Inherited public predicates:
block_size/1 digest/2 digest_size/1 hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

md5, sha1

object

1.88.27 sha3_224

FIPS 202 SHA3-224 hash function.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-04-15

Compilation flags:
static, context_switching_calls

Extends:
public fips202_fixed_hash(144,6,28)

Remarks:
(none)

Inherited public predicates:
block_size/1 digest/2 digest_size/1 hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

sha3_256, sha3_384, sha3_512, shake128(OutputBytes), shake256(OutputBytes)

object

1.88.28 sha3_256

FIPS 202 SHA3-256 hash function.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-04-15

Compilation flags:
static, context_switching_calls

Extends:
public fips202_fixed_hash(136,6,32)

Remarks:
(none)

Inherited public predicates:
block_size/1 digest/2 digest_size/1 hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

sha3_224, sha3_384, sha3_512, sha256, shake128(OutputBytes), shake256(OutputBytes)

object

1.88.29 sha3_384

FIPS 202 SHA3-384 hash function.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-04-15

Compilation flags:
static, context_switching_calls

Extends:
public fips202_fixed_hash(104,6,48)

Remarks:
(none)

Inherited public predicates:
block_size/1 digest/2 digest_size/1 hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

sha3_224, sha3_256, sha3_512, shake128(OutputBytes), shake256(OutputBytes)

object

1.88.30 sha3_512

FIPS 202 SHA3-512 hash function.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-04-15

Compilation flags:
static, context_switching_calls

Extends:
public fips202_fixed_hash(72,6,64)

Remarks:
(none)

Inherited public predicates:
block_size/1 digest/2 digest_size/1 hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

sha3_224, sha3_256, sha3_384, shake128(OutputBytes), shake256(OutputBytes)

object

1.88.31 shake128(OutputBytes)

- OutputBytes - Number of output bytes to generate.

FIPS 202 SHAKE128 extensible-output function.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-04

Compilation flags:
static, context_switching_calls

Extends:
public fips202_hash(168,31,OutputBytes)

Remarks:
(none)

Inherited public predicates:
hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

shake256(OutputBytes), sha3_224, sha3_256, sha3_384, sha3_512

object

1.88.32 shake256(OutputBytes)

- OutputBytes - Number of output bytes to generate.

FIPS 202 SHAKE256 extensible-output function.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-04

Compilation flags:

static, context_switching_calls

Extends:
public fips202_hash(136,31,OutputBytes)

Remarks:
(none)

Inherited public predicates:
hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

shake128(OutputBytes), sha3_224, sha3_256, sha3_384, sha3_512

object

1.88.33 siphash_2_4

SipHash-2-4 hash function using the standard reference key 00 01 02 … 0f.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-04

Compilation flags:
static, context_switching_calls

Extends:
public siphash_2_4([0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15])

Remarks:
(none)

Inherited public predicates:
hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

siphash_2_4(Key), fnv1a_64, crc32b, crc32c

object

1.88.34 siphash_2_4(Key)

- Key - A list of 16 bytes.

SipHash-2-4 keyed hash function.

Availability:
logtalk_load(hashes(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-04

Compilation flags:
static, context_switching_calls

Implements:
public hash_protocol
Uses:
hash_common_64
list

Remarks:
(none)

Inherited public predicates:

hash/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

siphash_2_4, fnv1a_64, crc32b, crc32c

1.89 hdbscan_clusterer

object

1.89.1 hdbscan_clusterer

HDBSCAN clusterer for continuous datasets using a mutual-reachability hierarchy, condensed tree pruning, and stability-based cluster selection.

Availability:
logtalk_load(hdbscan_clusterer(loader))

Author: Paulo Moura

Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public clusterer_common
Uses:
format
list
numberlist
pairs
type
union_find

Remarks:
(none)

Inherited public predicates:
check_clusterer/1 check_option/1 check_options/1 cluster/3 clusterer_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 print_clusterer/1 valid_clusterer/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

clusterer_protocol, clustering_dataset_protocol, dbscan_clusterer, optics_clusterer

1.90 heaps

object

1.90.1 binary_heap(Order)

- Order - Either < for a min heap or > for a max heap.

Heap implementation, parameterized by the order to be used to compare keys (< or >).

Availability:
logtalk_load(heaps(loader))

Author: Richard O'Keefe; adapted to Logtalk by Paulo Moura and Victor Lagerkvist.
Version: 1:2:0
Date: 2026-01-28

Compilation flags:
static, context_switching_calls

Implements:
public heap_protocol
Extends:
public compound

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 as_heap/2 as_list/2 check/1 delete/4 depth/2 empty/1 ground/1 insert/4 insert_all/3 merge/3 new/1 numbervars/1 numbervars/3 occurs/2 singletons/2 size/2 subsumes/2 subterm/2 top/3 top_next/5 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

binary_heap_min, binary_heap_max

object

1.90.2 binary_heap_max

Max-heap implementation. Uses standard order to compare keys.

Availability:
logtalk_load(heaps(loader))

Author: Paulo Moura

Version: 1:0:0
Date: 2010-02-19

Compilation flags:
static, context_switching_calls

Extends:
public binary_heap(>)

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 as_heap/2 as_list/2 check/1 delete/4 depth/2 empty/1 ground/1 insert/4 insert_all/3 merge/3 new/1 numbervars/1 numbervars/3 occurs/2 singletons/2 size/2 subsumes/2 subterm/2 top/3 top_next/5 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

binary_heap_min, binary_heap(Order)

object

1.90.3 binary_heap_min

Min-heap implementation. Uses standard order to compare keys.

Availability:
logtalk_load(heaps(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2010-02-19

Compilation flags:
static, context_switching_calls

Extends:
public binary_heap(<)

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 as_heap/2 as_list/2 check/1 delete/4 depth/2 empty/1 ground/1 insert/4 insert_all/3 merge/3 new/1 numbervars/1 numbervars/3 occurs/2 singletons/2 size/2 subsumes/2 subterm/2 top/3 top_next/5 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

binary_heap_max, binary_heap(Order)

object

1.90.4 heap(Order)

- Order - Either < for a min heap or > for a max heap.

Heap implementation, parameterized by the order to be used to compare keys (< or >). Deprecated. Use the binary_heap/1 object instead.

Availability:
logtalk_load(heaps(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-01-28

Compilation flags:
static, context_switching_calls

Extends:
public binary_heap(Order)

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 as_heap/2 as_list/2 check/1 delete/4 depth/2 empty/1 ground/1 insert/4 insert_all/3 merge/3 new/1 numbervars/1 numbervars/3 occurs/2 singletons/2 size/2 subsumes/2 subterm/2 top/3 top_next/5 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

binary_heap(Order)

protocol

1.90.5 heap_protocol

Heap protocol. Key-value pairs are represented as Key-Value.

Availability:
logtalk_load(heaps(loader))

Author: Richard O'Keefe; adapted to Logtalk by Paulo Moura and Victor Lagerkvist.

Version: 1:0:1
Date: 2010-11-13

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - insert/4
  - insert_all/3
  - delete/4
  - merge/3
  - empty/1
  - size/2
  - as_list/2
  - as_heap/2
  - top/3
  - top_next/5
- Protected predicates
- Private predicates
- Operators

Public predicates

insert/4

Inserts the new pair into a heap, returning the updated heap.

Compilation flags:
    static

Template:
    insert(Key,Value,Heap,NewHeap)
Mode and number of proofs:
    insert(+key,+value,+heap,-heap) - one

---

insert_all/3

Inserts a list of pairs into a heap, returning the updated heap.

Compilation flags:
    static

Template:
    insert_all(List,Heap,NewHeap)
Mode and number of proofs:
    insert_all(@list(pairs),+heap,-heap) - one

---

delete/4

Deletes and returns the top pair in a heap returning the updated heap.

Compilation flags:
    static

Template:
    delete(Heap,TopKey,TopValue,NewHeap)
Mode and number of proofs:
    delete(+heap,?key,?value,-heap) - zero_or_one

---

merge/3

Merges two heaps.

Compilation flags:
    static

Template:
    merge(Heap1,Heap2,NewHeap)
Mode and number of proofs:
    merge(+heap,+heap,-heap) - one

---

empty/1

True if the heap is empty.

Compilation flags:
    static

Template:
    empty(Heap)
Mode and number of proofs:
    empty(@heap) - zero_or_one

---

size/2

Returns the number of heap elements.

Compilation flags:
    static

Template:
    size(Heap,Size)
Mode and number of proofs:

size(+heap,?integer) - zero_or_one

---

as_list/2

Returns the current set of pairs in the heap as a list, sorted into ascending order of the keys.

Compilation flags:
static

Template:
as_list(Heap,List)
Mode and number of proofs:
as_list(+heap,-list) - one

---

as_heap/2

Constructs a heap from a list of pairs.

Compilation flags:
static

Template:
as_heap(List,Heap)
Mode and number of proofs:
as_heap(+list,-heap) - one

---

top/3

Returns the top pair in the heap. Fails if the heap is empty.

Compilation flags:
static

Template:

top(Heap,TopKey,TopValue)
Mode and number of proofs:
top(+heap,?key,?value) - zero_or_one

---

top_next/5

Returns the top pair and the next pair in the heap. Fails if the heap does not have at least two elements.

Compilation flags:
static

Template:
top_next(Heap,TopKey,TopValue,NextKey,NextValue)
Mode and number of proofs:
top_next(+heap,?key,?value,?key,?value) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

heap(Order), pairing_heap(Order)

protocol

1.90.6 heapp

Heap protocol. Deprecated. Use the heap_protocol protocol instead.

Availability:
logtalk_load(heaps(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-01-28

Compilation flags:
static

Extends:
public heap_protocol

Remarks:
(none)

Inherited public predicates:
as_heap/2 as_list/2 delete/4 empty/1 insert/4 insert_all/3 merge/3 size/2 top/3 top_next/5

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

heap_protocol

object

1.90.7 maxheap

Max-heap implementation. Uses standard order to compare keys. Deprecated. Use the binary_heap_max object instead.

Availability:
logtalk_load(heaps(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2010-02-19

Compilation flags:
static, context_switching_calls

Extends:
public binary_heap(>)

Remarks:
(none)

Inherited public predicates:

(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 as_heap/2 as_list/2 check/1 delete/4 depth/2 empty/1 ground/1 insert/4 insert_all/3 merge/3 new/1 numbervars/1 numbervars/3 occurs/2 singletons/2 size/2 subsumes/2 subterm/2 top/3 top_next/5 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

binary_heap_max

object

1.90.8 minheap

Min-heap implementation. Uses standard order to compare keys. Deprecated. Use the binary_heap_min object instead.

Availability:
logtalk_load(heaps(loader))

Author: Paulo Moura
Version: 1:0:0

Date: 2010-02-19

Compilation flags:
static, context_switching_calls

Extends:
public binary_heap(<)

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 as_heap/2 as_list/2 check/1 delete/4 depth/2 empty/1 ground/1 insert/4 insert_all/3 merge/3 new/1 numbervars/1 numbervars/3 occurs/2 singletons/2 size/2 subsumes/2 subterm/2 top/3 top_next/5 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

binary_heap_min

object

1.90.9 pairing_heap(Order)

- Order - Either < for a min heap or > for a max heap.

Pairing heap implementation, parameterized by the order to be used to compare keys (< or >).

Availability:
logtalk_load(heaps(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-04-17

Compilation flags:
static, context_switching_calls

Implements:
public heap_protocol
Extends:
public compound

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 as_heap/2 as_list/2 check/1 delete/4 depth/2 empty/1 ground/1 insert/4 insert_all/3 merge/3 new/1 numbervars/1 numbervars/3 occurs/2 singletons/2 size/2 subsumes/2 subterm/2 top/3 top_next/5 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

pairing_heap_min, pairing_heap_max

object

1.90.10 pairing_heap_max

Max-pairing heap implementation. Uses standard order to compare keys.

Availability:
logtalk_load(heaps(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-01-28

Compilation flags:
static, context_switching_calls

Extends:
public pairing_heap(>)

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 as_heap/2 as_list/2 check/1 delete/4 depth/2 empty/1 ground/1 insert/4 insert_all/3 merge/3 new/1 numbervars/1 numbervars/3 occurs/2 singletons/2 size/2 subsumes/2 subterm/2 top/3 top_next/5 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

pairing_heap_min, pairing_heap(Order)

object

1.90.11 pairing_heap_min

Min-pairing heap implementation. Uses standard order to compare keys.

Availability:
logtalk_load(heaps(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-01-28

Compilation flags:
static, context_switching_calls

Extends:
public pairing_heap(<)

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 as_heap/2 as_list/2 check/1 delete/4 depth/2 empty/1 ground/1 insert/4 insert_all/3 merge/3 new/1 numbervars/1 numbervars/3 occurs/2 singletons/2 size/2 subsumes/2 subterm/2 top/3 top_next/5 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

pairing_heap_max, pairing_heap(Order)

1.91 help

object

1.91.1 help

Command-line help for Logtalk tools, libraries, entities, predicates, and non-terminals.

Availability:
logtalk_load(help(loader))

Author: Paulo Moura
Version: 0:43:0
Date: 2026-02-04

Compilation flags:
static, context_switching_calls, complements(allow)

Implements:
public forwarding
Uses:
atom
integer
list
os
user

Remarks:
(none)

Inherited public predicates:
forward/1

- Public predicates
  - help/0
  - handbook/0
  - apis/0
  - apis/1
  - tools/0
  - tool/1
  - libraries/0
  - library/1
  - entity/1
  - (/)/2
  - (//)/2
  - man/1
  - completion/2
  - completions/2
  - built_in_directive/4
  - built_in_predicate/4
  - built_in_method/4
  - control_construct/4
  - built_in_non_terminal/4
- Protected predicates
- Private predicates
- Operators

Public predicates

help/0

Provides instructions on how to use the help tool.

Compilation flags:
    static

Mode and number of proofs:
    help - one

---

handbook/0

Provides access to the Handbook.

Compilation flags:
    static

Mode and number of proofs:
    handbook - one

---

apis/0

Provides access to the APIs documentation.

Compilation flags:
    static

Mode and number of proofs:
    apis - one

---

apis/1

Provides help on the given predicate or non-terminal.

Compilation flags:
    static

Template:
    apis(Indicator)
Mode and number of proofs:
    apis(+predicate_indicator) - one
    apis(+non_terminal_indicator) - one

---

tools/0

Provides access to the developer tools documentation.

Compilation flags:
    static

Mode and number of proofs:
    tools - one

---

tool/1

Provides help on the given developer tool.

Compilation flags:
    static

Template:
    tool(Tool)
Mode and number of proofs:
    tool(+atom) - zero_or_one

---

libraries/0

Provides access to the standard libraries documentation.

Compilation flags:
    static

Mode and number of proofs:
    libraries - one

---

library/1

Provides help on the given standard library.

Compilation flags:
    static

Template:
    library(Library)
Mode and number of proofs:
    library(+atom) - zero_or_one

---

entity/1

Provides help on the given built-in, tool, and library entity (object, protocol, or category).

Compilation flags:
    static

Template:
    entity(Entity)
Mode and number of proofs:
    entity(+entity_identifier) - zero_or_one

(/)/2

Provides help on the Name/Arity built-in control construct, directive, predicate, or method.

Compilation flags:
    static

Template:
    Name/Arity
Mode and number of proofs:
    +atom/ +integer - zero_or_one

(//)/2

Provides help on the Name//Arity built-in non-terminal.

Compilation flags:
    static

Template:
    Name//Arity
Mode and number of proofs:
    +atom// +integer - zero_or_one

---

man/1

Opens the man page of the given script. On POSIX systems, the page is open inline. On Windows system, the HTML version of the man page is open on the operating-system default browser.

Compilation flags:
    static

Template:
    man(Page)
Mode and number of proofs:
    man(+atom) - one

---

completion/2

Provides a completion pair, Completion-Page, for a given prefix.

Compilation flags:
    static

Template:
    completion(Prefix,Completion)
Mode and number of proofs:
    completion(+atom,-pair) - zero_or_more

---

completions/2

Provides a list of completions pairs, Completion-Page, for a given prefix.

Compilation flags:
    static

Template:
    completions(Prefix,Completions)
Mode and number of proofs:
    completions(+atom,-lists(pair)) - zero_or_more

---

built_in_directive/4

Provides access to the HTML documenting files describing built-in directives.

Compilation flags:
    static

Template:
    built_in_directive(Name,Arity,Directory,Basename)
Mode and number of proofs:

built_in_directive(?atom,?integer,-atom,-atom) - zero_or_more

---

built_in_predicate/4

Provides access to the HTML documenting files describing built-in predicates.

Compilation flags:
static

Template:
built_in_predicate(Name,Arity,Directory,Basename)
Mode and number of proofs:
built_in_predicate(?atom,?integer,-atom,-atom) - zero_or_more

---

built_in_method/4

Provides access to the HTML documenting files describing built-in methods.

Compilation flags:
static

Template:
built_in_method(Name,Arity,Directory,Basename)
Mode and number of proofs:
built_in_method(?atom,?integer,-atom,-atom) - zero_or_more

---

control_construct/4

Provides access to the HTML documenting files describing built-in control constructs.

Compilation flags:
static

Template:

control_construct(Name,Arity,Directory,Basename)
Mode and number of proofs:
control_construct(?atom,?integer,-atom,-atom) - zero_or_more

---

built_in_non_terminal/4

Provides access to the HTML documenting files describing built-in DCG non-terminals.

Compilation flags:
static

Template:
built_in_non_terminal(Name,Arity,Directory,Basename)
Mode and number of proofs:
built_in_non_terminal(?atom,?integer,-atom,-atom) - zero_or_more

---

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.92 hierarchical_clustering

object

1.92.1 hierarchical_clustering

Hierarchical clusterer for continuous datasets. Learns a full agglomerative_clusterer merge tree from a dataset object implementing the clustering_dataset_protocol protocol and then cuts the hierarchy to the requested number of clusters for prediction and export.

Availability:
logtalk_load(hierarchical_clustering(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public clusterer_common
Uses:
avltree
binary_heap_min
format
list
numberlist
pairs
type

Remarks:
(none)

Inherited public predicates:
check_clusterer/1 check_option/1 check_options/1 cluster/3 clusterer_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 print_clusterer/1 valid_clusterer/1 valid_option/1 valid_options/1

- Public predicates
  - cut/3
- Protected predicates
- Private predicates

- Operators

Public predicates

cut/3

Re-cuts a learned hierarchy to a new number of clusters without retraining.

Compilation flags:
    static

Template:
    cut(Clusterer,K,RecutClusterer)
Mode and number of proofs:
    cut(+compound,+integer,-compound) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

clusterer_protocol, clustering_dataset_protocol, agglomerative_clusterer

1.93 hierarchies

category

1.93.1 class_hierarchy

Class hierarchy predicates.

Availability:
logtalk_load(hierarchies(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2006-02-20

Compilation flags:
static

Implements:
public class_hierarchyp

Remarks:
(none)

Inherited public predicates:
ancestor/1 ancestors/1 class/1 classes/1 descendant/1 descendant_class/1 descendant_classes/1 descendant_instance/1 descendant_instances/1 descendants/1 instance/1 instances/1 leaf/1 leaf_class/1 leaf_classes/1 leaf_instance/1 leaf_instances/1 leaves/1 subclass/1 subclasses/1 superclass/1 superclasses/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.93.2 class_hierarchyp

Class hierarchy protocol.

Availability:
logtalk_load(hierarchies(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2000-07-24

Compilation flags:
static

Extends:
public hierarchyp

Remarks:
(none)

Inherited public predicates:
ancestor/1 ancestors/1 descendant/1 descendants/1 leaf/1 leaves/1

- Public predicates
  - class/1
  - classes/1
  - instance/1
  - instances/1
  - subclass/1
  - subclasses/1
  - superclass/1
  - superclasses/1
  - leaf_instance/1
  - leaf_instances/1
  - leaf_class/1
  - leaf_classes/1
  - descendant_instance/1
  - descendant_instances/1
  - descendant_class/1
  - descendant_classes/1
- Protected predicates
- Private predicates
- Operators

Public predicates

class/1

Returns, by backtracking, all object classes.

Compilation flags:
    static

Template:
    class(Class)
Mode and number of proofs:
    class(?object) - zero_or_more

classes/1

List of all object classes.

Compilation flags:
    static

Template:
    classes(Classes)
Mode and number of proofs:
    classes(-list) - one

---

instance/1

Returns, by backtracking, all class instances.

Compilation flags:
    static

Template:
    instance(Instance)
Mode and number of proofs:
    instance(?object) - zero_or_more

---

instances/1

List of all class instances.

Compilation flags:
    static

Template:
    instances(Instances)
Mode and number of proofs:
    instances(-list) - one

---

subclass/1

Returns, by backtracking, all class subclasses.

Compilation flags:
    static

Template:
    subclass(Subclass)
Mode and number of proofs:
    subclass(?object) - zero_or_more

---

subclasses/1

List of all class subclasses.

Compilation flags:
    static

Template:
    subclasses(Subclasses)
Mode and number of proofs:
    subclasses(-list) - one

---

superclass/1

Returns, by backtracking, all class superclasses.

Compilation flags:
    static

Template:
    superclass(Superclass)
Mode and number of proofs:
    superclass(?object) - zero_or_more

---

superclasses/1

List of all class superclasses.

Compilation flags:
    static

Template:
    superclasses(Superclasses)
Mode and number of proofs:
    superclasses(-list) - one

---

leaf_instance/1

Returns, by backtracking, all class leaf instances.

Compilation flags:
    static

Template:
    leaf_instance(Leaf)
Mode and number of proofs:
    leaf_instance(?object) - zero_or_more

---

leaf_instances/1

List of all class leaf instances.

Compilation flags:
    static

Template:
    leaf_instances(Leaves)
Mode and number of proofs:
    leaf_instances(-list) - one

---

leaf_class/1

Returns, by backtracking, all class leaf subclasses.

Compilation flags:
    static

Template:
    leaf_class(Leaf)
Mode and number of proofs:
    leaf_class(?object) - zero_or_more

---

leaf_classes/1

List of all class leaf leaf subclasses.

Compilation flags:
    static

Template:
    leaf_classes(Leaves)
Mode and number of proofs:
    leaf_classes(-list) - one

---

descendant_instance/1

Returns, by backtracking, all class descendant instances.

Compilation flags:
    static

Template:
    descendant_instance(Descendant)
Mode and number of proofs:
    descendant_instance(?object) - zero_or_more

---

descendant_instances/1

List of all class descendant instances.

Compilation flags:
    static

Template:
    descendant_instances(Descendants)
Mode and number of proofs:
    descendant_instances(-list) - one

---

descendant_class/1

Returns, by backtracking, all class descendant subclasses.

Compilation flags:
    static

Template:
    descendant_class(Descendant)
Mode and number of proofs:
    descendant_class(?object) - zero_or_more

---

descendant_classes/1

List of all class descendant subclasses.

Compilation flags:
    static

Template:
    descendant_classes(Descendants)
Mode and number of proofs:
    descendant_classes(-list) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

class_hierarchy

protocol

1.93.3 hierarchyp

Common hierarchy protocol for prototype and class hierarchies.

Availability:
logtalk_load(hierarchies(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2000-07-24

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - ancestor/1
  - ancestors/1
  - leaf/1
  - leaves/1
  - descendant/1
  - descendants/1
- Protected predicates
- Private predicates
- Operators

Public predicates

ancestor/1

Returns, by backtracking, all object ancestors.

Compilation flags:
    static

Template:
    ancestor(Ancestor)
Mode and number of proofs:
    ancestor(?object) - zero_or_more

---

ancestors/1

List of all object ancestors.

Compilation flags:
    static

Template:
    ancestors(Ancestors)
Mode and number of proofs:
    ancestors(-list) - one

---

leaf/1

Returns, by backtracking, all object leaves.

Compilation flags:
    static

Template:
    leaf(Leaf)
Mode and number of proofs:
    leaf(?object) - zero_or_more

---

leaves/1

List of all object leaves.

Compilation flags:
    static

Template:
    leaves(Leaves)
Mode and number of proofs:
    leaves(-list) - one

---

descendant/1

Returns, by backtracking, all object descendants.

Compilation flags:
    static

Template:
    descendant(Descendant)
Mode and number of proofs:

descendant(?object) - zero_or_more

---

descendants/1

List of all object descendants.

Compilation flags:
static

Template:
descendants(Descendants)
Mode and number of proofs:
descendants(-list) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

category

1.93.4 proto_hierarchy

Prototype hierarchy predicates.

Availability:
logtalk_load(hierarchies(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2006-02-20

Compilation flags:
static

Implements:
public proto_hierarchyp

Remarks:
(none)

Inherited public predicates:
ancestor/1 ancestors/1 descendant/1 descendants/1 extension/1 extensions/1 leaf/1 leaves/1 parent/1 parents/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.93.5 proto_hierarchyp

Prototype hierarchy protocol.

Availability:
logtalk_load(hierarchies(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2006-02-20

Compilation flags:
static

Extends:
public hierarchyp

Remarks:
(none)

Inherited public predicates:
ancestor/1 ancestors/1 descendant/1 descendants/1 leaf/1 leaves/1

- Public predicates
  - parent/1
  - parents/1
  - extension/1
  - extensions/1
- Protected predicates
- Private predicates
- Operators

Public predicates

parent/1

Returns, by backtracking, all object parents.

Compilation flags:
    static

Template:
    parent(Parent)
Mode and number of proofs:
    parent(?object) - zero_or_more

---

parents/1

List of all object parents.

Compilation flags:
    static

Template:
    parents(Parents)
Mode and number of proofs:
    parents(-list) - one

---

extension/1

Returns, by backtracking, all object direct descendants.

Compilation flags:
    static

Template:
    extension(Extension)
Mode and number of proofs:
    extension(?object) - zero_or_more

---

extensions/1

List of all object direct descendants.

Compilation flags:
    static

Template:
    extensions(Extensions)
Mode and number of proofs:
    extensions(-list) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

proto_hierarchy

1.94 hmac

object

1.94.1 hmac

HMAC (Keyed-Hash Message Authentication Code) implementation as specified in RFC 2104.

Availability:
logtalk_load(hmac(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-16

Compilation flags:
static, context_switching_calls

Implements:
public hmac_protocol
Uses:
hash_common_32
list

Remarks:
(none)

Inherited public predicates:
digest/4 digest/5 hex_digest/4 hex_digest/5

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

md5, sha1, sha256, sha3_224, sha3_256, sha3_384, sha3_512

protocol

1.94.2 hmac_protocol

Protocol for one-shot HMAC digest computation using hash objects implementing hash_digest_protocol.

Availability:
logtalk_load(hmac(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-16

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:

(none)

- Public predicates
  - digest/4
  - hex_digest/4
  - digest/5
  - hex_digest/5
- Protected predicates
- Private predicates
- Operators

Public predicates

digest/4

Computes the full HMAC digest for a hash object, key bytes, and message bytes.

Compilation flags:
static

Template:
digest(Hash,KeyBytes,MessageBytes,DigestBytes)
Mode and number of proofs:
digest(+object_identifier,+list(byte),+list(byte),--list(byte)) - one

---

hex_digest/4

Computes the full HMAC digest for a hash object, key bytes, and message bytes, returning a lowercase hexadecimal atom.

Compilation flags:
static

Template:
hex_digest(Hash,KeyBytes,MessageBytes,HexDigest)
Mode and number of proofs:

hex_digest(+object_identifier,+list(byte),+list(byte),--atom) - one

---

digest/5

Computes a truncated HMAC digest containing the requested number of leftmost bytes.

Compilation flags:
static

Template:
digest(Hash,KeyBytes,MessageBytes,Length,DigestBytes)
Mode and number of proofs:
digest(+object_identifier,+list(byte),+list(byte),+integer,--list(byte)) - one

---

hex_digest/5

Computes a truncated HMAC digest containing the requested number of leftmost bytes and returns it as a lowercase hexadecimal atom.

Compilation flags:
static

Template:
hex_digest(Hash,KeyBytes,MessageBytes,Length,HexDigest)
Mode and number of proofs:
hex_digest(+object_identifier,+list(byte),+list(byte),+integer,--atom) - one

---

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

hash_digest_protocol

1.95 hodge_rank

object

1.95.1 hodge_rank

HodgeRank pairwise measurement ranker. Learns one deterministic zero-sum score per item from a dataset object implementing the pairwise_measurement_dataset_protocol protocol by solving the weighted graph-Laplacian least-squares system induced by signed edge measurements and returns a self-describing ranker term with diagnostics and residuals.

Availability:
    logtalk_load(hodge_rank(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
    static, context_switching_calls

Imports:
    public ranking_dataset_common
    public score_ranker_model_common
    public pairwise_strength_ranker_common
Uses:
    avltree
    list
    numberlist

Remarks:

(none)

Inherited public predicates:

check_option/1 check_options/1 check_ranker/1 default_option/1 default_options/1
diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 grouped_dataset_groups/2
grouped_dataset_items/2 grouped_dataset_items/3 grouped_dataset_summary/2 learn/2
learn/3 option/2 option/3 pairwise_dataset_connected_components/2
pairwise_dataset_declared_items/2 pairwise_dataset_items/2 pairwise_dataset_matchups/2
pairwise_dataset_preferences/2 pairwise_dataset_summary/2 pairwise_dataset_win_totals/2
pairwise_measurement_dataset_connected_components/2
pairwise_measurement_dataset_declared_items/2 pairwise_measurement_dataset_items/2
pairwise_measurement_dataset_measurements/2 pairwise_measurement_dataset_summary/2
print_ranker/1 rank/3 ranker_options/2 scores/2
temporal_pairwise_dataset_connected_components/2
temporal_pairwise_dataset_declared_items/2 temporal_pairwise_dataset_games/2
temporal_pairwise_dataset_games/3 temporal_pairwise_dataset_items/2
temporal_pairwise_dataset_periods/2 temporal_pairwise_dataset_summary/2 valid_option/1
valid_options/1 valid_ranker/1 validate_grouped_dataset/1 validate_grouped_dataset/2
validate_pairwise_dataset/1 validate_pairwise_dataset/2
validate_pairwise_measurement_dataset/1 validate_pairwise_measurement_dataset/2
validate_temporal_pairwise_dataset/1 validate_temporal_pairwise_dataset/2

- Public predicates
  - residuals/2
- Protected predicates
- Private predicates
- Operators

Public predicates

residuals/2

Returns the learned weighted edge residuals as residual(Item1,Item2,Residual,Weight) terms preserving measurement enumeration order.

Compilation flags:
static

Template:
residuals(Ranker,Residuals)
Mode and number of proofs:
residuals(+compound,-list(compound)) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

pairwise_measurement_dataset_protocol, ranker_protocol, massey_ranker

1.96 hook_flows

object

1.96.1 hook_pipeline(Pipeline)

- Pipeline - List of hook objects.

Use a pipeline (represented using a list) of hook objects to expand terms and goals. The expansion results from a hook object are passed to the next hook object in the pipeline.

Availability:
logtalk_load(hook_flows(loader))

Author: Paulo Moura
Version: 2:0:0
Date: 2024-09-27

Compilation flags:
static, context_switching_calls

Implements:
public expanding

Remarks:

- Usage: Compile source files that should be expanded using the pipeline of hook objects using the compiler option hook(hook_pipeline(Pipeline)).

Inherited public predicates:
goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

hook_set(Set)

object

1.96.2 hook_set(Set)

- Set - Set (list) of hook objects.

Use a set (represented using a list) of hook objects to expand terms and goals. The hook objects are tried in sequence until one of them succeeds in expanding the current term (goal) into a different term (goal).

Availability:
logtalk_load(hook_flows(loader))

Author: Paulo Moura
Version: 2:0:0
Date: 2024-09-27

Compilation flags:
static, context_switching_calls

Implements:
public expanding

Remarks:

- Usage: Compile source files that should be expanded using the set of hook objects using the compiler option hook(hook_set(Set)).

Inherited public predicates:
goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

hook_pipeline(Pipeline)

1.97 hook_objects

object

1.97.1 backend_adapter_hook

This hook object applies the expansion rules defined in the Prolog backend adapter file.

Availability:
logtalk_load(hook_objects(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2020-02-17

Compilation flags:
static, context_switching_calls

Implements:
public expanding

Remarks:

(none)

Inherited public predicates:

goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

default_workflow_hook, identity_hook, grammar_rules_hook, prolog_module_hook(Module), object_wrapper_hook, write_to_stream_hook(Stream,Options), write_to_stream_hook(Stream), print_goal_hook, suppress_goal_hook

object

1.97.2 default_workflow_hook

Use this object as the default hook object to restore the default expansion pipeline semantics used by the compiler.

Availability:
logtalk_load(hook_objects(loader))

Author: Paulo Moura
Version: 1:0:1
Date: 2020-03-24

Compilation flags:
static, context_switching_calls

Implements:
public expanding

Remarks:
(none)

Inherited public predicates:
goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

backend_adapter_hook, identity_hook, grammar_rules_hook, prolog_module_hook(Module), object_wrapper_hook, write_to_stream_hook(Stream,Options), write_to_stream_hook(Stream), print_goal_hook, suppress_goal_hook

object

1.97.3 grammar_rules_hook

This hook object expands grammar rules into clauses.

Availability:
logtalk_load(hook_objects(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2020-02-14

Compilation flags:
static, context_switching_calls

Implements:
public expanding

Remarks:
(none)

Inherited public predicates:
goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

backend_adapter_hook, default_workflow_hook, identity_hook, prolog_module_hook(Module), object_wrapper_hook, write_to_stream_hook(Stream,Options), write_to_stream_hook(Stream), print_goal_hook, suppress_goal_hook

object

1.97.4 identity_hook

Use this object as a file specific hook object to prevent any (other) user-defined expansion rules to be applied when compiling the file.

Availability:
logtalk_load(hook_objects(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2020-02-15

Compilation flags:
static, context_switching_calls

Implements:
public expanding

Remarks:
(none)

Inherited public predicates:
goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

backend_adapter_hook, default_workflow_hook, grammar_rules_hook, prolog_module_hook(Module), object_wrapper_hook, write_to_stream_hook(Stream,Options), write_to_stream_hook(Stream), print_goal_hook, suppress_goal_hook

object

1.97.5 object_wrapper_hook

Use this object to wrap the contents of a plain Prolog file in an object named after the file. The wrapper sets the context_switching_calls flag to allow, enabling calling of the wrapped predicates using the <</2 control construct.

Availability:
logtalk_load(hook_objects(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2020-10-30

Compilation flags:
static, context_switching_calls

Implements:
public expanding
Uses:
os

Remarks:
(none)

Inherited public predicates:
goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

object_wrapper_hook(Protocol), object_wrapper_hook(Name,Relations), backend_adapter_hook, default_workflow_hook, grammar_rules_hook, prolog_module_hook(Module), write_to_stream_hook(Stream,Options), write_to_stream_hook(Stream), print_goal_hook, suppress_goal_hook

object

1.97.6 object_wrapper_hook(Protocol)

Use this object to wrap the contents of a plain Prolog file in an object named after the file that implements the given protocol.

Availability:
logtalk_load(hook_objects(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2021-11-24

Compilation flags:
static, context_switching_calls

Implements:
public expanding
Uses:
os

Remarks:
(none)

Inherited public predicates:
goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

object_wrapper_hook, object_wrapper_hook(Name,Relations), backend_adapter_hook, default_workflow_hook, grammar_rules_hook, prolog_module_hook(Module), write_to_stream_hook(Stream,Options), write_to_stream_hook(Stream), print_goal_hook, suppress_goal_hook

object

1.97.7 object_wrapper_hook(Name,Relations)

Use this object to wrap the contents of a plain Prolog file in an object with the given name and object entity relations (a list).

Availability:
logtalk_load(hook_objects(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2022-02-03

Compilation flags:
static, context_switching_calls

Implements:
public expanding

Remarks:
(none)

Inherited public predicates:
goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

object_wrapper_hook, object_wrapper_hook(Protocol), backend_adapter_hook, default_workflow_hook, grammar_rules_hook, prolog_module_hook(Module), write_to_stream_hook(Stream,Options), write_to_stream_hook(Stream), print_goal_hook, suppress_goal_hook

object

1.97.8 print_goal_hook

Use this object to easily print entity predicate goals before, after, or before and after calling them.

Availability:
logtalk_load(hook_objects(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2020-03-14

Compilation flags:
static, context_switching_calls

Implements:
public expanding

Remarks:

- Usage: Mark a goal to be printed by prefixing it with an operator. Printing uses a comment message.
- To print goal before calling it: - Goal.
- To print goal after calling it: + Goal.

- To print goal before and after calling it: * Goal.
- Operators: This hook object uses the standard - and + prefix operators and also defines a global * prefix operator with the same type and priority.

Inherited public predicates:

goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

backend_adapter_hook, default_workflow_hook, grammar_rules_hook, identity_hook, prolog_module_hook(Module), object_wrapper_hook, write_to_stream_hook(Stream,Options), write_to_stream_hook(Stream), suppress_goal_hook

object

1.97.9 prolog_module_hook(Module)

This hook object applies the expansion rules defined in a Prolog module (e.g., user).

Availability:
logtalk_load(hook_objects(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2020-02-17

Compilation flags:
static, context_switching_calls

Implements:
public expanding

Remarks:
(none)

Inherited public predicates:
goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

backend_adapter_hook, default_workflow_hook, identity_hook, grammar_rules_hook, object_wrapper_hook, write_to_stream_hook(Stream,Options), write_to_stream_hook(Stream), print_goal_hook, suppress_goal_hook

object

1.97.10 suppress_goal_hook

Use this object to easily suppress a goal in a clause body.

Availability:
logtalk_load(hook_objects(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2020-05-04

Compilation flags:
static, context_switching_calls

Implements:
public expanding

Remarks:

- Usage: Mark a goal to be suppressed by prefixing it with the -- operator.
- Operators: This hook object uses the -- prefix operator declared by Logtalk for use in mode/2 directives.

Inherited public predicates:

goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

backend_adapter_hook, default_workflow_hook, grammar_rules_hook, identity_hook, prolog_module_hook(Module), object_wrapper_hook, write_to_stream_hook(Stream,Options), write_to_stream_hook(Stream), print_goal_hook

object

1.97.11 write_to_file_hook(File)

This hook object writes term-expansion results to a file in canonical format. The terms are terminated by a period and a new line.

Availability:
logtalk_load(hook_objects(loader))

Author: Paulo Moura
Version: 1:0:0

Date: 2022-07-06

Compilation flags:
static, context_switching_calls

Extends:
public write_to_file_hook(File,[quoted(true),ignore_ops(true)])

Remarks:
(none)

Inherited public predicates:
goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

backend_adapter_hook, default_workflow_hook, identity_hook, grammar_rules_hook, prolog_module_hook(Module), object_wrapper_hook, write_to_file_hook(File,Options),

write_to_stream_hook(Stream,Options), write_to_stream_hook(Stream), print_goal_hook, suppress_goal_hook

object

1.97.12 write_to_file_hook(File,Options)

This hook object writes term-expansion results to a file using a list of write_term/3 options. The terms are terminated by a period and a new line.

Availability:
logtalk_load(hook_objects(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2022-07-06

Compilation flags:
static, context_switching_calls

Implements:
public expanding

Remarks:
(none)

Inherited public predicates:
goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

backend_adapter_hook, default_workflow_hook, identity_hook, grammar_rules_hook, prolog_module_hook(Module), object_wrapper_hook, write_to_file_hook(File), write_to_stream_hook(Stream,Options), write_to_stream_hook(Stream), print_goal_hook, suppress_goal_hook

object

1.97.13 write_to_stream_hook(Stream)

This hook object writes term-expansion results to a stream in canonical format. The terms are terminated by a period and a new line.

Availability:
logtalk_load(hook_objects(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2020-02-16

Compilation flags:
static, context_switching_calls

Extends:
public write_to_stream_hook(Stream,[quoted(true),ignore_ops(true)])

Remarks:

(none)

Inherited public predicates:

goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

backend_adapter_hook, default_workflow_hook, identity_hook, grammar_rules_hook, prolog_module_hook(Module), object_wrapper_hook, write_to_stream_hook(Stream,Options), write_to_file_hook(File,Options), write_to_file_hook(File), print_goal_hook, suppress_goal_hook

object

1.97.14 write_to_stream_hook(Stream,Options)

This hook object writes term-expansion results to a stream using a list of write_term/3 options. The terms are terminated by a period and a new line.

Availability:
logtalk_load(hook_objects(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2020-02-16

Compilation flags:
static, context_switching_calls

Implements:
public expanding

Remarks:
(none)

Inherited public predicates:
goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

backend_adapter_hook, default_workflow_hook, identity_hook, grammar_rules_hook, prolog_module_hook(Module), object_wrapper_hook, write_to_stream_hook(Stream), write_to_file_hook(File,Options), write_to_file_hook(File), print_goal_hook, suppress_goal_hook

1.98 html

category

1.98.1 html

HTML generation.

Availability:
logtalk_load(html(loader))

Author: Paul Brown and Paulo Moura
Version: 0:3:0
Date: 2021-03-30

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - generate/2
  - void_element/1
  - normal_element/2
- Protected predicates
- Private predicates
  - doctype/1
- Operators

Public predicates

generate/2

Generates HTML content using the representation specified in the first argument (stream(Stream) or file(Path)) for the term in the second argument.

Compilation flags:
static

Template:
generate(Sink,Term)
Mode and number of proofs:
generate(+compound,++term) - one_or_error

---

void_element/1

Enumerates, by backtracking, all void elements.

Compilation flags:
static

Template:
void_element(Element)

Mode and number of proofs:
    void_element(?atom) - zero_or_more

---

normal_element/2

Enumerates, by backtracking, all normal elements. The value of the Display argument is either inline or block.

Compilation flags:
    static

Template:
    normal_element(Element,Display)
Mode and number of proofs:
    normal_element(?atom,?atom) - zero_or_more

Protected predicates

(none)

Private predicates

doctype/1

Doctype text.

Compilation flags:
    static

Template:
    doctype(DocType)
Mode and number of proofs:
    doctype(?atom) - one

Operators

(none)

object

1.98.2 html5

HTML content generation using the HTML 5 doctype.

Availability:
logtalk_load(html(loader))

Author: Paul Brown and Paulo Moura
Version: 1:0:0
Date: 2021-03-29

Compilation flags:
static, context_switching_calls

Imports:
public html

Remarks:
(none)

Inherited public predicates:
generate/2 normal_element/2 void_element/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.98.3 xhtml11

XHTML content generation using the XHTML 1.1 doctype.

Availability:
logtalk_load(html(loader))

Author: Paul Brown and Paulo Moura
Version: 1:0:0
Date: 2021-03-29

Compilation flags:
static, context_switching_calls

Imports:
public html

Remarks:
(none)

Inherited public predicates:
generate/2 normal_element/2 void_element/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.99 ica_projection

object

1.99.1 ica_projection

Independent Component Analysis reducer for continuous datasets using a portable FastICA-style solver with symmetric eigendecomposition whitening.

Availability:
logtalk_load(ica_projection(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:

public dimension_reducer_common

Uses:

format
linear_algebra
list
numberlist
pairs
type

Remarks:

(none)

Inherited public predicates:

check_dimension_reducer/1 check_option/1 check_options/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 dimension_reducer_options/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 print_dimension_reducer/1 transform/3 valid_dimension_reducer/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

nmf_projection, pca_projection, random_projection, truncated_svd_projection

1.100 ids

object

1.100.1 ids

Generator of random identifiers represented as atoms with 160 bits (20 bytes) of randomness.

Availability:
logtalk_load(ids(loader))

Author: Paulo Moura
Version: 1:2:0
Date: 2026-02-26

Compilation flags:
static, context_switching_calls

Extends:
public ids(atom,20)

Remarks:
(none)

Inherited public predicates:
generate/1

- Public predicates
- Protected predicates
- Private predicates

- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

ids(Representation,Bytes), cuid2, ksuid, nanoid, snowflakeid, uuid, ulid

object

1.100.2 ids(Representation,Bytes)

- Representation - Text representation for the identifier. Possible values are atom, chars, and codes.
- Bytes - Number of bytes of randomness.

Generator of random identifiers.

Availability:
logtalk_load(ids(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-02-26

Compilation flags:
static, context_switching_calls

Uses:

base64
fast_random
list
os

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - generate/1
- Protected predicates
- Private predicates
- Operators

Public predicates

generate/1

Generate a random identifier.

Compilation flags:
static

Template:
generate(Identifier)
Mode and number of proofs:
generate(--textids) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

ids, nanoid(Representation,Size,Alphabet), cuid2(Representation,Size,Alphabet), ksuid(Representation,Alphabet), uuid(Representation), snowflakeid(Representation,EpochMilliseconds,TimeUnitMillisecon ulid(Representation)

1.101 ieee_754

object

1.101.1 ieee_754(Precision,ByteOrder,NaNRepresentation)

- Precision - Floating-point format precision. Supported values are half, single, and double.
- ByteOrder - Byte order used when parsing and generating byte sequences. Supported values are big and little.
- NaNRepresentation - NaN decoding and generation policy. Supported values are canonical and payloads.

IEEE 754 floating-point encoder and decoder.

Availability:
logtalk_load(ieee_754(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-11

Compilation flags:
static, context_switching_calls

Implements:

public ieee_754_protocol
Imports:
public ieee_754_common(Precision,ByteOrder)
Uses:
type

Remarks:
(none)

Inherited public predicates:
byte_count/1 exactly_representable/1 generate/2 generate/3 nan_representation/1 order/1 parse/2 precision/1 valid/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.101.2 ieee_754_common(Precision,ByteOrder)

Shared IEEE 754 exact bit and byte handling predicates for the high-level codec and low-level field inspection objects.

Availability:
logtalk_load(ieee_754(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-11

Compilation flags:
static

Uses:
list
type

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
  - source_bits/2
  - bits_fields/5
  - bits_finite_binary_rational/4
  - bits_nan_payload/2
  - bits_nan_kind/2
  - canonical_nan_bits/1
  - nan_bits/1
  - valid_bits/1
  - canonical_order_bytes/3
  - order_bytes/3

  - bytes_tail/3
  - bytes_to_unsigned_integer/2
  - bytes_to_unsigned_integer/3
  - integer_to_bytes/3
  - zero_bits/2
  - zero_from_sign/2
  - apply_float_sign/3
  - precision_spec/5
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

source_bits/2

Validates and normalizes a source term into canonical-order unsigned integer bits for the selected precision and byte order.

Compilation flags:
    static

Template:
    source_bits(Source,Bits)
Mode and number of proofs:
    source_bits(++compound,--integer) - one_or_error

bits_fields/5

Extracts exact IEEE 754 sign, exponent bits, mantissa bits, and a class atom from a validated bit pattern.

Compilation flags:
    static

Template:
    bits_fields(Bits,Sign,ExponentBits,MantissaBits,Class)
Mode and number of proofs:
    bits_fields(+integer,-integer,-integer,-integer,-atom) - one

---

bits_finite_binary_rational/4

Extracts an exact finite IEEE 754 value from a validated bit pattern as (-1)^Sign * Significand * 2^Exponent.

Compilation flags:
    static

Template:
    bits_finite_binary_rational(Bits,Sign,Significand,Exponent)
Mode and number of proofs:
    bits_finite_binary_rational(+integer,-integer,-integer,-integer) - zero_or_one

---

bits_nan_payload/2

Extracts the raw NaN mantissa payload bits from a validated bit pattern.

Compilation flags:
    static

Template:
    bits_nan_payload(Bits,PayloadBits)
Mode and number of proofs:
    bits_nan_payload(+integer,-integer) - zero_or_one

---

bits_nan_kind/2

Classifies a validated NaN bit pattern as quiet or signaling using the quiet/signaling discriminator bit for the selected precision.

Compilation flags:
    static

Template:
    bits_nan_kind(Bits,Kind)
Mode and number of proofs:
    bits_nan_kind(+integer,-atom) - zero_or_one

---

canonical_nan_bits/1

Returns the canonical quiet NaN bit pattern for the selected precision.

Compilation flags:
    static

Template:
    canonical_nan_bits(Bits)
Mode and number of proofs:
    canonical_nan_bits(-integer) - one

---

nan_bits/1

True when the validated bit pattern encodes a NaN value for the selected precision.

Compilation flags:
    static

Template:
    nan_bits(Bits)
Mode and number of proofs:
    nan_bits(+integer) - zero_or_one

---

valid_bits/1

True when the integer fits in the selected precision bit width.

Compilation flags:
    static

Template:
    valid_bits(Bits)
Mode and number of proofs:
    valid_bits(+integer) - zero_or_one

---

canonical_order_bytes/3

Converts byte lists from the selected byte order to canonical big-endian order.

Compilation flags:
    static

Template:
    canonical_order_bytes(ByteOrder,Bytes,CanonicalBytes)
Mode and number of proofs:
    canonical_order_bytes(+atom,+list(integer),-list(integer)) - one

---

order_bytes/3

Converts canonical big-endian bytes to the selected byte order.

Compilation flags:
    static

Template:
    order_bytes(ByteOrder,CanonicalBytes,Bytes)
Mode and number of proofs:

order_bytes(+atom,+list(integer),-list(integer)) - one

---

bytes_tail/3

Builds a difference list from a byte list.

Compilation flags:
static

Template:
bytes_tail(Bytes,DifferenceList,Tail)
Mode and number of proofs:
bytes_tail(+list(integer),-list(integer),?list(integer)) - one

---

bytes_to_unsigned_integer/2

Converts canonical-order bytes into an unsigned integer.

Compilation flags:
static

Template:
bytes_to_unsigned_integer(Bytes,Integer)
Mode and number of proofs:
bytes_to_unsigned_integer(+list(integer),-integer) - one

---

bytes_to_unsigned_integer/3

Worker predicate for converting canonical-order bytes into an unsigned integer.

Compilation flags:
static

Template:

bytes_to_unsigned_integer(Bytes,Integer0,Integer)
Mode and number of proofs:
bytes_to_unsigned_integer(+list(integer),+integer,-integer) - one

---

integer_to_bytes/3

Converts an unsigned integer into a canonical-order byte list with the requested length.

Compilation flags:
static

Template:
integer_to_bytes(Count,Integer,Bytes)
Mode and number of proofs:
integer_to_bytes(+integer,+integer,-list(integer)) - one

---

zero_bits/2

Returns the zero encoding bits with the requested sign bit.

Compilation flags:
static

Template:
zero_bits(Sign,Bits)
Mode and number of proofs:
zero_bits(+integer,-integer) - one

---

zero_from_sign/2

Constructs a positive or negative floating-point zero from a sign bit.

Compilation flags:
    static

Template:
    zero_from_sign(Sign,Zero)
Mode and number of proofs:
    zero_from_sign(+integer,-float) - one

---

apply_float_sign/3

Applies a sign bit to a positive floating-point magnitude.

Compilation flags:
    static

Template:
    apply_float_sign(Sign,Magnitude,Value)
Mode and number of proofs:
    apply_float_sign(+integer,+float,-float) - one

---

precision_spec/5

Returns the exponent width, mantissa width, exponent bias, and byte count for a supported IEEE 754 precision.

Compilation flags:
    static

Template:
    precision_spec(Precision,ExponentWidth,MantissaWidth,Bias,ByteCount)
Mode and number of proofs:
    precision_spec(+atom,-integer,-integer,-integer,-integer) - one

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.101.3 ieee_754_fields(Precision,ByteOrder)

- Precision - Floating-point format precision. Supported values are half, single, and double.
- ByteOrder - Byte order used when reading byte sequences. Supported values are big and little.

IEEE 754 exact bit-field classifier and decomposition object.

Availability:
logtalk_load(ieee_754(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-09

Compilation flags:
static, context_switching_calls

Implements:
public ieee_754_fields_protocol
Imports:
public ieee_754_common(Precision,ByteOrder)

Remarks:
(none)

Inherited public predicates:
byte_count/1 classify/2 fields/5 finite_binary_rational/4 nan_kind/2 nan_payload/2 order/1 precision/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.101.4 ieee_754_fields_protocol

IEEE 754 exact bit-field support library protocol. Intended to be implemented by parameterized objects such as ieee_754_fields(Precision, ByteOrder).

Availability:
logtalk_load(ieee_754(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-09

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - classify/2
  - fields/5
  - finite_binary_rational/4
  - nan_payload/2
  - nan_kind/2
  - precision/1
  - order/1
  - byte_count/1
- Protected predicates
- Private predicates
- Operators

Public predicates

classify/2

Classifies an IEEE 754 encoding source term as zero, subnormal, normal, infinity, or not_a_number for the selected precision and byte order. Supported source terms are bytes(Bytes) and bits(Bits).

Compilation flags:
static

Template:
classify(Source,Class)
Mode and number of proofs:
classify(++compound,--atom) - one_or_error

Exceptions:
Source is a variable:
instantiation_error
Source is neither a variable nor a valid source term:

domain_error(ieee_754_source,Source)
Source does not contain a valid encoding for the selected object:
domain_error(ieee_754_encoding,Source)

---

fields/5

Extracts the exact IEEE 754 sign bit, exponent bits, mantissa bits, and classification from a source term.

Compilation flags:
static

Template:
fields(Source,Sign,ExponentBits,MantissaBits,Class)
Mode and number of proofs:
fields(++compound,--integer,--integer,--integer,--atom) - one_or_error

Exceptions:
Source is a variable:
instantiation_error
Source is neither a variable nor a valid source term:
domain_error(ieee_754_source,Source)
Source does not contain a valid encoding for the selected object:
domain_error(ieee_754_encoding,Source)

---

finite_binary_rational/4

Extracts the exact finite IEEE 754 value as (-1)^Sign * Significand * 2^Exponent from a source term. Zero encodings return a zero significand and exponent 0.

Compilation flags:
static

Template:
finite_binary_rational(Source,Sign,Significand,Exponent)
Mode and number of proofs:
finite_binary_rational(++compound,--integer,--integer,--integer) - one_or_error

Exceptions:
Source is a variable:
instantiation_error
Source is neither a variable nor a valid source term:
domain_error(ieee_754_source,Source)
Source does not contain a valid encoding for the selected object:
domain_error(ieee_754_encoding,Source)
Source does not encode a finite IEEE 754 value:
domain_error(ieee_754_finite_encoding,Source)

---

nan_payload/2

Extracts the raw NaN mantissa payload bits from a source term.

Compilation flags:
static

Template:
nan_payload(Source,PayloadBits)
Mode and number of proofs:
nan_payload(++compound,--integer) - one_or_error

Exceptions:
Source is a variable:
instantiation_error
Source is neither a variable nor a valid source term:
domain_error(ieee_754_source,Source)
Source does not contain a valid encoding for the selected object:
domain_error(ieee_754_encoding,Source)
Source does not encode a NaN value:
domain_error(ieee_754_nan_encoding,Source)

---

nan_kind/2

Classifies a NaN source term as quiet or signaling using the exact IEEE 754 quiet/signaling discriminator bit for the selected precision.

Compilation flags:
    static

Template:
    nan_kind(Source,Kind)
Mode and number of proofs:
    nan_kind(++compound,--atom) - one_or_error

Exceptions:
    Source is a variable:
        instantiation_error
    Source is neither a variable nor a valid source term:
        domain_error(ieee_754_source,Source)
    Source does not contain a valid encoding for the selected object:
        domain_error(ieee_754_encoding,Source)
    Source does not encode a NaN value:
        domain_error(ieee_754_nan_encoding,Source)

---

precision/1

Returns the selected IEEE 754 precision. Expected values are half, single, or double.

Compilation flags:
    static

Template:
    precision(Precision)
Mode and number of proofs:
    precision(-atom) - one

---

order/1

Returns the selected byte order. Expected values are big or little.

Compilation flags:
    static

Template:
    order(ByteOrder)
Mode and number of proofs:
    order(-atom) - one

---

byte_count/1

Returns the number of bytes used by the selected precision.

Compilation flags:
    static

Template:
    byte_count(ByteCount)
Mode and number of proofs:
    byte_count(-integer) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

protocol

1.101.5 ieee_754_protocol

IEEE 754 floating-point support library protocol. Intended to be implemented by parameterized objects such as ieee_754(Precision, ByteOrder, NaNRepresentation).

Availability:
logtalk_load(ieee_754(loader))

Author: Paulo Moura
Version: 0:1:0
Date: 2026-05-09

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - parse/2
  - generate/2
  - generate/3
  - valid/1
  - exactly_representable/1
  - precision/1
  - order/1
  - nan_representation/1

    - byte_count/1
- Protected predicates
- Private predicates
- Operators

Public predicates

parse/2

Parses an IEEE 754 value from a source term for the selected precision, byte order, and NaN representation policy. Supported source terms are bytes(Bytes) and bits(Bits).

Compilation flags:
    static

Template:
    parse(Source,Value)
Mode and number of proofs:
    parse(++compound,--term) - one_or_error

Exceptions:
    Source is a variable:
        instantiation_error
    Source is neither a variable nor a valid source term:
        domain_error(ieee_754_source,Source)
    Source does not contain a valid encoding for the selected object:
        domain_error(ieee_754_encoding,Source)

---

generate/2

Generates an IEEE 754 encoding for a value term using the selected precision and byte order. Supported sink terms are bytes(Bytes) and bits(Bits). Finite values are encoded using the selected precision rounding semantics.

Compilation flags:
    static

Template:

generate(Sink,Value)
Mode and number of proofs:
generate(+compound,++term) - one_or_error

Exceptions:
Sink is a variable:
instantiation_error
Sink is neither a variable nor a valid sink term:
domain_error(ieee_754_sink,Sink)
Value is neither a valid finite float nor a supported IEEE 754 special value term:
domain_error(ieee_754_value,Value)
Value cannot be represented in the selected precision:
domain_error(ieee_754_representation,Value)

---

generate/3

Generates an IEEE 754 encoding as a list of bytes with an open tail. Intended for use by libraries that build larger binary payloads using difference lists.

Compilation flags:
static

Template:
generate(Value,Bytes,Tail)
Mode and number of proofs:
generate(+term,-list(byte),--variable) - one_or_error

Exceptions:
Value is neither a valid finite float nor a supported IEEE 754 special value term:
domain_error(ieee_754_value,Value)
Value cannot be represented in the selected precision:
domain_error(ieee_754_representation,Value)

---

valid/1

True iff the argument is a valid value term for the selected object. Valid terms are finite floats plus the special values supported by the selected NaN representation policy.

Compilation flags:
    static

Template:
    valid(Value)
Mode and number of proofs:
    valid(@term) - zero_or_one

---

exactly_representable/1

True iff the argument can be encoded in the selected precision and decoded again without loss of value information under the selected NaN representation policy. Intended for callers such as MessagePack that choose the smallest exact binary format.

Compilation flags:
    static

Template:
    exactly_representable(Value)
Mode and number of proofs:
    exactly_representable(@term) - zero_or_one

---

precision/1

Returns the selected IEEE 754 precision. Expected values are half, single, or double.

Compilation flags:
    static

Template:
    precision(Precision)
Mode and number of proofs:

precision(-atom) - one

---

order/1

Returns the selected byte order. Expected values are big or little.

Compilation flags:
static

Template:
order(ByteOrder)
Mode and number of proofs:
order(-atom) - one

---

nan_representation/1

Returns the selected NaN representation policy. Expected values are canonical and payloads.

Compilation flags:
static

Template:
nan_representation(NaNRepresentation)
Mode and number of proofs:
nan_representation(-atom) - one

---

byte_count/1

Returns the number of bytes used by the selected precision.

Compilation flags:
static

Template:

byte_count(ByteCount)
Mode and number of proofs:
byte_count(-integer) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.102 intervals

object

1.102.1 interval

Basic temporal interval relations. An interval is represented by a compound term, i/2, with two ground arguments, the start and end points.

Availability:
logtalk_load(intervals(loader))

Author: Paulo Moura
Version: 1:3:0
Date: 2026-05-13

Compilation flags:
static, context_switching_calls

Implements:
public interval_protocol
Aliases:
interval_protocol before/2 as b/2
interval_protocol after/2 as bi/2

interval_protocol meets/2 as m/2
interval_protocol met_by/2 as mi/2
interval_protocol overlaps/2 as o/2
interval_protocol overlapped_by/2 as oi/2
interval_protocol starts/2 as s/2
interval_protocol started_by/2 as si/2
interval_protocol during/2 as d/2
interval_protocol contains/2 as di/2
interval_protocol finishes/2 as f/2
interval_protocol finished_by/2 as fi/2
interval_protocol equal/2 as eq/2

Remarks:
(none)

Inherited public predicates:
after/2 before/2 contains/2 during/2 equal/2 finished_by/2 finishes/2 meets/2 met_by/2 new/3 overlapped_by/2 overlaps/2 relation/3 started_by/2 starts/2 valid/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.102.2 interval_algebra

Allen interval algebra over the 13 base relation atoms. Composition results are returned as canonical ordered duplicate-free lists.

Availability:
logtalk_load(intervals(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-13

Compilation flags:
static, context_switching_calls

Implements:
public interval_algebra_protocol

Remarks:
(none)

Inherited public predicates:
compose/3 converse/2 relation/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.102.3 interval_algebra_protocol

Allen interval algebra protocol over the 13 base relation atoms.

Availability:
logtalk_load(intervals(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-13

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - relation/1
  - converse/2
  - compose/3
- Protected predicates
- Private predicates
- Operators

Public predicates

relation/1

Enumerates valid Allen base relation atoms in canonical order.

Compilation flags:
    static

Template:
    relation(Relation)
Mode and number of proofs:
    relation(?atom) - zero_or_more

---

converse/2

Relates an Allen base relation atom with its converse relation atom.

Compilation flags:
    static

Template:
    converse(Relation,Converse)
Mode and number of proofs:
    converse(?atom,?atom) - zero_or_more

---

compose/3

Returns the canonical ordered list of Allen base relation atoms resulting from composing two Allen base relation atoms.

Compilation flags:
static

Template:
compose(Relation1,Relation2,Relations)
Mode and number of proofs:
compose(+atom,+atom,-list(atom)) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

interval_protocol, interval, interval_algebra, interval_relation_set_protocol, interval_relation_set

object

1.102.4 interval_constraint_network

Allen interval constraint-network operations built on canonical relation sets, intended for small-to-medium symbolic networks.

Availability:
logtalk_load(intervals(loader))

Author: Paulo Moura
Version: 1:0:0

Date: 2026-05-13

Compilation flags:
static, context_switching_calls

Implements:
public interval_constraint_network_protocol
Uses:
interval_relation_set
list

Remarks:
(none)

Inherited public predicates:
consistent/1 constraints/2 contradiction/2 contradiction_explanations/2 entailment_explanations/5 entails/4 entails/5 equivalent/2 excluded/4 network/1 new/2 nodes/2 path_consistency/2 possible/4 propagate/2 propagate/3 propagation_triple/2 propagation_triples/2 refine/5 refine_propagate/3 refine_propagate/4 refine_propagate/5 refine_propagate/6 relation/4 subsumes/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.102.5 interval_constraint_network_protocol

Allen interval constraint-network protocol built on canonical relation sets, intended for small-to-medium symbolic networks.

Availability:
logtalk_load(intervals(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-13

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - network/1
  - new/2
  - nodes/2
  - relation/4
  - entails/4

  - entails/5
  - possible/4
  - excluded/4
  - contradiction/2
  - entailment_explanations/5
  - contradiction_explanations/2
  - refine/5
  - consistent/1
  - propagate/2
  - propagate/3
  - refine_propagate/5
  - refine_propagate/6
  - refine_propagate/3
  - refine_propagate/4
  - propagation_triple/2
  - propagation_triples/2
  - path_consistency/2
  - constraints/2
  - subsumes/2
  - equivalent/2
- Protected predicates
- Private predicates
- Operators

Public predicates

network/1

True if Network is a valid interval constraint network represented as a network(Nodes, Constraints) term.

Compilation flags:
    static

Template:
    network(Network)
Mode and number of proofs:
    network(?compound) - zero_or_more

---

new/2

Constructs a new network for a list of distinct nodes. Distinct node pairs are initialized with the universal relation set.

Compilation flags:
    static

Template:
    new(Nodes,Network)
Mode and number of proofs:
    new(+list,-compound) - zero_or_one

---

nodes/2

Returns the list of nodes in a network.

Compilation flags:
    static

Template:
    nodes(Network,Nodes)
Mode and number of proofs:
    nodes(+compound,-list) - zero_or_one

---

relation/4

Returns the current relation set between two nodes in a network. A node is always equal to itself.

Compilation flags:
    static

Template:
    relation(Network,Node1,Node2,RelationSet)

Mode and number of proofs:
relation(+compound,@term,@term,-list(atom)) - zero_or_one

---

entails/4

True if the current relation set between two nodes is a subset of a queried relation set.

Compilation flags:
static

Template:
entails(Network,Node1,Node2,RelationSet)
Mode and number of proofs:
entails(+compound,@term,@term,+list(atom)) - zero_or_one

---

entails/5

True if the current relation set between two nodes is a subset of a queried relation set and returns an explanation term describing either a direct constraint or a supporting propagation path.

Compilation flags:
static

Template:
entails(Network,Node1,Node2,RelationSet,Explanation)
Mode and number of proofs:
entails(+compound,@term,@term,+list(atom),-compound) - zero_or_one

---

possible/4

True if the current relation set between two nodes intersects a queried relation set.

Compilation flags:
    static

Template:
    possible(Network,Node1,Node2,RelationSet)
Mode and number of proofs:
    possible(+compound,@term,@term,+list(atom)) - zero_or_one

---

excluded/4

True if the current relation set between two nodes is disjoint from a queried relation set.

Compilation flags:
    static

Template:
    excluded(Network,Node1,Node2,RelationSet)
Mode and number of proofs:
    excluded(+compound,@term,@term,+list(atom)) - zero_or_one

---

contradiction/2

Returns a simple contradiction explanation for an inconsistent network as contradiction(Node1, Node2, Cause).

Compilation flags:
    static

Template:
    contradiction(Network,Explanation)
Mode and number of proofs:
    contradiction(+compound,-compound) - zero_or_more

---

entailment_explanations/5

Returns the list of immediate explanations supporting an entailed relation query, using identity, direct, and propagated explanation terms.

Compilation flags:
    static

Template:
    entailment_explanations(Network,Node1,Node2,RelationSet,Explanations)
Mode and number of proofs:
    entailment_explanations(+compound,@term,@term,+list(atom),-list(compound)) - zero_or_one

---

contradiction_explanations/2

Returns the list of contradiction explanations for all empty explicit pair constraints in a network.

Compilation flags:
    static

Template:
    contradiction_explanations(Network,Explanations)
Mode and number of proofs:
    contradiction_explanations(+compound,-list(compound)) - zero_or_one

---

refine/5

Refines the relation set between two distinct nodes by intersecting it with a new relation set.

Compilation flags:
    static

Template:
    refine(Network,Node1,Node2,RelationSet,RefinedNetwork)

Mode and number of proofs:
    refine(+compound,@term,@term,+list(atom),-compound) - zero_or_one

---

consistent/1

True if all explicit pair constraints in a network are non-empty.

Compilation flags:
    static

Template:
    consistent(Network)
Mode and number of proofs:
    consistent(+compound) - zero_or_one

---

propagate/2

Computes a path-consistency closure and succeeds only if the resulting network is consistent. Intended for small-to-medium symbolic networks; not a large-scale temporal CSP solver.

Compilation flags:
    static

Template:
    propagate(Network,Closure)
Mode and number of proofs:
    propagate(+compound,-compound) - zero_or_one

---

propagate/3

Computes a path-consistency closure, returns the list of propagated changes, and succeeds only if the resulting network is consistent. Intended for small-to-medium symbolic networks; not a large-scale temporal CSP solver.

Compilation flags:
    static

Template:
    propagate(Network,Closure,Changes)
Mode and number of proofs:
    propagate(+compound,-compound,-list(compound)) - zero_or_one

---

refine_propagate/5

Refines the relation set between two distinct nodes and then propagates the consequences, succeeding only if the resulting network is consistent. Intended for small-to-medium symbolic networks; not a large-scale temporal CSP solver.

Compilation flags:
    static

Template:
    refine_propagate(Network,Node1,Node2,RelationSet,Closure)
Mode and number of proofs:
    refine_propagate(+compound,@term,@term,+list(atom),-compound) - zero_or_one

---

refine_propagate/6

Refines the relation set between two distinct nodes, propagates the consequences incrementally, and returns the list of direct and propagated changes, succeeding only if the resulting network is consistent. Intended for small-to-medium symbolic networks; not a large-scale temporal CSP solver.

Compilation flags:
    static

Template:

refine_propagate(Network,Node1,Node2,RelationSet,Closure,Changes)
Mode and number of proofs:
refine_propagate(+compound,@term,@term,+list(atom),-compound,-list(compound)) - zero_or_one

---

refine_propagate/3

Applies a batch of relation-set refinements represented as constraint(Node1, Node2, RelationSet) terms and then propagates the combined consequences, succeeding only if the resulting network is consistent. Intended for small-to-medium symbolic networks; not a large-scale temporal CSP solver.

Compilation flags:
static

Template:
refine_propagate(Network,Constraints,Closure)
Mode and number of proofs:
refine_propagate(+compound,+list(compound),-compound) - zero_or_one

---

refine_propagate/4

Applies a batch of relation-set refinements, propagates the combined consequences incrementally, and returns the list of direct and propagated changes, succeeding only if the resulting network is consistent. Intended for small-to-medium symbolic networks; not a large-scale temporal CSP solver.

Compilation flags:
static

Template:
refine_propagate(Network,Constraints,Closure,Changes)
Mode and number of proofs:
refine_propagate(+compound,+list(compound),-compound,-list(compound)) - zero_or_one

---

propagation_triple/2

Extracts the causing node triple triple(Node1, Node3, Node2) from a propagated change term.

Compilation flags:
    static

Template:
    propagation_triple(Change,Triple)
Mode and number of proofs:
    propagation_triple(+compound,-compound) - zero_or_one

---

propagation_triples/2

Collects the unique node triples that caused propagated refinements in a list of change terms.

Compilation flags:
    static

Template:
    propagation_triples(Changes,Triples)
Mode and number of proofs:
    propagation_triples(+list(compound),-list(compound)) - zero_or_one

---

path_consistency/2

Computes the path-consistency closure of a network by iterative refinement using relation-set composition and intersection. Intended for small-to-medium symbolic networks; not a large-scale temporal CSP solver.

Compilation flags:
    static

Template:
    path_consistency(Network,Closure)
Mode and number of proofs:
    path_consistency(+compound,-compound) - zero_or_one

---

constraints/2

Returns the explicit pair constraints in canonical storage order using constraint(Node1, Node2, RelationSet) terms.

Compilation flags:
    static

Template:
    constraints(Network,Constraints)
Mode and number of proofs:
    constraints(+compound,-list(compound)) - zero_or_one

---

subsumes/2

True if the first network is at least as general as the second network, i.e. every pair relation set in the second network is a subset of the corresponding relation set in the first network.

Compilation flags:
    static

Template:
    subsumes(Network1,Network2)
Mode and number of proofs:
    subsumes(+compound,+compound) - zero_or_one

---

equivalent/2

True if two networks contain the same nodes and mutually subsume each other.

Compilation flags:
    static

Template:

equivalent(Network1,Network2)
Mode and number of proofs:
equivalent(+compound,+compound) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

interval_relation_set_protocol, interval_relation_set, interval_algebra_protocol, interval_algebra

protocol

1.102.6 interval_protocol

Basic temporal interval relations protocol (based on James F. Allen Interval Algebra work).

Availability:
logtalk_load(intervals(loader))

Author: Paulo Moura
Version: 1:2:0
Date: 2026-05-13

Compilation flags:
static

Dependencies:
(none)

Remarks:

(none)

Inherited public predicates:
(none)

- Public predicates
  - new/3
  - valid/1
  - relation/3
  - before/2
  - after/2
  - meets/2
  - met_by/2
  - overlaps/2
  - overlapped_by/2
  - starts/2
  - started_by/2
  - during/2
  - contains/2
  - finishes/2
  - finished_by/2
  - equal/2
- Protected predicates
- Private predicates
- Operators

Public predicates

new/3

Constructs a new interval given start and end points. The start point must strictly precede the end point.

Compilation flags:
static

Template:

new(Start,End,Interval)
Mode and number of proofs:
new(@ground,@ground,-interval) - zero_or_one

---

valid/1

True if Interval is a valid interval.

Compilation flags:
static

Template:
valid(Interval)
Mode and number of proofs:
valid(@interval) - zero_or_one

---

relation/3

Classifies two valid intervals according to their unique Allen base relation.

Compilation flags:
static

Template:
relation(Interval1,Interval2,Relation)
Mode and number of proofs:
relation(@interval,@interval,?atom) - zero_or_one

---

before/2

True if Interval1 takes place before Interval2.

Compilation flags:
    static

Template:
    before(Interval1,Interval2)
Mode and number of proofs:
    before(@interval,@interval) - zero_or_one

---

after/2

True if Interval1 takes place after Interval2.

Compilation flags:
    static

Template:
    after(Interval1,Interval2)
Mode and number of proofs:
    after(@interval,@interval) - zero_or_one

---

meets/2

True if Interval1 meets Interval2.

Compilation flags:
    static

Template:
    meets(Interval1,Interval2)
Mode and number of proofs:
    meets(@interval,@interval) - zero_or_one

---

met_by/2

True if Interval1 is met by Interval2.

Compilation flags:
    static

Template:
    met_by(Interval1,Interval2)
Mode and number of proofs:
    met_by(@interval,@interval) - zero_or_one

---

overlaps/2

True if Interval1 overlaps with Interval2.

Compilation flags:
    static

Template:
    overlaps(Interval1,Interval2)
Mode and number of proofs:
    overlaps(@interval,@interval) - zero_or_one

---

overlapped_by/2

True if Interval1 is overlapped by Interval2.

Compilation flags:
    static

Template:
    overlapped_by(Interval1,Interval2)
Mode and number of proofs:
    overlapped_by(@interval,@interval) - zero_or_one

---

starts/2

True if Interval1 starts Interval2.

Compilation flags:
    static

Template:
    starts(Interval1,Interval2)
Mode and number of proofs:
    starts(@interval,@interval) - zero_or_one

---

started_by/2

True if Interval1 is started by Interval2.

Compilation flags:
    static

Template:
    started_by(Interval1,Interval2)
Mode and number of proofs:
    started_by(@interval,@interval) - zero_or_one

---

during/2

True if Interval1 occurs during Interval2.

Compilation flags:
    static

Template:
    during(Interval1,Interval2)
Mode and number of proofs:
    during(@interval,@interval) - zero_or_one

---

contains/2

True if Interval1 contains Interval2.

Compilation flags:
    static

Template:
    contains(Interval1,Interval2)
Mode and number of proofs:
    contains(@interval,@interval) - zero_or_one

---

finishes/2

True if Interval1 finishes Interval2.

Compilation flags:
    static

Template:
    finishes(Interval1,Interval2)
Mode and number of proofs:
    finishes(@interval,@interval) - zero_or_one

---

finished_by/2

True if Interval1 is finished by Interval2.

Compilation flags:
    static

Template:
    finished_by(Interval1,Interval2)
Mode and number of proofs:
    finished_by(@interval,@interval) - zero_or_one

---

equal/2

True if Interval1 is equal to Interval2.

Compilation flags:
    static

Template:
    equal(Interval1,Interval2)
Mode and number of proofs:
    equal(@interval,@interval) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

interval, interval_algebra_protocol, interval_algebra, interval_relation_set_protocol, interval_relation_set

object

1.102.7 interval_relation_set

Allen interval relation-set operations using canonical ordered duplicate-free lists of base relation atoms.

Availability:
    logtalk_load(intervals(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-13

Compilation flags:
static, context_switching_calls

Implements:
public interval_relation_set_protocol
Uses:
interval_algebra
list

Remarks:
(none)

Inherited public predicates:
compose/3 converse/2 empty/1 intersection/3 member/2 normalize/2 relation_set/1 singleton/2 subset/2 union/3 universal/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.102.8 interval_relation_set_protocol

Allen interval relation-set protocol using canonical ordered duplicate-free lists of base relation atoms.

Availability:
logtalk_load(intervals(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-13

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - relation_set/1
  - empty/1
  - universal/1
  - singleton/2
  - normalize/2
  - member/2
  - subset/2
  - intersection/3
  - union/3

    - converse/2
    - compose/3
- Protected predicates
- Private predicates
- Operators

Public predicates

relation_set/1

True if RelationSet is a canonical ordered duplicate-free list of Allen base relation atoms.

Compilation flags:
    static

Template:
    relation_set(RelationSet)
Mode and number of proofs:
    relation_set(?list(atom)) - zero_or_more

---

empty/1

Returns the empty Allen relation set.

Compilation flags:
    static

Template:
    empty(RelationSet)
Mode and number of proofs:
    empty(?list(atom)) - one

---

universal/1

Returns the universal Allen relation set containing all 13 base relation atoms.

Compilation flags:
    static

Template:
    universal(RelationSet)
Mode and number of proofs:
    universal(?list(atom)) - one

---

singleton/2

Relates an Allen base relation atom with its singleton relation set.

Compilation flags:
    static

Template:
    singleton(Relation,RelationSet)
Mode and number of proofs:
    singleton(?atom,?list(atom)) - zero_or_more

---

normalize/2

Normalizes a list of Allen base relation atoms into canonical ordered duplicate-free form.

Compilation flags:
    static

Template:
    normalize(Relations,RelationSet)
Mode and number of proofs:
    normalize(+list(atom),-list(atom)) - zero_or_one

---

member/2

True if Relation is a member of RelationSet.

Compilation flags:
    static

Template:
    member(Relation,RelationSet)
Mode and number of proofs:
    member(?atom,+list(atom)) - zero_or_more

---

subset/2

True if RelationSet1 is a subset of RelationSet2.

Compilation flags:
    static

Template:
    subset(RelationSet1,RelationSet2)
Mode and number of proofs:
    subset(+list(atom),+list(atom)) - zero_or_one

---

intersection/3

Computes the intersection of two relation sets.

Compilation flags:
    static

Template:
    intersection(RelationSet1,RelationSet2,Intersection)
Mode and number of proofs:
    intersection(+list(atom),+list(atom),-list(atom)) - zero_or_one

---

union/3

Computes the union of two relation sets.

Compilation flags:
    static

Template:
    union(RelationSet1,RelationSet2,Union)
Mode and number of proofs:
    union(+list(atom),+list(atom),-list(atom)) - zero_or_one

---

converse/2

Maps a relation set to the canonical ordered set of converse relations.

Compilation flags:
    static

Template:
    converse(RelationSet,ConverseSet)
Mode and number of proofs:
    converse(+list(atom),-list(atom)) - zero_or_one

---

compose/3

Computes the canonical ordered relation set resulting from composing two relation sets.

Compilation flags:
    static

Template:
    compose(RelationSet1,RelationSet2,Composition)
Mode and number of proofs:
    compose(+list(atom),+list(atom),-list(atom)) - zero_or_one

---

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

interval_algebra_protocol, interval_algebra, interval_relation_set

1.103 iqr_anomaly_detector

object

1.103.1 iqr_anomaly_detector

Statistical interquartile-range anomaly detector for continuous datasets. Learns per-attribute quartiles and interquartile ranges from baseline training examples selected from a dataset object implementing the anomaly_dataset_protocol protocol and returns a detector term that can be used for scoring, prediction, and export.

Availability:
logtalk_load(iqr_anomaly_detector(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public anomaly_detector_common
Uses:
format
list
numberlist

pairs
sample
type

Remarks:
(none)

Inherited public predicates:
anomaly_detector_options/2 check_anomaly_detector/1 check_option/1 check_options/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 predict/4 print_anomaly_detector/1 score/3 score_all/3 valid_anomaly_detector/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

anomaly_dataset_protocol, anomaly_detector_protocol, modified_z_score_anomaly_detector, z_score_anomaly_detector, isolation_forest_anomaly_detector, knn_distance_anomaly_detector, lof_anomaly_detector

1.104 iso8601

object

1.104.1 iso8601

ISO 8601 (and European civil calendar) compliant library of date and time predicates.

Availability:
logtalk_load(iso8601(loader))

Author: Daniel L. Dudley and Paulo Moura
Version: 1:3:0
Date: 2026-04-08

Compilation flags:
static, context_switching_calls

Uses:
list
os

Remarks:

- Scope: This object currently provides a powerful, versatile and efficient set of date-handling predicates, which–thanks to Logtalk–may be used as is on a wide range of Prolog compilers. Besides taking time to familiarize oneself with each predicate, the user should take note of the following information.
- Validation of dates: Date parts are not validated–that is the caller's responsibility! However, not being quite heartless yet, we do provide a predicate for this purpose.
- Date arithmetic: Many of the examples illustrate a simplified method of doing date arithmetic. Note, however, that we do not generally recommend this practice–it is all too easy to make mistakes. The safest way of finding the day difference between two dates is to first convert the dates to their Julian day numbers and then subtract one from the other. Similarly, the safe way to add or subtract a day offset to a particular date is to first convert the date to its Julian day number, add or subtract the day offset, and then convert the result to its corresponding date.
- BC years: ISO 8601 specifies that the Gregorian calendar be used, yet requires that years prior to 1 AD be handled arithmetically, i.e., the year we know as 1 BC is year 0, 2 BC is year -1, 3 BC is year -2 and so on. We do not follow ISO 8601 with regard to the handling of BC years. Our date predicates will accept and interpret an input year 0 as 1 BC; however, a negative year, Year, should always be interpreted as abs(Year) =:= Year BC. We believe that the average person will find our handling of BC years more user-friendly than the ISO 8601 one, but we encourage feedback from users with a view to a possible change in future versions.
- Week numbers: It is possible for a day (date) to have a week number that belongs to another year. Up to three of the first days of a calendar year may belong to the last week (number) of the prior calendar

year, and up to three days of the last days of a calendar year may belong to the first week (number) of the next calendar year. It for this reason that the Week parameter in date/6-7 is a compound term, namely week(WeekNo,ActualYear).

- Computation of Gregorian Easter Sunday: The algorithm is based upon the "Gaussian rule". Proleptic use is limited to years > 1582 AD, that is, after the introduction of the Gregorian calendar.
- Some Christian feast day offsets from Easter Sunday: Carnival Monday: -48 days, Mardi Gras (Shrove Tuesday): -47 days, Ash Wednesday: -46 days, Palm Sunday: -7 days, Easter Friday: -2 days, Easter Saturday: -1 day, Easter Monday: +1 day, Ascension of Christ: +39 days, Whitsunday: +49 days, Whitmonday: +50 days, Feast of Corpus Christi: +60 days.

Inherited public predicates:
(none)

- Public predicates
  - date/4
  - date/5
  - date/6
  - date/7
  - date_string/3
  - time_string/3
  - date_time_string/3
  - duration_string/2
  - interval_string/2
  - valid_date/3
  - leap_year/1
  - calendar_month/3
  - easter_day/3
- Protected predicates
- Private predicates
- Operators

Public predicates

date/4

Get the system date and/or its Julian Day # or convert a Julian Day # to/from given date parts.

Compilation flags:
    static

Template:
    date(JD,Year,Month,Day)
        JD - Julian day serial number.
        Year - 0 or negative if converted BC year, positive otherwise.
        Month - Normally an integer between 1 and 12 inclusive.
        Day - Normally an integer between 1 and 31 inclusive depending upon month.
Mode and number of proofs:
    date(?integer,?integer,?integer,?integer) - zero_or_one

Examples:
    Current date (i.e., today)
        date(JD,Year,Month,Day)
        JD=2453471,Year=2005,Month=4,Day=10
    Convert a date to its Julian day number
        date(JD,2000,2,29)
        JD=2451604
    Convert a Julian day number to its date
        date(2451604,Year,Month,Day)
        Year=2000,Month=2,Day=29
    What is the date of day # 60 in year 2000?
        date(JD,2000,1,60)
        JD=2451604
    What is the Julian of the 1st day prior to 2000-1-1?
        date(JD,2000,1,0)
        JD=2451544
    What is the Julian of the 60th day prior to 2000-1-1?
        date(JD,2000,1,-59)
        JD=2451485
    Illegal date is auto-adjusted (see also next query)
        date(JD,1900,2,29)
        JD=2415080
    This is the correct date!
        date(2415080,Year,Month,Day)
        Year=1900,Month=3,Day=1

date/5

Ditto date/4 + get/check its day-of-week #.

Compilation flags:
    static

Template:
    date(JD,Year,Month,Day,DoW)
        JD - Julian day serial number.
        Year - 0 or negative if converted BC year, positive otherwise.
        Month - Normally an integer between 1 and 12 inclusive.
        Day - Normally an integer between 1 and 31 inclusive depending upon month.
        DoW - Day of week, where Monday=1, Tuesday=2, ..., Sunday=7.
Mode and number of proofs:
    date(?integer,?integer,?integer,?integer,?integer) - zero_or_one

Examples:
    Get the Julian and the day-of-week # of a date
        date(JD,2000,2,29,DoW)
        JD=2451604,DoW=2
    Check the validity of a given date (day-of-week is 2, not 4)
        date(_,2002,3,5,4)
        no
    Get the Julian day of a given date if it is a Sunday
        date(JD,2004,2,29,7)
        JD=2453065
    Get the date and day-of-week # of a Julian
        date(2451545,Year,Month,Day,DoW)
        Year=2000,Month=1,Day=1,DoW=6

---

date/6

Ditto date/5 + get/check its week #.

Compilation flags:
    static

Template:
    date(JD,Year,Month,Day,DoW,Week)
        JD - Julian day serial number.

Year - 0 or negative if converted BC year, positive otherwise.
Month - Normally an integer between 1 and 12 inclusive.
Day - Normally an integer between 1 and 31 inclusive depending upon month.
DoW - Day of week, where Monday=1, Tuesday=2, ..., Sunday=7.
Week - Compound term, week(WeekNo,ActualYear), of a day.

Mode and number of proofs:

date(?integer,?integer,?integer,?integer,?integer,?compound) - zero_or_one

Examples:

Get the day-of-week and week number of a date

```
date(_,2000,1,1,DoW,Week)
DoW=6,Week=week(52,1999)
```

Get the week number and year of this week

```
date(_,_,_,_,_,Week)
Week=week(7,2004)
```

Get the Julian number and the week of a date if it is a Sunday

```
date(JD,2004,2,29,7,Week)
JD=2453065,Week=week(9,2004)
```

Get the day-of-week and week of a Julian day number

```
date(2453066,_,_,_,DoW,Week)
DoW=1,Week=week(10,2004)
```

Check that given date data matches

```
date(_,2004,3,1,1,week(10,2004))
yes
```

What is the date of a day of week (default is 1) in given week # and year?

```
date(_,Year,Month,Day,DoW,week(26,2004))
Year=2004,Month=6,Day=21,DoW=1
```

Ditto for Sunday

```
date(_,Year,Month,Day,7,week(1,2005))
Year=2005,Month=1,Day=9
```

Ditto for Tuesday in following week

```
date(_,Year,Month,Day,9,week(1,2005))
Year=2005,Month=1,Day=11
```

Ditto for Thursday in the prior week

```
date(_,Year,Month,Day,4,week(0,2005))
Year=2004,Month=12,Day=30
```

Ditto for Tuesday two weeks prior

```
date(_,Year,Month,Day,2,week(-1,2005))
Year=2004,Month=12,Day=21
```

Ditto for Saturday

```
date(_,Year,Month,Day,6,week(53,2004))
Year=2005,Month=1,Day=1
```

Ditto for Monday (note automatic compensation of nonexistent week number)

```
date(_,Year,Month,Day,1,week(60,2004))
Year=2005,Month=2,Day=14
```

date/7

Ditto date/6 + get/check its day-of-year #.

Compilation flags:
    static

Template:
    date(JD,Year,Month,Day,DoW,Week,DoY)
        JD - Julian day serial number.
        Year - 0 or negative if converted BC year, positive otherwise.
        Month - Normally an integer between 1 and 12 inclusive.
        Day - Normally an integer between 1 and 31 inclusive depending upon month.
        DoW - Day of week, where Monday=1, Tuesday=2, ..., Sunday=7.
        Week - Compound term, week(WeekNo,ActualYear), of a day.
        DoY - Day of year (NB! calendar year, not week # year).
Mode and number of proofs:
    date(?integer,?integer,?integer,?integer,?integer,?compound,?integer) - zero_or_one

Examples:
    Get the date and day-of-year of a Julian number
        date(2451649,Year,Month,Day,_,_,DoY)
        Year=2000,Month=4,Day=14,DoY=105
    Get the Julian number, week number and day-of-year of a date, confirming that it is a Sunday
        date(JD,2004,2,29,7,Week,DoY)
        JD=2453065,Week=week(9,2004),DoY=60
    Confirm that a date is, in fact, a specific day-of-year
        date(_,2004,3,1,_,_,61)
        yes
    Get the Julian number, week day and day-of-year of a date
        date(JD,2004,10,18,DoW,_,DoY)
        JD=2453297,DoW=1,DoY=292
    Get today's day-of-year
        date(_,_,_,_,_,_,DoY)
        DoY=54
    Get all missing date data (excl. Julian number) for the 60th calendar day of 2004
        date(_,2004,Month,Day,DoW,Week,60)
        Month=2,Day=29,DoW=7,Week=week(9,2004)
    Match given date data and, if true, return the missing data (excl. Julian number)
        date(_,2004,3,Day,DoW,Week,61)
        Day=1,DoW=1,Week=week(10,2004)
    Ditto (the 61st day-of-year cannot be both day 1 and 2 of the month)

date(_,2004,_,2,_,_,61)
no

---

date_string/3

Conversion between an ISO 8601 compliant date string and its components (truncated and expanded date representations are currently unsupported). Note that date components are not validated; that is the caller's responsibility!

Compilation flags:
static

Template:
date_string(Format,Components,String)
Format - ISO 8601 format.
Components - When bound and String is free, either a Julian number or a [Year,Month,Day] term; it binds to the system day/date if free When free and String is bound, it binds to an integer list representing the numeric elements of String.
String - ISO 8601 formatted string correspondent to Components.
Mode and number of proofs:
date_string(+atom,+integer,?atom) - zero_or_one
date_string(+atom,?list,?atom) - zero_or_one

Examples:
Date, complete, basic (section 5.2.1.1)
date_string('YYYYMMDD',[2004,2,29],String)
String='20040229'
Date, complete, basic (section 5.2.1.1)
date_string('YYYYMMDD',Components,'20040229')
Components=[2004,2,29]
Date, complete, extended (section 5.2.1.1)
date_string('YYYY-MM-DD',[2003,12,16],String)
String='2003-12-16'
Date, complete, extended (section 5.2.1.1)
date_string('YYYY-MM-DD',Components,'2003-12-16')
Components=[2003,12,16]
Date, complete, extended (section 5.2.1.1)
date_string('YYYY-MM-DD',_,String)
String='2004-02-17'
Date, complete, extended (section 5.2.1.1)
date_string('YYYY-MM-DD',Components,'2004-02-17')
Components=[2004,2,17]
Date, reduced, month (section 5.2.1.2 a)

```
        date_string('YYYY-MM',[2004,9,18],String)
        String='2004-09'
    Date, reduced, month (section 5.2.1.2 a)
        date_string('YYYY-MM',Components,'2004-09')
        Components=[2004,9]
    Date, reduced, year (section 5.2.1.2 b)
        date_string('YYYY',[1900,7,24],String)
        String='1900'
    Date, reduced, year (section 5.2.1.2 b)
        date_string('YYYY',Components,'1900')
        Components=[1900]
    Date, reduced, century (section 5.2.1.2 c)
        date_string('YY',2456557,String)
        String='20'
    Date, reduced, century (section 5.2.1.2 c)
        date_string('YY',Components,'20')
        Components=[20]
    Date, ordinal, complete (section 5.2.2.1)
        date_string('YYYYDDD',[2005,3,25],String)
        String='2005084'
    Date, ordinal, complete (section 5.2.2.1)
        date_string('YYYYDDD',Components,'2005084')
        Components=[2005,84]
    Date, ordinal, extended (section 5.2.2.1)
        date_string('YYYY-DDD',[1854,12,4],String)
        String='1854-338'
    Date, ordinal, extended (section 5.2.2.1)
        date_string('YYYY-DDD',Components,'1854-338')
        Components=[1854,338]
    Week, complete, basic (section 5.2.3.1)
        date_string('YYYYWwwD',[2000,1,2],String)
        String='1999W527'
    Week, complete, basic (section 5.2.3.1)
        date_string('YYYYWwwD',Components,'1999W527')
        Components=[1999,52,7]
    Week, complete, extended (section 5.2.3.1)
        date_string('YYYY-Www-D',[2003,12,29],String)
        String='2004-W01-1'
    Week, complete, extended (section 5.2.3.1)
        date_string('YYYY-Www-D',Components,'2004-W01-1')
        Components=[2004,1,1]
    Week, complete, extended (section 5.2.3.1)
        date_string('YYYY-Www-D',2453167,String)
        String='2004-W24-4'
    Week, complete, extended (section 5.2.3.1)
        date_string('YYYY-Www-D',Components,'2004-W24-4')
        Components=[2004,24,4]
```

Week, reduced, basic (section 5.2.3.2)
date_string('YYYYWww',[2004,2,29],String)
String='2004W09'
Week, reduced, basic (section 5.2.3.2)
date_string('YYYYWww',Components,'2004W09')
Components=[2004,9]
Week, reduced, extended (section 5.2.3.2)
date_string('YYYY-Www',[2004,2,29],String)
String='2004-W09'
Week, reduced, extended (section 5.2.3.2)
date_string('YYYY-Www',Components,'2004-W09')
Components=[2004,9]

---

time_string/3

Conversion between an ISO 8601 compliant time-of-day string and a time(Hours,Minutes,Seconds) term. Supported forms include basic and extended notations, with optional fractional seconds.

Compilation flags:
static

Template:
time_string(Format,Time,String)
Mode and number of proofs:
time_string(+atom,?compound,?atom) - zero_or_one

Examples:
Time, complete, extended
time_string('Thh:mm:ss',time(14,30,0),String)
String='T14:30:00'
Time, complete, basic
time_string('Thhmmss',Time,'T143000')
Time=time(14,30,0)
Time, complete, extended, fractional seconds
time_string('Thh:mm:ss.s',Time,'T14:30:00.125')
Time=time(14,30,0.125)
Time, complete, extended, comma fractional seconds
time_string('Thh:mm:ss,s',Time,'T14:30:00,125')
Time=time(14,30,0.125)

---

date_time_string/3

Conversion between an ISO 8601 compliant combined date-time string and a date_time/6 or date_time/7 term. Offset-aware terms use date_time(Year,Month,Day,Hours,Minutes,Seconds,OffsetSeconds) where OffsetSeconds is the UTC offset in seconds.

Compilation flags:
    static

Template:
    date_time_string(Format,DateTime,String)
Mode and number of proofs:
    date_time_string(+atom,?compound,?atom) - zero_or_one

Examples:
    Date-time, complete, extended
        date_time_string('YYYY-MM-DDThh:mm:ss',date_time(2026,4,7,14,30,0),String)
        String='2026-04-07T14:30:00'
    Date-time, ordinal, extended, UTC
        date_time_string('YYYY-DDDThh:mm:ssZ',DateTime,'2026-097T14:30:00Z')
        DateTime=date_time(2026,4,7,14,30,0,0)
    Date-time, week, complete, extended
        date_time_string('YYYY-Www-DThh:mm:ss',date_time(2026,4,7,14,30,0),String)
        String='2026-W15-2T14:30:00'
    Date-time, complete, basic, UTC
        date_time_string('YYYYMMDDThhmmssZ',DateTime,'20260407T143000Z')
        DateTime=date_time(2026,4,7,14,30,0,0)
    Date-time, complete, extended, offset
        date_time_string('YYYY-MM-DDThh:mm:ss+hh:mm',DateTime,
        '2026-04-07T14:30:00+05:45')
        DateTime=date_time(2026,4,7,14,30,0,20700)
    Date-time, complete, extended, comma fractional seconds
        date_time_string('YYYY-MM-DDThh:mm:ss,sZ',DateTime,'2026-04-07T14:30:00,125Z')
        DateTime=date_time(2026,4,7,14,30,0.125,0)

duration_string/2

Conversion between an ISO 8601 duration string and a duration(Years,Months,Days,Hours,Minutes,Seconds) term.

Compilation flags:
static

Template:
duration_string(Duration,String)
Mode and number of proofs:
duration_string(++compound,?atom) - zero_or_one
duration_string(?compound,+atom) - zero_or_one

Remarks:

- Duration string: An ISO 8601 duration string encodes a length of time (not a calendar date), starting with P and optionally using date and time parts such as P3D or P1Y2M3DT4H5M6S.

Examples:
Parse a duration string
duration_string(Duration,'PT45M')
Duration=duration(0,0,0,0,45,0)
Format a duration term
duration_string(duration(1,2,3,4,5,6),String)
String='P1Y2M3DT4H5M6S'

---

interval_string/2

Conversion between an ISO 8601 interval string and an interval(Start,End), interval(Start,Duration), or interval(Duration,End) term where endpoint terms are dates ([Year,Month,Day]) or date-times (date_time/6-7 terms).

Compilation flags:
static

Template:
interval_string(Interval,String)
Mode and number of proofs:
interval_string(++compound,?atom) - zero_or_one

interval_string(?compound,+atom) - zero_or_one

Remarks:

- Interval string: An ISO 8601 interval string encodes a time interval using two parts separated by /; each part is a date, a date-time, or a duration, e.g. 2026-02-25/2026-03-01, 2026-04-07T14:30:00Z/2026-04-07T15:00:00Z, or 2026-02-25/P3D.

Examples:

Parse a start/date + duration interval

```
interval_string(Interval,'2026-02-25/P3D')
Interval=interval([2026,2,25],duration(0,0,3,0,0,0))
```

Format a date/date interval

```
interval_string(interval([2026,2,25],[2026,3,1]),String)
String='2026-02-25/2026-03-01'
```

Format a date-time/date-time interval

```
interval_string(interval(date_time(2026,4,7,14,30,0,0),date_time(2026,4,7,15,0,0,0)),String)
String='2026-04-07T14:30:00Z/2026-04-07T15:00:00Z'
```

---

valid_date/3

Validate a given date in the Gregorian calendar.

Compilation flags:

static

Template:

valid_date(Year,Month,Day)

Mode and number of proofs:

valid_date(+integer,+integer,+integer) - zero_or_one

Examples:

Yes, the recent millennium was a leap year

```
valid_date(2000,2,29)
yes
```

2004 was also a leap year

```
valid_date(2004,2,29)
yes
```

Only 30 days in April

```
valid_date(2004,4,31)
no
```

1 BC was a leap year
valid_date(-1,2,29)
yes

---

leap_year/1

Succeed if given year is a leap year in the Gregorian calendar.

Compilation flags:
static

Template:
leap_year(Year)
Year - The Gregorian calendar year to investigate. If free, it binds to the system year.
Mode and number of proofs:
leap_year(?integer) - zero_or_one

Examples:
No, the prior centenary was not a leap year
leap_year(1900)
no
The recent millennium
leap_year(2000)
yes
This year
leap_year(Year)
Year=2004
This year (equivalent to prior query)
leap_year(_)
yes
Next centennial
leap_year(2100)
no
Year 0, equivalent to 1 BC
leap_year(0)
yes
1 BC
leap_year(-1)
yes
4 BC
leap_year(-4)
no

5 BC
leap_year(-5)
yes

---

calendar_month/3

Compute a calendar month.

Compilation flags:
static

Template:
calendar_month(Year,Month,Calendar)
Year - The calendar year.
Month - The calendar month.
Calendar - A compound term, m/3, composed of three main arguments specifying year, month, and a list of week and week day numbers (calendar body).
Mode and number of proofs:
calendar_month(?integer,?integer,-compound) - zero_or_one

Examples:
Compute the calendar of March, 2005
calendar_month(2005,3,Calendar)
Calendar=m(2005,3,[w(9,[0,1,2,3,4,5,6]),w(10,[7,8,9,10,11,12,13]),w(11,[14,15,16,17,18,19,20]),w(12,[21,22,23,24,25,26,27]),w(13,[28,29,30,31,0,0,0]),w(0,[0,0,0,0,0,0,0])])

---

easter_day/3

Compute a Gregorian Easter Sunday.

Compilation flags:
static

Template:
easter_day(Year,Month,Day)
Year - Integer specifying the year to be investigated.
Month - Month in which Easter Sunday falls for given year.
Day - Day of month in which Easter Sunday falls for given year.

Mode and number of proofs:
easter_day(?integer,-integer,-integer) - zero_or_one

Examples:
Compute Easter Sunday for a particular year
easter_day(2006,Month,Day)
Month=4,Day=16
Compute Easter Sunday for the current year
easter_day(Year,Month,Day)
Year=2005,Month=3,Day=27

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.105 iso_13616

object

1.105.1 iso_13616

ISO 13616 IBAN parser and normalizer based on the public ISO and SWIFT registry structure definitions and MOD-97 checksum algorithm.

Availability:
logtalk_load(iso_13616(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:

static, context_switching_calls

Implements:
public iso_13616_protocol
Uses:
iso_13616_registry
iso_3166

Remarks:
(none)

Inherited public predicates:
canonical_iban/2 formatted_iban/2 iban/4

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.105.2 iso_13616_protocol

Protocol for ISO 13616 IBAN structure parsing, checksum validation, and normalization.

Availability:
logtalk_load(iso_13616(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - iban/4
  - canonical_iban/2
  - formatted_iban/2
- Protected predicates
- Private predicates
- Operators

Public predicates

iban/4

Parses and validates an ISO 13616 IBAN atom into ISO 3166-1 alpha-2 country code, two-character check digits, and BBAN atom. Input may use lowercase letters and embedded spaces; successful results are normalized to uppercase compact components.

Compilation flags:
    static

Template:
    iban(IBAN,CountryAlpha2,CheckDigits,BBAN)
Mode and number of proofs:
    iban(+atom,?atom,?atom,?atom) - zero_or_one

---

canonical_iban/2

Returns the canonical electronic-format IBAN for a valid input by removing spaces and converting letters to uppercase.

Compilation flags:
    static

Template:
    canonical_iban(IBAN,Canonical)
Mode and number of proofs:
    canonical_iban(+atom,?atom) - zero_or_one

---

formatted_iban/2

Returns a valid IBAN formatted in groups of four characters separated by spaces.

Compilation flags:
    static

Template:

formatted_iban(IBAN,Formatted)
Mode and number of proofs:
formatted_iban(+atom,?atom) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

object

1.105.3 iso_13616_registry

Public SWIFT IBAN registry snapshot facts for ISO 13616 country-specific validation using a derived Prolog BBAN segment representation.

Availability:
logtalk_load(iso_13616(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:
static, context_switching_calls

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:

(none)

- Public predicates
  - country_spec/3
- Protected predicates
- Private predicates
- Operators

Public predicates

country_spec/3

Maps an IBAN country code to the registered total IBAN length and a derived BBAN structure term. BBANPattern is a list of Type-Count pairs, where Type is a for uppercase letters, n for digits, and c for uppercase alphanumeric characters. For example, [a-4, n-6, n-8] denotes four letters followed by six digits and then eight digits.

Compilation flags:
static

Template:
country_spec(CountryAlpha2,IBANLength,BBANPattern)
Mode and number of proofs:
country_spec(?atom,?integer,?list) - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.106 iso_3166

object

1.106.1 iso_3166

Facade object for ISO 3166 code queries.

Availability:
logtalk_load(iso_3166(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:
static, context_switching_calls

Implements:
public iso_3166_protocol
Uses:
iso_3166_1
iso_3166_2

Remarks:
(none)

Inherited public predicates:
country/4 subdivision/4

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.106.2 iso_3166_1

Generated ISO 3166-1 country facts extracted from the UN M49 overview page, which republishes ISO alpha-2 and alpha-3 codes together with M49 numeric codes.

Availability:
logtalk_load(iso_3166(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:
static, context_switching_calls

Dependencies:
(none)

Remarks:

- Source URL: https://unstats.un.org/unsd/methodology/m49/overview/
- Generated entries: 248

Inherited public predicates:
(none)

- Public predicates
  - country/4
- Protected predicates
- Private predicates
- Operators

Public predicates

country/4

Generated ISO 3166-1 country fact table.

Compilation flags:
static

Template:
country(Alpha2,Alpha3,Numeric,Name)
Mode and number of proofs:
country(?atom,?atom,?integer,?atom) - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

object

1.106.3 iso_3166_2

Generated ISO 3166-2 subdivision facts extracted from the Debian iso-codes machine-readable JSON snapshot.

Availability:
logtalk_load(iso_3166(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:
static, context_switching_calls

Dependencies:
(none)

Remarks:

- Source URL: https://salsa.debian.org/iso-codes-team/iso-codes/-/raw/main/data/iso_3166-2.json
- Generated entries: 5046

Inherited public predicates:
(none)

- Public predicates
  - subdivision/4
- Protected predicates
- Private predicates
- Operators

Public predicates

subdivision/4

Generated ISO 3166-2 subdivision fact table.

Compilation flags:
    static

Template:
    subdivision(Code,CountryAlpha2,Name,Category)
Mode and number of proofs:
    subdivision(?atom,?atom,?atom,?atom) - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

protocol

1.106.4 iso_3166_protocol

Protocol for ISO 3166 country and subdivision registry lookups.

Availability:
    logtalk_load(iso_3166(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:
    static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - country/4
  - subdivision/4
- Protected predicates
- Private predicates
- Operators

Public predicates

country/4

Enumerates ISO 3166-1 country entries as alpha-2, alpha-3, numeric, and English short name tuples.

Compilation flags:
static

Template:
country(Alpha2,Alpha3,Numeric,Name)
Mode and number of proofs:
country(?atom,?atom,?integer,?atom) - zero_or_more

subdivision/4

Enumerates ISO 3166-2 subdivision entries as code, parent alpha-2 country code, name, and category tuples.

Compilation flags:
    static

Template:
    subdivision(Code,CountryAlpha2,Name,Category)
Mode and number of proofs:
    subdivision(?atom,?atom,?atom,?atom) - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.107 iso_4217

object

1.107.1 iso_4217

Generated ISO 4217 active currency and fund facts extracted from the SIX list-one.xml snapshot.

Availability:
    logtalk_load(iso_4217(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:
static, context_switching_calls

Implements:
public iso_4217_protocol

Remarks:

- Source publication: 2026-01-01
- Generated currency entries: 269
- Generated fund entries: 8

Inherited public predicates:
(none)

- Public predicates
  - currency/5
  - fund_currency/5
- Protected predicates
- Private predicates
- Operators

Public predicates

currency/5

Generated ISO 4217 active non-fund currency fact table.

Compilation flags:
static

Template:
currency(Alphabetic,Numeric,MinorUnit,Name,Entity)
Mode and number of proofs:
currency(?atom,?integer,?term,?atom,?atom) - zero_or_more

fund_currency/5

Generated ISO 4217 active fund currency fact table.

Compilation flags:
    static

Template:
    fund_currency(Alphabetic,Numeric,MinorUnit,Name,Entity)
Mode and number of proofs:
    fund_currency(?atom,?integer,?term,?atom,?atom) - zero_or_more

---

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.107.2 iso_4217_protocol

Protocol for ISO 4217 active currency and fund code lookups.

Availability:
    logtalk_load(iso_4217(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:
    static

Dependencies:

(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - currency/5
  - fund_currency/5
- Protected predicates
- Private predicates
- Operators

Public predicates

currency/5

Enumerates ISO 4217 active non-fund entries as alphabetic code, numeric code, minor unit, currency name, and entity tuples. The minor unit is either an integer or the atom na when the standard lists N.A..

Compilation flags:
static

Template:
currency(Alphabetic,Numeric,MinorUnit,Name,Entity)
Mode and number of proofs:
currency(?atom,?integer,?term,?atom,?atom) - zero_or_more

fund_currency/5

Enumerates ISO 4217 active fund entries as alphabetic code, numeric code, minor unit, currency name, and entity tuples. The minor unit is either an integer or the atom na when the standard lists N.A..

Compilation flags:
    static

Template:
    fund_currency(Alphabetic,Numeric,MinorUnit,Name,Entity)
Mode and number of proofs:
    fund_currency(?atom,?integer,?term,?atom,?atom) - zero_or_more

---

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.108 iso_639

object

1.108.1 iso_639

Facade object for ISO 639 language and language-group queries.

Availability:
    logtalk_load(iso_639(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:
static, context_switching_calls

Implements:
public iso_639_protocol
Uses:
iso_639_1
iso_639_2
iso_639_3
iso_639_5

Remarks:
(none)

Inherited public predicates:
language/3 language/5 language_code/5 language_group/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.108.2 iso_639_1

Generated ISO 639-1 fact table.

Availability:
logtalk_load(iso_639(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:
static, context_switching_calls

Dependencies:
(none)

Remarks:

- Source update: 2026-01-15
- Generated entries: 183
- Source: Library of Congress ISO 639-2 code list.

Inherited public predicates:
(none)

- Public predicates
  - language/3
- Protected predicates
- Private predicates
- Operators

Public predicates

language/3

Generated ISO 639-1 fact table.

Compilation flags:
    static

Template:
    language(Alpha2,Alpha3,Name)
Mode and number of proofs:
    language(?atom,?atom,?atom) - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

object

1.108.3 iso_639_2

Generated ISO 639-2 fact table.

Availability:
    logtalk_load(iso_639(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:
    static, context_switching_calls

Dependencies:
(none)

Remarks:

- Source update: 2026-01-15
- Generated entries: 487
- Source: Library of Congress ISO 639-2 code list.

Inherited public predicates:
(none)

- Public predicates
  - language_code/5
- Protected predicates
- Private predicates
- Operators

Public predicates

language_code/5

Generated ISO 639-2 fact table.

Compilation flags:
static

Template:
language_code(Bibliographic,Terminologic,Alpha2,Name,Class)
Mode and number of proofs:
language_code(?atom,?atom,?atom,?atom,?atom) - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

object

1.108.4 iso_639_3

Generated ISO 639-3 fact table.

Availability:
logtalk_load(iso_639(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:
static, context_switching_calls

Dependencies:
(none)

Remarks:

- Source update: 2026-04-15
- Generated entries: 7929
- Source: SIL ISO 639-3 tab-delimited code set.

Inherited public predicates:
(none)

- Public predicates
  - language/5
- Protected predicates
- Private predicates
- Operators

Public predicates

language/5

Generated ISO 639-3 fact table.

Compilation flags:
    static

Template:
    language(Alpha3,Alpha2,Scope,Type,Name)
Mode and number of proofs:
    language(?atom,?atom,?atom,?atom,?atom) - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

object

1.108.5 iso_639_5

Generated ISO 639-5 fact table.

Availability:
logtalk_load(iso_639(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:
static, context_switching_calls

Dependencies:
(none)

Remarks:

- Source update: 2026-04-15
- Generated entries: 115
- Source: Library of Congress ISO 639-5 identifier list.

Inherited public predicates:
(none)

- Public predicates
  - language_group/2
- Protected predicates
- Private predicates
- Operators

Public predicates

language_group/2

Generated ISO 639-5 fact table.

Compilation flags:
    static

Template:
    language_group(Code,Name)
Mode and number of proofs:
    language_group(?atom,?atom) - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

protocol

1.108.6 iso_639_protocol

Protocol for ISO 639 language and language-group lookups.

Availability:
    logtalk_load(iso_639(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:
    static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - language/3
  - language_code/5
  - language/5
  - language_group/2
- Protected predicates
- Private predicates
- Operators

Public predicates

language/3

Enumerates ISO 639-1 entries as alpha-2, alpha-3, and English name tuples.

Compilation flags:
static

Template:
language(Alpha2,Alpha3,Name)
Mode and number of proofs:
language(?atom,?atom,?atom) - zero_or_more

language_code/5

Enumerates ISO 639-2 entries as bibliographic code, terminologic code, alpha-2 code, English name, and class tuples. The class is one of individual, macrolanguage, collective, special, or local_use.

Compilation flags:
    static

Template:
    language_code(Bibliographic,Terminologic,Alpha2,Name,Class)
Mode and number of proofs:
    language_code(?atom,?atom,?atom,?atom,?atom) - zero_or_more

---

language/5

Enumerates ISO 639-3 entries as alpha-3 code, alpha-2 code, scope, type, and English name tuples. The scope is one of individual, macrolanguage, collective, or special and the type is one of living, extinct, ancient, constructed, or special.

Compilation flags:
    static

Template:
    language(Alpha3,Alpha2,Scope,Type,Name)
Mode and number of proofs:
    language(?atom,?atom,?atom,?atom,?atom) - zero_or_more

---

language_group/2

Enumerates ISO 639-5 entries as group code and English name tuples.

Compilation flags:
    static

Template:
    language_group(Code,Name)
Mode and number of proofs:

language_group(?atom,?atom) - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.109 iso_9362

object

1.109.1 iso_9362

ISO 9362 BIC parser and normalizer based on the public Swift and ISO 9362 structure definitions.

Availability:
logtalk_load(iso_9362(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:
static, context_switching_calls

Implements:
public iso_9362_protocol
Uses:
iso_3166

Remarks:
(none)

Inherited public predicates:
bic/5 canonical_bic/2 primary_office_bic/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.109.2 iso_9362_protocol

Protocol for ISO 9362 BIC structure parsing and normalization.

Availability:
logtalk_load(iso_9362(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:
static

Dependencies:

(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - bic/5
  - canonical_bic/2
  - primary_office_bic/1
- Protected predicates
- Private predicates
- Operators

Public predicates

bic/5

Parses and validates an ISO 9362 BIC atom into business party prefix, ISO 3166-1 alpha-2 country code, business party suffix, and branch identifier. Eight-character BICs are normalized to the primary-office branch code XXX.

Compilation flags:
static

Template:
bic(BIC,Prefix,CountryAlpha2,Suffix,Branch)
Mode and number of proofs:
bic(+atom,?atom,?atom,?atom,?atom) - zero_or_one

canonical_bic/2

Returns the canonical 11-character form of a valid ISO 9362 BIC, using the branch code XXX when the input is an eight-character primary-office BIC.

Compilation flags:
    static

Template:
    canonical_bic(BIC,Canonical)
Mode and number of proofs:
    canonical_bic(+atom,?atom) - zero_or_one

---

primary_office_bic/1

True when a valid ISO 9362 BIC denotes the primary office, either because it is eight characters long or because its branch identifier is XXX.

Compilation flags:
    static

Template:
    primary_office_bic(BIC)
Mode and number of proofs:
    primary_office_bic(+atom) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.110 isolation_forest_anomaly_detector

object

1.110.1 isolation_forest_anomaly_detector

Extended Isolation Forest (EIF) algorithm for anomaly detection. Implements the improved version described by Hariri et al. (2019) that uses random hyperplane cuts instead of axis-aligned cuts, eliminating score bias artifacts. Builds an ensemble of isolation trees from baseline training examples selected from a dataset object implementing the anomaly_dataset_protocol protocol. Missing attribute values are represented using anonymous variables.

Availability:
    logtalk_load(isolation_forest_anomaly_detector(loader))

Author: Paulo Moura
Version: 2:0:0
Date: 2026-05-06

Compilation flags:
    static, context_switching_calls

Imports:
    public anomaly_detector_common
Uses:
    fast_random(Algorithm)
    format
    integer
    list
    numberlist
    pairs
    type

Remarks:
    (none)

Inherited public predicates:

anomaly_detector_options/2 check_anomaly_detector/1 check_option/1 check_options/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 predict/4 print_anomaly_detector/1 valid_anomaly_detector/1 valid_option/1 valid_options/1

- Public predicates
  - score/3
  - score_all/3
- Protected predicates
- Private predicates
- Operators

Public predicates

score/3

Computes the anomaly score for a given instance using the learned model. The instance is a list of Attribute-Value pairs where missing values are represented using anonymous variables. The score is in the range [0.0, 1.0]. Scores close to 1.0 indicate anomalies. Scores close to 0.5 or below indicate normal instances.

Compilation flags:
    static

Template:
    score(Model,Instance,Score)
Mode and number of proofs:
    score(+compound,+list,-float) - one

---

score_all/3

Computes the anomaly scores for all instances in the dataset. Returns a list of Id-Class-Score triples sorted by descending anomaly score.

Compilation flags:
    static

Template:

score_all(Dataset,Model,Scores)
Mode and number of proofs:
score_all(+object_identifier,+compound,-list) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

anomaly_dataset_protocol, anomaly_detector_protocol, knn_distance_anomaly_detector, lof_anomaly_detector

1.111 issue_creator

object

1.111.1 issue_creator

Support for automatically creating bug report issues for failed tests in GitHub or GitLab servers.

Availability:
logtalk_load(issue_creator(loader))

Author: Paulo Moura
Version: 0:12:1
Date: 2025-03-03

Compilation flags:
static, context_switching_calls

Provides:

logtalk::message_hook/4

Uses:

git
os
term_io
user

Remarks:

- Usage: This tool is automatically loaded and used from the logtalk_tester automation script when using its -b option. See the script man page for details.

Inherited public predicates:

(none)

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.112 java

object

1.112.1 java

Abstract interface to JPL API utility predicates.

Availability:
logtalk_load(java(loader))

Author: Paulo Moura
Version: 1:8:0
Date: 2023-03-15

Compilation flags:
static, context_switching_calls

Implements:
public java_utils_protocol
Uses:
user

Remarks:
(none)

Inherited public predicates:
array_list/2 array_to_list/2 array_to_terms/2 array_to_terms/3 decode_exception/2 decode_exception/3 false/1 is_false/1 is_null/1 is_object/1 is_true/1 is_void/1 iterator_element/2 list_to_array/2 map_element/2 null/1 set_element/2 terms_to_array/2 true/1 value_reference/2 void/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

java(Reference,ReturnValue), java(Reference), java_hook

object

1.112.2 java(Reference)

- Reference - Either a class name or a Java reference to an object.

Minimal abstraction of the JPL API for calling Java from Logtalk using familiar message-sending syntax and a forward/1 handler to resolve methods.

Availability:
logtalk_load(java(loader))

Author: Paulo Moura and Sergio Castro
Version: 1:0:1
Date: 2019-06-13

Compilation flags:
static, context_switching_calls

Extends:
public java(Reference,_)

Remarks:

- Usage: Send to this object any valid message as listed in the JavaDocs for the given reference.

Inherited public predicates:
forward/1 get_field/2 invoke/1 invoke/2 new/1 new/2 set_field/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

java(Reference,ReturnValue), java, java_hook

object

1.112.3 java(Reference,ReturnValue)

- Reference - Either a class name or a Java reference to an object.
- ReturnValue - Value returned by a method call (possibly the Java value void).

Minimal abstraction of the JPL API for calling Java from Logtalk using familiar message-sending syntax and a forward/1 handler to resolve methods.

Availability:
logtalk_load(java(loader))

Author: Paulo Moura and Sergio Castro
Version: 1:4:0
Date: 2023-03-13

Compilation flags:
static, context_switching_calls

Implements:
public forwarding
public java_access_protocol

Remarks:

- Usage: Send to this object any valid message as listed in the JavaDocs for the given reference.

Inherited public predicates:
forward/1 get_field/2 invoke/1 invoke/2 new/1 new/2 set_field/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

java(Reference), java, java_hook

protocol

1.112.4 java_access_protocol

Protocol for a minimal abstraction for calling Java from Logtalk using familiar message-sending syntax.

Availability:
logtalk_load(java(loader))

Author: Paulo Moura and Sergio Castro
Version: 1:2:1
Date: 2023-03-16

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - get_field/2
  - set_field/2
  - new/2
  - new/1
  - invoke/1

    - invoke/2
- Protected predicates
- Private predicates
- Operators

Public predicates

get_field/2

Gets the value of a class or object field.

Compilation flags:
    static

Template:
    get_field(Field,Value)
Mode and number of proofs:
    get_field(+atom,?nonvar) - zero_or_one

---

set_field/2

Sets the value of a class or object field.

Compilation flags:
    static

Template:
    set_field(Field,Value)
Mode and number of proofs:
    set_field(+atom,+nonvar) - one

---

new/2

Creates a new instance using the specified parameter values.

Compilation flags:
    static

Template:
    new(Parameters,Instance)
Mode and number of proofs:
    new(+list(nonvar),-reference) - one

---

new/1

Creates a new instance using default parameter values.

Compilation flags:
    static

Template:
    new(Instance)
Mode and number of proofs:
    new(-reference) - one

---

invoke/1

Invokes a method. This is a more efficient compared with relying on the forward/1 handler to resolve methods.

Compilation flags:
    static

Template:
    invoke(Method)
Mode and number of proofs:
    invoke(@nonvar) - one

---

invoke/2

Invokes a method. This is a more efficient compared with relying on the forward/1 handler to resolve methods.

Compilation flags:
    static

Template:
    invoke(Functor,Arguments)
Mode and number of proofs:
    invoke(@nonvar,@list) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

object

1.112.5 java_hook

Hook object to optimize messages to the java/1-2 objects that otherwise would trigger the forward/1 handler.

Availability:
    logtalk_load(java(loader))

Author: Paulo Moura
Version: 1:0:1
Date: 2019-06-13

Compilation flags:
static, context_switching_calls

Implements:
public expanding

Remarks:

- Usage: Compile source files with messages to the java/1-2 objects using the compiler option hook(java_hook).

Inherited public predicates:
goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

java(Reference,ReturnValue), java(Reference)

protocol

1.112.6 java_utils_protocol

Abstract interface to Java utility predicates.

Availability:
logtalk_load(java(loader))

Author: Paulo Moura
Version: 1:6:0
Date: 2023-03-13

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - value_reference/2
  - true/1
  - false/1
  - void/1
  - null/1
  - is_true/1
  - is_false/1
  - is_void/1
  - is_null/1
  - is_object/1
  - terms_to_array/2
  - array_to_terms/3
  - array_to_terms/2

  - array_to_list/2
  - list_to_array/2
  - array_list/2
  - iterator_element/2
  - map_element/2
  - set_element/2
  - decode_exception/2
  - decode_exception/3
- Protected predicates
- Private predicates
- Operators

Public predicates

value_reference/2

Returns an opaque term that represents the Java value with the given name.

Compilation flags:
    static

Template:
    value_reference(Value,Reference)
Mode and number of proofs:
    value_reference(?atom,--ground) - one_or_more

---

true/1

Returns an opaque term that represents the Java value true.

Compilation flags:
    static

Template:
    true(Reference)
Mode and number of proofs:

true(--ground) - one

---

false/1

Returns an opaque term that represents the Java value false.

Compilation flags:
static

Template:
false(Reference)
Mode and number of proofs:
false(--ground) - one

---

void/1

Returns an opaque term that represents the Java value void.

Compilation flags:
static

Template:
void(Reference)
Mode and number of proofs:
void(--ground) - one

---

null/1

Returns an opaque term that represents the Java value null.

Compilation flags:
static

Template:

null(Reference)
Mode and number of proofs:
null(--ground) - one

---

is_true/1

True when the argument is the Java value true. Fails if the argument is not instantiated.

Compilation flags:
static

Template:
is_true(Reference)
Mode and number of proofs:
is_true(@term) - zero_or_one

---

is_false/1

True when the argument is the Java value false. Fails if the argument is not instantiated.

Compilation flags:
static

Template:
is_false(Reference)
Mode and number of proofs:
is_false(@term) - zero_or_one

---

is_void/1

True when the argument is the Java value void. Fails if the argument is not instantiated.

Compilation flags:
    static

Template:
    is_void(Reference)
Mode and number of proofs:
    is_void(@term) - zero_or_one

---

is_null/1

True when the argument is the Java value null. Fails if the argument is not instantiated.

Compilation flags:
    static

Template:
    is_null(Reference)
Mode and number of proofs:
    is_null(@term) - zero_or_one

---

is_object/1

True when the argument is a reference to a Java object. Fails if the argument is not instantiated.

Compilation flags:
    static

Template:
    is_object(Reference)
Mode and number of proofs:
    is_object(@term) - zero_or_one

---

terms_to_array/2

Converts a list of ground Prolog terms to an array (a Java reference).

Compilation flags:
    static

Template:
    terms_to_array(Terms,Array)
Mode and number of proofs:
    terms_to_array(++list(ground),-array) - one

---

array_to_terms/3

Converts an array (a Java reference) to a list of ground Prolog terms returning also its length. The array elements must be atoms, integers, floats, or compound terms. Fails otherwise.

Compilation flags:
    static

Template:
    array_to_terms(Array,Terms,Length)
Mode and number of proofs:
    array_to_terms(+array,-list(ground),-integer) - one

---

array_to_terms/2

Converts an array (a Java reference) to a list of ground Prolog terms. The array elements must be atoms, integers, floats, or ground compound terms. Fails otherwise.

Compilation flags:
    static

Template:
    array_to_terms(Array,Terms)
Mode and number of proofs:
    array_to_terms(+array,-list(term)) - one

---

array_to_list/2

Converts an array (a Java reference) to a list of Java references or their values.

Compilation flags:
    static

Template:
    array_to_list(Array,List)
Mode and number of proofs:
    array_to_list(+array,-list) - one

---

list_to_array/2

Converts a list of Java references or values to an array (a Java reference).

Compilation flags:
    static

Template:
    list_to_array(List,Array)
Mode and number of proofs:
    list_to_array(+list,-array) - one

---

array_list/2

Converts between an array (a Java reference) and a list of Java references or their values. Deprecated. Use the array_to_list/2 and list_to_array/2 predicates instead.

Compilation flags:
    static

Template:
    array_list(Array,List)

Mode and number of proofs:
array_list(+array,-list) - one
array_list(-array,+list) - one

---

iterator_element/2

Enumerates, by backtracking, all iterator elements.

Compilation flags:
static

Template:
iterator_element(Iterator,Element)
Mode and number of proofs:
iterator_element(+iterator,-element) - zero_or_more

---

map_element/2

Enumerates, by backtracking, all map elements.

Compilation flags:
static

Template:
map_element(Map,Element)
Mode and number of proofs:
map_element(+iterator,-element) - zero_or_more

---

set_element/2

Enumerates, by backtracking, all set elements.

Compilation flags:
    static

Template:
    set_element(Set,Element)
Mode and number of proofs:
    set_element(+iterator,-element) - zero_or_more

---

decode_exception/2

Decodes an exception into its corresponding cause. Fails if the exception is not a Java exception.

Compilation flags:
    static

Template:
    decode_exception(Exception,Cause)
Mode and number of proofs:
    decode_exception(+callable,-atom) - zero_or_one

---

decode_exception/3

Decodes an exception into its corresponding cause and a stack trace. Fails if the exception is not a Java exception.

Compilation flags:
    static

Template:
    decode_exception(Exception,Cause,StackTrace)
Mode and number of proofs:
    decode_exception(+callable,-atom,-list(atom)) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.113 json

object

1.113.1 json

JSON parser and generator. Uses curly terms for parsed JSON objects, dashes for parsed JSON pairs, and atoms for parsed JSON strings.

Availability:
logtalk_load(json(loader))

Author: Paulo Moura and Jacinto Dávila
Version: 1:1:0
Date: 2022-11-14

Compilation flags:
static, context_switching_calls

Extends:
public json(curly,dash,atom)

Remarks:
(none)

Inherited public predicates:
generate/2 parse/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.113.2 json(StringRepresentation)

- StringRepresentation - Text representation to be used when decoding JSON strings. Possible values are atom (default), chars, and codes.

JSON parser and generator. Uses curly terms for parsed JSON objects and dashes for parsed JSON pairs.

Availability:
    logtalk_load(json(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2022-11-14

Compilation flags:
    static, context_switching_calls

Extends:
    public json(curly,dash,StringRepresentation)

Remarks:
(none)

Inherited public predicates:
generate/2 parse/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.113.3 json(ObjectRepresentation,PairRepresentation,StringRepresentation)

- ObjectRepresentation - Object representation to be used when decoding JSON objects. Possible values are curly (default) and list.
- PairRepresentation - Pair representation to be used when decoding JSON objects. Possible values are dash (default), equal, and colon.
- StringRepresentation - Text representation to be used when decoding JSON strings. Possible values are atom (default), chars, and codes.

JSON parser and generator.

Availability:
logtalk_load(json(loader))

Author: Paulo Moura and Jacinto Dávila
Version: 0:14:0
Date: 2026-01-25

Compilation flags:
static, context_switching_calls

Implements:
public json_protocol
Uses:
reader

Remarks:
(none)

Inherited public predicates:
generate/2 parse/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.113.4 json_protocol

JSON parser and generator protocol.

Availability:
logtalk_load(json(loader))

Author: Paulo Moura and Jacinto Dávila
Version: 0:11:0
Date: 2022-11-09

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - parse/2
  - generate/2
- Protected predicates
- Private predicates
- Operators

Public predicates

parse/2

Parses the JSON contents read from the given source (codes(List), stream(Stream), line(Stream), file(Path), chars(List), or atom(Atom)) into a term. Fails if the JSON contents cannot be parsed.

Compilation flags:
    static

Template:
    parse(Source,Term)
Mode and number of proofs:
    parse(++compound,--term) - one_or_error

---

generate/2

Generates the content using the representation specified in the first argument (codes(List), stream(Stream), file(Path), chars(List), or atom(Atom)) for the term in the second argument. Fails if this term cannot be processed.

Compilation flags:
    static

Template:
    generate(Sink,Term)
Mode and number of proofs:
    generate(+compound,++term) - one_or_error

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.114 json_ld

object

1.114.1 json_ld

JSON-LD 1.1 parser, generator, and processor. Uses curly terms for parsed JSON objects, dashes for parsed JSON pairs, and atoms for parsed JSON strings.

Availability:
logtalk_load(json_ld(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-05

Compilation flags:
static, context_switching_calls

Extends:
public json_ld(curly,dash,atom)

Remarks:
(none)

Inherited public predicates:
compact/3 expand/2 flatten/2 generate/2 parse/2

- Public predicates
- Protected predicates
- Private predicates

- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.114.2 json_ld(StringRepresentation)

- StringRepresentation - Text representation to be used when decoding JSON strings. Possible values are atom (default), chars, and codes.

JSON-LD 1.1 parser, generator, and processor. Uses curly terms for parsed JSON objects and dashes for parsed JSON pairs.

Availability:
logtalk_load(json_ld(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-05

Compilation flags:
static, context_switching_calls

Extends:
public json_ld(curly,dash,StringRepresentation)

Remarks:
(none)

Inherited public predicates:
compact/3 expand/2 flatten/2 generate/2 parse/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.114.3 json_ld(ObjectRepresentation,PairRepresentation,StringRepresentation)

- ObjectRepresentation - Object representation to be used when decoding JSON objects. Possible values are curly (default) and list.
- PairRepresentation - Pair representation to be used when decoding JSON objects. Possible values are dash (default), equal, and colon.
- StringRepresentation - Text representation to be used when decoding JSON strings. Possible values are atom (default), chars, and codes.

JSON-LD 1.1 parser, generator, and processor. Builds on top of the json library for JSON parsing and generation.

Availability:
logtalk_load(json_ld(loader))

Author: Paulo Moura

Version: 1:1:1
Date: 2026-04-22

Compilation flags:
static, context_switching_calls

Implements:
public json_ld_protocol
Uses:
json(ObjectRepresentation,PairRepresentation,StringRepresentation)
list

Remarks:
(none)

Inherited public predicates:
compact/3 expand/2 flatten/2 generate/2 parse/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.114.4 json_ld_protocol

JSON-LD 1.1 parser, generator, and processor protocol.

Availability:
logtalk_load(json_ld(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-05

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - parse/2
  - generate/2
  - expand/2
  - compact/3
  - flatten/2
- Protected predicates
- Private predicates
- Operators

Public predicates

parse/2

Parses a JSON-LD document from the given source (file(Path), stream(Stream), codes(List), chars(List), or atom(Atom)) into a term.

Compilation flags:
    static

Template:
    parse(Source,Term)
Mode and number of proofs:
    parse(++compound,--term) - one_or_error

Exceptions:
    Source is a variable:
        instantiation_error
    Source is neither a variable nor a valid source:
        domain_error(json_ld_source,Source)

---

generate/2

Generates the content using the representation specified in the first argument (file(Path), stream(Stream), codes(List), chars(List), or atom(Atom)) for the term in the second argument.

Compilation flags:
    static

Template:
    generate(Sink,Term)
Mode and number of proofs:
    generate(+compound,++term) - one_or_error

Exceptions:
    Sink is a variable:
        instantiation_error
    Sink cannot be generated:
        domain_error(json_ld_sink,Sink)

expand/2

Expands a parsed JSON-LD document. Expansion removes the context and represents all properties and types as full IRIs. The result is a list of node objects in expanded document form.

Compilation flags:
    static

Template:
    expand(Document,Expanded)
Mode and number of proofs:
    expand(+term,--list) - one_or_error

Exceptions:
    Document is a variable:
        instantiation_error

compact/3

Compacts an expanded JSON-LD document using the given context. Compaction applies the context to shorten IRIs to terms or compact IRIs.

Compilation flags:
    static

Template:
    compact(Document,Context,Compacted)
Mode and number of proofs:
    compact(+term,+term,--term) - one_or_error

Exceptions:
    Document is a variable:
        instantiation_error
    Context is a variable:
        instantiation_error

flatten/2

Flattens an expanded JSON-LD document. Flattening collects all node objects into a flat @graph array, with nested nodes replaced by references. Blank node identifiers are generated for nodes without @id.

Compilation flags:
    static

Template:
    flatten(Document,Flattened)
Mode and number of proofs:
    flatten(+term,--term) - one_or_error

Exceptions:
    Document is a variable:
        instantiation_error

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.115 json_lines

object

1.115.1 json_lines

JSON Lines parser and generator. Uses curly terms for parsed JSON objects, dashes for parsed JSON pairs, and atoms for parsed JSON strings.

Availability:
logtalk_load(json_lines(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2025-05-27

Compilation flags:
static, context_switching_calls

Extends:
public json_lines(curly,dash,atom)

Remarks:
(none)

Inherited public predicates:
generate/2 parse/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.115.2 json_lines(StringRepresentation)

- StringRepresentation - Text representation to be used when decoding JSON strings. Possible values are atom (default), chars, and codes.

JSON Lines parser and generator. Uses curly terms for parsed JSON objects and dashes for parsed JSON pairs.

Availability:
logtalk_load(json_lines(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2025-05-27

Compilation flags:
static, context_switching_calls

Extends:
public json_lines(curly,dash,StringRepresentation)

Remarks:
(none)

Inherited public predicates:
generate/2 parse/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.115.3 json_lines(ObjectRepresentation,PairRepresentation,StringRepresentation)

- ObjectRepresentation - Object representation to be used when decoding JSON objects. Possible values are curly (default) and list.
- PairRepresentation - Pair representation to be used when decoding JSON objects. Possible values are dash (default), equal, and colon.
- StringRepresentation - Text representation to be used when decoding JSON strings. Possible values are atom (default), chars, and codes.

JSON Lines parser and generator.

Availability:
logtalk_load(json_lines(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-01-25

Compilation flags:
static, context_switching_calls

Implements:
public json_lines_protocol
Uses:
reader

Remarks:
(none)

Inherited public predicates:
generate/2 parse/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.115.4 json_lines_protocol

JSON Lines parser and generator protocol.

Availability:
logtalk_load(json_lines(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2025-05-27

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - parse/2
  - generate/2
- Protected predicates
- Private predicates
- Operators

Public predicates

parse/2

Parses the JSON Lines contents read from the given source (file(Path), stream(Stream), line(Stream), codes(Codes), chars(Chars), or atom(Atom)) into a list of ground terms.

Compilation flags:
static

Template:
parse(Source,Terms)
Mode and number of proofs:
parse(++compound,--list(ground)) - one_or_error

Exceptions:
Source is a variable:
instantiation_error
Source is neither a variable nor a valid source:
domain_error(json_lines_source,Source)

---

generate/2

Generates the content using the representation specified in the first argument (file(Path), stream(Stream), codes(Codes), chars(Chars), or atom(Atom)) for the list of ground terms in the second argument.

Compilation flags:
static

Template:
generate(Sink,Terms)
Mode and number of proofs:
generate(+compound,++list(ground)) - one_or_error

Exceptions:
Sink is a variable:
instantiation_error
Terms is a variable:
instantiation_error
Terms is neither a variable nor a list:
type_error(list,Terms)
Term is a non-ground element of the list Terms:
instantiation_error
Term is an element of the list Terms but not a valid JSON term:
domain_error(json_term,Term)
Sink cannot be generated:
domain_error(json_lines_sink,Sink)

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.116 json_pointer

object

1.116.1 json_pointer

JSON Pointer (RFC 6901) parser, generator, and evaluator using atoms for text representation.

Availability:
    logtalk_load(json_pointer(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-11

Compilation flags:
    static, context_switching_calls

Extends:
    public json_pointer(atom)

Remarks:
    (none)

Inherited public predicates:
    evaluate/3 generate/2 generate_fragment/2 parse/2 parse_fragment/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.116.2 json_pointer(StringRepresentation)

- StringRepresentation - Text representation to be used for reference tokens. Possible values are atom (default), chars, and codes.

JSON Pointer (RFC 6901) parser, generator, and evaluator.

Availability:
logtalk_load(json_pointer(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-11

Compilation flags:
static, context_switching_calls

Implements:
public json_pointer_protocol

Uses:
list
utf_8

Remarks:
(none)

Inherited public predicates:
evaluate/3 generate/2 generate_fragment/2 parse/2 parse_fragment/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.116.3 json_pointer_protocol

JSON Pointer (RFC 6901) parser, generator, and evaluator protocol.

Availability:
logtalk_load(json_pointer(loader))

Author: Paulo Moura

Version: 1:0:0
Date: 2026-05-11

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - parse/2
  - generate/2
  - parse_fragment/2
  - generate_fragment/2
  - evaluate/3
- Protected predicates
- Private predicates
- Operators

Public predicates

parse/2

Parses a JSON Pointer from the given source (codes(Codes), chars(Chars), or atom(Atom)) into a list of reference tokens. Fails if the pointer cannot be parsed.

Compilation flags:
static

Template:
parse(Source,Pointer)
Mode and number of proofs:

parse(++compound,--list(ground)) - zero_or_one

Exceptions:
Source is a variable:
instantiation_error
Source is neither a variable nor a valid source:
domain_error(json_pointer_source,Source)

---

generate/2

Generates a JSON Pointer using the representation specified in the first argument (codes(Codes), chars(Chars), or atom(Atom)) for the list of reference tokens in the second argument.

Compilation flags:
static

Template:
generate(Sink,Pointer)
Mode and number of proofs:
generate(+compound,++list(ground)) - one_or_error

Exceptions:
Sink is a variable:
instantiation_error
Pointer is neither a variable nor a list:
type_error(list,Pointer)
Pointer contains an invalid reference token:
domain_error(json_pointer_token,Token)
Sink cannot be generated:
domain_error(json_pointer_sink,Sink)

---

parse_fragment/2

Parses a URI fragment representation of a JSON Pointer from the given source (codes(Codes), chars(Chars), or atom(Atom)) into a list of reference tokens. Fails if the fragment cannot be parsed.

Compilation flags:
    static

Template:
    parse_fragment(Source,Pointer)
Mode and number of proofs:
    parse_fragment(++compound,--list(ground)) - zero_or_one

Exceptions:
    Source is a variable:
        instantiation_error
    Source is neither a variable nor a valid source:
        domain_error(json_pointer_fragment_source,Source)

---

generate_fragment/2

Generates the URI fragment representation of a JSON Pointer using the representation specified in the first argument (codes(Codes), chars(Chars), or atom(Atom)) for the list of reference tokens in the second argument.

Compilation flags:
    static

Template:
    generate_fragment(Sink,Pointer)
Mode and number of proofs:
    generate_fragment(+compound,++list(ground)) - one_or_error

Exceptions:
    Sink is a variable:
        instantiation_error
    Pointer is neither a variable nor a list:
        type_error(list,Pointer)
    Pointer contains an invalid reference token:
        domain_error(json_pointer_token,Token)

Sink cannot be generated:
domain_error(json_pointer_fragment_sink,Sink)

---

evaluate/3

Evaluates a parsed JSON Pointer against a JSON term. Fails if the pointer does not identify a value in the given JSON term.

Compilation flags:
static

Template:
evaluate(Pointer,JSON,Value)
Mode and number of proofs:
evaluate(++list(ground),++term,?term) - zero_or_one

Exceptions:
Pointer is a variable:
instantiation_error
JSON is a variable:
instantiation_error
Pointer is neither a variable nor a list:
type_error(list,Pointer)
Pointer contains an invalid reference token:
domain_error(json_pointer_token,Token)

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.117 json_rpc

object

1.117.1 json_rpc

JSON-RPC 2.0 protocol encoding and decoding. Provides predicates for constructing and parsing JSON-RPC 2.0 request, notification, response, and error objects. Uses the json library for JSON parsing and generation.

Availability:
logtalk_load(json_rpc(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-24

Compilation flags:
static, context_switching_calls

Uses:
json
list
reader

Remarks:

- Specification: Implements the JSON-RPC 2.0 specification: https://www.jsonrpc.org/specification
- JSON representation: Uses the json library default representation: curly terms for objects, dashes for pairs, and atoms for strings.
- Request: A JSON-RPC 2.0 request is represented as {jsonrpc-'2.0', method-Method, params-Params, id-Id}.
- Notification: A JSON-RPC 2.0 notification is represented as {jsonrpc-'2.0', method-Method, params-Params}.
- Successful response: A JSON-RPC 2.0 successful response is represented as {jsonrpc-'2.0', result-Result, id-Id}.
- Error response: A JSON-RPC 2.0 error response is represented as {jsonrpc-'2.0', error-{code-Code, message-Message}, id-Id} or {jsonrpc-'2.0', error-{code-Code, message-Message, data-Data}, id-Id}.

- Batch request: A JSON-RPC 2.0 batch request is represented as a list of request and/or notification terms.
- Error codes: Standard error codes: -32700 (parse error), -32600 (invalid request), -32601 (method not found), -32602 (invalid params), -32603 (internal error). Server errors: -32000 to -32099.

Inherited public predicates:
(none)

- Public predicates
  - request/4
  - request/3
  - notification/3
  - notification/2
  - response/3
  - error_response/4
  - error_response/5
  - parse_error/1
  - invalid_request/1
  - method_not_found/2
  - invalid_params/2
  - internal_error/2
  - encode/2
  - decode/2
  - is_request/1
  - is_notification/1
  - is_response/1
  - is_error_response/1
  - is_batch/1
  - id/2
  - method/2
  - params/2
  - result/2
  - error/2
  - error_code/2
  - error_message/2

    - error_data/2
    - write_message/2
    - read_message/2
    - write_framed_message/2
    - read_framed_message/2
- Protected predicates
- Private predicates
- Operators

Public predicates

request/4

Constructs a JSON-RPC 2.0 request term.

Compilation flags:
    static

Template:
    request(Method,Params,Id,Request)
Mode and number of proofs:
    request(+atom,+list,+nonvar,--compound) - one

---

request/3

Constructs a JSON-RPC 2.0 request term with no parameters.

Compilation flags:
    static

Template:
    request(Method,Id,Request)
Mode and number of proofs:
    request(+atom,+nonvar,--compound) - one

---

notification/3

Constructs a JSON-RPC 2.0 notification term (a request without an id).

Compilation flags:
    static

Template:
    notification(Method,Params,Notification)
Mode and number of proofs:
    notification(+atom,+list,--compound) - one

---

notification/2

Constructs a JSON-RPC 2.0 notification term with no parameters.

Compilation flags:
    static

Template:
    notification(Method,Notification)
Mode and number of proofs:
    notification(+atom,--compound) - one

---

response/3

Constructs a JSON-RPC 2.0 successful response term.

Compilation flags:
    static

Template:
    response(Result,Id,Response)
Mode and number of proofs:
    response(+nonvar,+nonvar,--compound) - one

---

error_response/4

Constructs a JSON-RPC 2.0 error response term with a null id (used when the request id cannot be determined).

Compilation flags:
    static

Template:
    error_response(Code,Message,Id,ErrorResponse)
Mode and number of proofs:
    error_response(+integer,+atom,+nonvar,--compound) - one

---

error_response/5

Constructs a JSON-RPC 2.0 error response term with additional error data.

Compilation flags:
    static

Template:
    error_response(Code,Message,Data,Id,ErrorResponse)
Mode and number of proofs:
    error_response(+integer,+atom,+nonvar,+nonvar,--compound) - one

---

parse_error/1

Constructs a JSON-RPC 2.0 parse error response (-32700) with a null id.

Compilation flags:
    static

Template:
    parse_error(ErrorResponse)
Mode and number of proofs:
    parse_error(--compound) - one

---

invalid_request/1

Constructs a JSON-RPC 2.0 invalid request error response (-32600) with a null id.

Compilation flags:
    static

Template:
    invalid_request(ErrorResponse)
Mode and number of proofs:
    invalid_request(--compound) - one

---

method_not_found/2

Constructs a JSON-RPC 2.0 method not found error response (-32601).

Compilation flags:
    static

Template:
    method_not_found(Id,ErrorResponse)
Mode and number of proofs:
    method_not_found(+nonvar,--compound) - one

---

invalid_params/2

Constructs a JSON-RPC 2.0 invalid params error response (-32602).

Compilation flags:
    static

Template:
    invalid_params(Id,ErrorResponse)
Mode and number of proofs:

invalid_params(+nonvar,--compound) - one

---

internal_error/2

Constructs a JSON-RPC 2.0 internal error response (-32603).

Compilation flags:
static

Template:
internal_error(Id,ErrorResponse)
Mode and number of proofs:
internal_error(+nonvar,--compound) - one

---

encode/2

Encodes a JSON-RPC 2.0 term (request, notification, response, error, or batch) into a JSON atom.

Compilation flags:
static

Template:
encode(Term,JSON)
Mode and number of proofs:
encode(+compound,--atom) - one

---

decode/2

Decodes a JSON atom into a JSON-RPC 2.0 term (request, notification, response, error, or batch).

Compilation flags:
static

Template:

decode(JSON,Term)
Mode and number of proofs:
decode(+atom,--compound) - one_or_error

---

is_request/1

True if the term is a valid JSON-RPC 2.0 request (has jsonrpc, method, and id fields).

Compilation flags:
static

Template:
is_request(Term)
Mode and number of proofs:
is_request(+compound) - zero_or_one

---

is_notification/1

True if the term is a valid JSON-RPC 2.0 notification (has jsonrpc and method fields but no id field).

Compilation flags:
static

Template:
is_notification(Term)
Mode and number of proofs:
is_notification(+compound) - zero_or_one

---

is_response/1

True if the term is a valid JSON-RPC 2.0 successful response (has jsonrpc, result, and id fields).

Compilation flags:
    static

Template:
    is_response(Term)
Mode and number of proofs:
    is_response(+compound) - zero_or_one

---

is_error_response/1

True if the term is a valid JSON-RPC 2.0 error response (has jsonrpc, error, and id fields).

Compilation flags:
    static

Template:
    is_error_response(Term)
Mode and number of proofs:
    is_error_response(+compound) - zero_or_one

---

is_batch/1

True if the term is a valid JSON-RPC 2.0 batch (a non-empty list).

Compilation flags:
    static

Template:
    is_batch(Term)
Mode and number of proofs:
    is_batch(+compound) - zero_or_one

---

id/2

Extracts the id field from a JSON-RPC 2.0 message.

Compilation flags:
    static

Template:
    id(Message,Id)
Mode and number of proofs:
    id(+compound,--nonvar) - zero_or_one

---

method/2

Extracts the method field from a JSON-RPC 2.0 request or notification.

Compilation flags:
    static

Template:
    method(Message,Method)
Mode and number of proofs:
    method(+compound,--atom) - zero_or_one

---

params/2

Extracts the params field from a JSON-RPC 2.0 request or notification.

Compilation flags:
    static

Template:
    params(Message,Params)
Mode and number of proofs:
    params(+compound,--nonvar) - zero_or_one

---

result/2

Extracts the result field from a JSON-RPC 2.0 response.

Compilation flags:
    static

Template:
    result(Message,Result)
Mode and number of proofs:
    result(+compound,--nonvar) - zero_or_one

---

error/2

Extracts the error field from a JSON-RPC 2.0 error response.

Compilation flags:
    static

Template:
    error(Message,Error)
Mode and number of proofs:
    error(+compound,--compound) - zero_or_one

---

error_code/2

Extracts the error code from a JSON-RPC 2.0 error response.

Compilation flags:
    static

Template:
    error_code(Message,Code)
Mode and number of proofs:
    error_code(+compound,--integer) - zero_or_one

---

error_message/2

Extracts the error message from a JSON-RPC 2.0 error response.

Compilation flags:
    static

Template:
    error_message(Message,ErrorMessage)
Mode and number of proofs:
    error_message(+compound,--atom) - zero_or_one

---

error_data/2

Extracts the error data from a JSON-RPC 2.0 error response. Fails if no data field is present.

Compilation flags:
    static

Template:
    error_data(Message,Data)
Mode and number of proofs:
    error_data(+compound,--nonvar) - zero_or_one

---

write_message/2

Writes a JSON-RPC 2.0 message to an output stream as a single line of JSON followed by a newline. Flushes the output stream after writing.

Compilation flags:
    static

Template:
    write_message(Output,Message)
Mode and number of proofs:
    write_message(+stream,+compound) - one

---

read_message/2

Reads a JSON-RPC 2.0 message from an input stream. Reads a line of JSON text and parses it. Fails at end of stream.

Compilation flags:
    static

Template:
    read_message(Input,Message)
Mode and number of proofs:
    read_message(+stream,--compound) - zero_or_one

---

write_framed_message/2

Writes a JSON-RPC 2.0 message to an output stream using Content-Length framing (as used by LSP style protocols). The message is preceded by a Content-Length: N\r\n\r\n header where N is the byte length of the JSON body. Flushes the output stream after writing.

Compilation flags:
    static

Template:
    write_framed_message(Output,Message)
Mode and number of proofs:
    write_framed_message(+stream,+compound) - one

---

read_framed_message/2

Reads a JSON-RPC 2.0 message from an input stream using Content-Length framing (as used by LSP style protocols). Reads a Content-Length: N\r\n\r\n header followed by exactly N bytes of JSON body. Fails at end of stream or if the header is missing or malformed.

Compilation flags:
    static

Template:
read_framed_message(Input,Message)
Mode and number of proofs:
read_framed_message(+stream,--compound) - zero_or_one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.118 json_schema

object

1.118.1 json_schema

JSON Schema parser and validator. Uses curly terms for JSON objects, dashes for JSON pairs, and atoms for JSON strings.

Availability:
logtalk_load(json_schema(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-01-29

Compilation flags:
static, context_switching_calls

Extends:
public json_schema(curly,dash,atom)

Remarks:

(none)

Inherited public predicates:
parse/2 validate/2 validate/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.118.2 json_schema(StringRepresentation)

- StringRepresentation - String representation used for JSON strings. Possible values are atom (default), chars, and codes.

JSON Schema parser and validator. Uses curly terms for JSON objects and dashes for JSON pairs.

Availability:
logtalk_load(json_schema(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-01-29

Compilation flags:
static, context_switching_calls

Extends:
public json_schema(curly,dash,StringRepresentation)

Remarks:
(none)

Inherited public predicates:
parse/2 validate/2 validate/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.118.3 json_schema(ObjectRepresentation,PairRepresentation,StringRepresentation)

- ObjectRepresentation - Object representation used for JSON objects. Possible values are curly (default) and list.
- PairRepresentation - Pair representation used for JSON object pairs. Possible values are dash (default), equal, and colon.
- StringRepresentation - String representation used for JSON strings. Possible values are atom (default), chars, and codes.

JSON Schema parser and validator supporting JSON Schema draft-07/draft-2019-09/draft-2020-12.

Availability:
logtalk_load(json_schema(loader))

Author: Paulo Moura
Version: 1:3:0
Date: 2026-05-11

Compilation flags:
static, context_switching_calls

Implements:
public json_schema_protocol
Uses:
json(ObjectRepresentation,PairRepresentation,StringRepresentation)
json_pointer(StringRepresentation)
list
os
url(Representation)

Remarks:
(none)

Inherited public predicates:
parse/2 validate/2 validate/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.118.4 json_schema_protocol

JSON Schema parser and validator protocol.

Availability:
logtalk_load(json_schema(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-01-29

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - parse/2
  - validate/2
  - validate/3
- Protected predicates
- Private predicates
- Operators

Public predicates

parse/2

Parses a JSON schema from the given source (file(Path), stream(Stream), codes(List), chars(List), or atom(Atom)) into a schema term.

Compilation flags:
    static

Template:
    parse(Source,Schema)
Mode and number of proofs:
    parse(++compound,--term) - one_or_error

---

validate/2

Validates a JSON term against a parsed schema. Succeeds if the JSON term is valid according to the schema.

Compilation flags:
    static

Template:
    validate(Schema,JSON)
Mode and number of proofs:
    validate(+term,+term) - zero_or_one

---

validate/3

Validates a JSON term against a parsed schema. Returns a list of validation errors (empty list if valid).

Compilation flags:
    static

Template:
    validate(Schema,JSON,Errors)
Mode and number of proofs:
    validate(+term,+term,--list) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.119 kcenters_clusterer

object

1.119.1 kcenters_clusterer

k-Centers clusterer for continuous datasets. Learns from a dataset object implementing the clustering_dataset_protocol protocol and returns a clusterer term that can be used for assigning new instances to clusters and exported as predicate clauses.

Availability:
    logtalk_load(kcenters_clusterer(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public clusterer_common
Uses:
format
list
numberlist
pairs
type

Remarks:
(none)

Inherited public predicates:
check_clusterer/1 check_option/1 check_options/1 cluster/3 clusterer_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 print_clusterer/1 valid_clusterer/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

clusterer_protocol, clustering_dataset_protocol, kmeans_clusterer, kmedoids_clusterer, kmedians_clusterer

1.120 kemeny_young_ranker

object

1.120.1 kemeny_young_ranker

Kemeny-Young pairwise preference ranker. Learns one deterministic consensus order from a dataset object implementing the pairwise_ranking_dataset_protocol protocol by maximizing the total agreeing pairwise preference weight over all linear orders and returns a self-describing ranker term with diagnostics that can be used for ranking and export.

Availability:
logtalk_load(kemeny_young_ranker(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public ranking_dataset_common
public score_ranker_model_common
public condorcet_victory_common
Uses:
avltree
list

Remarks:

(none)

Inherited public predicates:

check_option/1 check_options/1 check_ranker/1 default_option/1 default_options/1
diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 grouped_dataset_groups/2
grouped_dataset_items/2 grouped_dataset_items/3 grouped_dataset_summary/2 learn/2
learn/3 option/2 option/3 pairwise_dataset_connected_components/2
pairwise_dataset_declared_items/2 pairwise_dataset_items/2 pairwise_dataset_matchups/2
pairwise_dataset_preferences/2 pairwise_dataset_summary/2 pairwise_dataset_win_totals/2
pairwise_measurement_dataset_connected_components/2
pairwise_measurement_dataset_declared_items/2 pairwise_measurement_dataset_items/2
pairwise_measurement_dataset_measurements/2 pairwise_measurement_dataset_summary/2
print_ranker/1 rank/3 ranker_options/2 scores/2
temporal_pairwise_dataset_connected_components/2
temporal_pairwise_dataset_declared_items/2 temporal_pairwise_dataset_games/2
temporal_pairwise_dataset_games/3 temporal_pairwise_dataset_items/2
temporal_pairwise_dataset_periods/2 temporal_pairwise_dataset_summary/2 valid_option/1
valid_options/1 valid_ranker/1 validate_grouped_dataset/1 validate_grouped_dataset/2
validate_pairwise_dataset/1 validate_pairwise_dataset/2
validate_pairwise_measurement_dataset/1 validate_pairwise_measurement_dataset/2
validate_temporal_pairwise_dataset/1 validate_temporal_pairwise_dataset/2

- Public predicates
  - consensus_ranking/2
  - consensus_score/2
- Protected predicates
- Private predicates
- Operators

Public predicates

consensus_ranking/2

Returns the learned full-item consensus ranking selected by the Kemeny-Young optimization.

Compilation flags:
static

Template:
consensus_ranking(Ranker,ConsensusRanking)
Mode and number of proofs:
consensus_ranking(+compound,-list) - one

---

consensus_score/2

Returns the maximum total pairwise agreement score of the learned consensus ranking.

Compilation flags:
    static

Template:
    consensus_score(Ranker,ConsensusScore)
Mode and number of proofs:
    consensus_score(+compound,-number) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

pairwise_ranking_dataset_protocol, ranker_protocol, ranked_pairs, schulze_ranker

1.121 kernel_pca_projection

object

1.121.1 kernel_pca_projection

Kernel Principal Component Analysis reducer for continuous datasets using a portable power-iteration eigensolver over centered kernel Gram matrices.

Availability:
logtalk_load(kernel_pca_projection(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public dimension_reducer_common
Uses:
format
linear_algebra
list
numberlist
pairs
type

Remarks:
(none)

Inherited public predicates:
check_dimension_reducer/1 check_option/1 check_options/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 dimension_reducer_options/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 print_dimension_reducer/1 transform/3 valid_dimension_reducer/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

pca_projection, random_projection

1.122 kernel_svm_classifier

object

1.122.1 kernel_svm_classifier

Kernel support vector machine classifier using one-vs-rest dual margin models with linear, polynomial, and radial basis function kernels.

Availability:
logtalk_load(kernel_svm_classifier(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-11

Compilation flags:
static, context_switching_calls

Imports:
public probabilistic_classifier_common
Uses:
format

linear_algebra
list
numberlist
pairs
type

Remarks:
(none)

Inherited public predicates:
check_classifier/1 check_option/1 check_options/1 classifier_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 predict_probabilities/3 print_classifier/1 valid_classifier/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

dataset_protocol, linear_svm_classifier, logistic_regression_classifier, kernel_pca_projection

1.123 kmeans_clusterer

object

1.123.1 kmeans_clusterer

k-Means clusterer for continuous datasets. Learns from a dataset object implementing the clustering_dataset_protocol protocol and returns a clusterer term that can be used for assigning new instances to clusters and exported as predicate clauses.

Availability:
logtalk_load(kmeans_clusterer(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public clusterer_common
Uses:
format
list
numberlist
pairs
type

Remarks:
(none)

Inherited public predicates:
check_clusterer/1 check_option/1 check_options/1 cluster/3 clusterer_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 print_clusterer/1 valid_clusterer/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates

- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

clusterer_protocol, clustering_dataset_protocol, nearest_centroid_classifier

1.124 kmedians_clusterer

object

1.124.1 kmedians_clusterer

k-Medians clusterer for continuous datasets. Learns from a dataset object implementing the clustering_dataset_protocol protocol and returns a clusterer term that can be used for assigning new instances to clusters and exported as predicate clauses.

Availability:
logtalk_load(kmedians_clusterer(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-07

Compilation flags:

static, context_switching_calls

Imports:
public clusterer_common
Uses:
format
list
numberlist
pairs
type

Remarks:
(none)

Inherited public predicates:
check_clusterer/1 check_option/1 check_options/1 cluster/3 clusterer_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 print_clusterer/1 valid_clusterer/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

clusterer_protocol, clustering_dataset_protocol, kmeans_clusterer, kmedoids_clusterer

1.125 kmedoids_clusterer

object

1.125.1 kmedoids_clusterer

k-Medoids clusterer for continuous datasets. Learns from a dataset object implementing the clustering_dataset_protocol protocol and returns a clusterer term that can be used for assigning new instances to clusters and exported as predicate clauses.

Availability:
logtalk_load(kmedoids_clusterer(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public clusterer_common
Uses:
format
list
numberlist
pairs
type

Remarks:

(none)

Inherited public predicates:

check_clusterer/1 check_option/1 check_options/1 cluster/3 clusterer_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 print_clusterer/1 valid_clusterer/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

clusterer_protocol, clustering_dataset_protocol, kmeans_clusterer

1.126 kmodes_clusterer

object

1.126.1 kmodes_clusterer

k-Modes clusterer for discrete datasets. Learns from a dataset object implementing the clustering_dataset_protocol protocol and returns a clusterer term that can be used for assigning new instances to clusters and exported as predicate clauses.

Availability:
logtalk_load(kmodes_clusterer(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public clusterer_common
Uses:
format
list
pairs
type

Remarks:
(none)

Inherited public predicates:
check_clusterer/1 check_option/1 check_options/1 cluster/3 clusterer_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 print_clusterer/1 valid_clusterer/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

clusterer_protocol, clustering_dataset_protocol, kprototypes_clusterer

1.127 knn_classifier

object

1.127.1 knn_classifier

k-Nearest Neighbors classifier with multiple distance metrics and weighting options. Learns from a dataset object implementing the dataset_protocol protocol and returns a classifier term that can be used for prediction and exported as predicate clauses.

Availability:
logtalk_load(knn_classifier(loader))

Author: Paulo Moura
Version: 2:0:0
Date: 2026-05-11

Compilation flags:
static, context_switching_calls

Imports:
public probabilistic_classifier_common
Uses:

format
list
pairs
type

Remarks:
(none)

Inherited public predicates:
check_classifier/1 check_option/1 check_options/1 classifier_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 predict_probabilities/3 print_classifier/1 valid_classifier/1 valid_option/1 valid_options/1

- Public predicates
  - predict/4
  - predict_probabilities/4
- Protected predicates
- Private predicates
- Operators

Public predicates

predict/4

Predicts the class label for a new instance using the learned classifier and the given options. The instance is a list of Attribute-Value pairs.

Compilation flags:
static

Template:
predict(Classifier,Instance,Class,Options)
Mode and number of proofs:
predict(+compound,+list,-atom,+list(compound)) - one

predict_probabilities/4

Predicts class probabilities for a new instance using the learned classifier and the given options. Returns a list of Class-Probability pairs. The instance is a list of Attribute-Value pairs.

Compilation flags:
    static

Template:
    predict_probabilities(Classifier,Instance,Probabilities,Options)
Mode and number of proofs:
    predict_probabilities(+compound,+list,-list,+list(compound)) - one

---

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

dataset_protocol, c45_classifier, isolation_forest_anomaly_detector, naive_bayes_classifier, nearest_centroid_classifier, random_forest_classifier, adaptive_boosting_classifier

1.128 knn_distance_anomaly_detector

object

1.128.1 knn_distance_anomaly_detector

k-nearest-neighbor distance anomaly detector with multiple distance metrics, mixed-feature support, and missing-value handling. Learns from baseline training examples selected from a dataset object implementing the anomaly_dataset_protocol protocol and returns a detector term that can be used for scoring, prediction, and export.

Availability:
logtalk_load(knn_distance_anomaly_detector(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public anomaly_detector_common
Uses:
format
list
numberlist
pairs
type

Remarks:
(none)

Inherited public predicates:
anomaly_detector_options/2 check_anomaly_detector/1 check_option/1 check_options/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 predict/4 print_anomaly_detector/1 score/3 score_all/3 valid_anomaly_detector/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

anomaly_dataset_protocol, anomaly_detector_protocol, isolation_forest_anomaly_detector, lof_anomaly_detector

1.129 knn_regression

object

1.129.1 knn_regression

k-Nearest Neighbors regressor with multiple distance metrics, weighting schemes, optional feature scaling, and mixed-feature support.

Availability:
logtalk_load(knn_regression(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public regressor_common
Uses:

format
list
numberlist
type

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_regressor/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 print_regressor/1 regressor_options/2 valid_option/1 valid_options/1 valid_regressor/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

linear_regression, regression_tree, random_forest_regression, gradient_boosting_regression

1.130 kprototypes_clusterer

object

1.130.1 kprototypes_clusterer

k-Prototypes clusterer for mixed datasets with continuous and discrete attributes. Learns from a dataset object implementing the clustering_dataset_protocol protocol and returns a clusterer term that can be used for assigning new instances to clusters and exported as predicate clauses.

Availability:
logtalk_load(kprototypes_clusterer(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public clusterer_common
Uses:
format
list
pairs
population
type

Remarks:
(none)

Inherited public predicates:
check_clusterer/1 check_option/1 check_options/1 cluster/3 clusterer_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 print_clusterer/1 valid_clusterer/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates

- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

clusterer_protocol, clustering_dataset_protocol, kmeans_clusterer, kmedoids_clusterer

1.131 ksuid

object

1.131.1 ksuid

KSUID generator using atom representation and the canonical Base62 alphabet.

Availability:
logtalk_load(ksuid(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-26

Compilation flags:
static, context_switching_calls

Extends:
public
ksuid(atom,0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz)

Remarks:
(none)

Inherited public predicates:
generate/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

ksuid(Representation,Alphabet), cuid2, nanoid, ids, ulid, snowflakeid, uuid

object

1.131.2 ksuid(Representation,Alphabet)

- Representation - Text representation for the KSUID. Possible values are atom, chars, and codes.
- Alphabet - Base62 alphabet used for encoding KSUIDs represented as an atom, list of characters, or list of character codes.

KSUID generator.

Availability:
logtalk_load(ksuid(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-26

Compilation flags:
static, context_switching_calls

Implements:
public ksuid_protocol
Uses:
fast_random(Algorithm)
iso8601
list
os

Remarks:
(none)

Inherited public predicates:
generate/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

ksuid, cuid2(Representation,Size,Alphabet), nanoid(Representation,Size,Alphabet), ids(Representation,Bytes), ulid(Representation), snowflakeid(Representation,EpochMilliseconds,TimeUnitMilliseconds,Tim uuid(Representation)

protocol

1.131.3 ksuid_protocol

KSUID generator protocol.

Availability:
logtalk_load(ksuid(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-26

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - generate/1
- Protected predicates
- Private predicates
- Operators

Public predicates

generate/1

Returns a KSUID.

Compilation flags:
static

Template:
generate(KSUID)
Mode and number of proofs:
generate(--ground) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.132 lasso_regression

object

1.132.1 lasso_regression

Lasso regression regressor supporting continuous and mixed-feature datasets using coordinate descent with coefficient-wise L1 shrinkage. Learns from a dataset object implementing the regression_dataset_protocol protocol and returns a regressor term that can be used for prediction and exported as predicate clauses.

Availability:
logtalk_load(lasso_regression(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:
static, context_switching_calls

Imports:
public regressor_common
Uses:
format
list
numberlist
type

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_regressor/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 print_regressor/1 regressor_options/2 valid_option/1 valid_options/1 valid_regressor/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

linear_regression, ridge_regression, knn_regression, regression_tree, random_forest_regression, gradient_boosting_regression

1.133 lda_classifier

object

1.133.1 lda_classifier

Linear Discriminant Analysis classifier for continuous datasets using a pooled regularized covariance model.

Availability:
logtalk_load(lda_classifier(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-11

Compilation flags:
static, context_switching_calls

Imports:
public classifier_common
Uses:
format
linear_algebra
list
pairs
type

Remarks:
(none)

Inherited public predicates:
check_classifier/1 check_option/1 check_options/1 classifier_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 print_classifier/1 valid_classifier/1 valid_option/1 valid_options/1

- Public predicates
  - predict_scores/3
- Protected predicates
- Private predicates
- Operators

Public predicates

predict_scores/3

Predicts class discriminant scores for a new instance.

Compilation flags:
static

Template:
predict_scores(Classifier,Instance,Scores)
Mode and number of proofs:
predict_scores(+compound,+list,-list) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

dataset_protocol, qda_classifier, nearest_centroid_classifier, lda_projection

1.134 lda_projection

object

1.134.1 lda_projection

Linear Discriminant Analysis projection for labeled continuous datasets using a portable regularized Fisher eigensolver.

Availability:
logtalk_load(lda_projection(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public dimension_reducer_common
Uses:
format
linear_algebra

list
pairs
type

Remarks:
(none)

Inherited public predicates:
check_dimension_reducer/1 check_option/1 check_options/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 dimension_reducer_options/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 print_dimension_reducer/1 transform/3 valid_dimension_reducer/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

pca_projection, random_projection

1.135 lgtdoc

object

1.135.1 lgtdoc

Documenting tool. Generates XML documenting files for loaded entities and for library, directory, entity, and predicate indexes.

Availability:
logtalk_load(lgtdoc(loader))

Author: Paulo Moura
Version: 11:3:0
Date: 2026-03-31

Compilation flags:
static, context_switching_calls

Implements:
public lgtdocp
Imports:
public tool_diagnostics_common
public options
Uses:
date
list
logtalk
os
term_io
type
user
varlist

Remarks:
(none)

Inherited public predicates:
all/0 all/1 check_option/1 check_options/1 context_summaries/2 default_option/1 default_options/1 diagnostic/2 diagnostic/3 diagnostic_rule/5 diagnostic_rules/1 diagnostic_target/1 diagnostics/2 diagnostics/3 diagnostics_breakdown/2 diagnostics_preflight/2 diagnostics_preflight/3 diagnostics_summary/2 diagnostics_summary/3 diagnostics_tool/5 directories/1 directories/2 directory/1 directory/2 file/1 file/2 files/1 files/2 fix_option/2

fix_options/2 libraries/1 libraries/2 library/1 library/2 merge_options/2 option/2 option/3 rdirectories/1 rdirectories/2 rdirectory/1 rdirectory/2 rlibraries/1 rlibraries/2 rlibrary/1 rlibrary/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
  - library_entity_/4
  - directory_entity_/4
  - type_entity_/4
  - predicate_entity_/4
  - cached_run_/5
  - active_diagnostic_/1
  - active_preflight_issue_/1
  - active_collection_mode_/1
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

library_entity_/4

Table of documented entities per library.

Compilation flags:
dynamic

Template:
library_entity_(Library,PrimarySortKey,SecondarySortKey,Entity)
Mode and number of proofs:
library_entity_(?atom,?nonvar,?nonvar,?atom) - zero_or_more

---

directory_entity_/4

Table of documented entities per directory.

Compilation flags:
    dynamic

Template:
    directory_entity_(Directory,PrimarySortKey,SecondarySortKey,Entity)
Mode and number of proofs:
    directory_entity_(?atom,?nonvar,?nonvar,?atom) - zero_or_more

---

type_entity_/4

Table of documented entities per type.

Compilation flags:
    dynamic

Template:
    type_entity_(Type,PrimarySortKey,SecondarySortKey,Entity)
Mode and number of proofs:
    type_entity_(?atom,?nonvar,?nonvar,?atom) - zero_or_more

---

predicate_entity_/4

Table of public predicates for all documented entities.

Compilation flags:
    dynamic

Template:
    predicate_entity_(Predicate,PrimarySortKey,SecondarySortKey,Entity)
Mode and number of proofs:

predicate_entity_(?predicate_indicator,?nonvar,?nonvar,?entity_identifier) - zero_or_more

---

cached_run_/5

Cache of diagnostics collected while generating documentation for a target and merged options.

Compilation flags:
dynamic

Template:
cached_run_(Target,Options,Contexts,Diagnostics,PreflightIssues)
Mode and number of proofs:
cached_run_(?nonvar,?list(compound),?list(compound),?list(compound),?list(compound)) - zero_or_more

---

active_diagnostic_/1

Diagnostics collected during the current documentation run.

Compilation flags:
dynamic

Template:
active_diagnostic_(Diagnostic)
Mode and number of proofs:
active_diagnostic_(?compound) - zero_or_more

---

active_preflight_issue_/1

Preflight issues collected during the current documentation run.

Compilation flags:
dynamic

Template:
active_preflight_issue_(Issue)
Mode and number of proofs:
active_preflight_issue_(?compound) - zero_or_more

---

active_collection_mode_/1

Current diagnostics collection mode.

Compilation flags:
dynamic

Template:
active_collection_mode_(Mode)
Mode and number of proofs:
active_collection_mode_(?atom) - zero_or_one

Operators

(none)

category

1.135.2 lgtdoc_messages

Logtalk documentation tool default message translations.

Availability:
logtalk_load(lgtdoc(loader))

Author: Paulo Moura
Version: 4:0:1
Date: 2024-12-02

Compilation flags:
static

Provides:

logtalk::message_prefix_stream/4
logtalk::message_tokens//2

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.135.3 lgtdocp

Documenting tool protocol.

Availability:
logtalk_load(lgtdoc(loader))

Author: Paulo Moura
Version: 6:0:0

Date: 2024-03-08

Compilation flags:
static

Dependencies:
(none)

Remarks:

- Compiling files for generating XML documentation: All source files must be compiled with the source_data flag turned on.
- xml_spec(Specification) option: XML documenting files specification format. Possible option values are dtd (DTD specification; default) and xsd (XML Schema specification).
- xml_spec_reference(Reference) option: Reference to the XML specification file in XML documenting files. Possible values are local (default; DTD/XSD file in same folder as XML files), web (logtalk.org website DTD/XSD file), and standalone (no reference to specification files).
- entity_xsl_file(File) option: XSLT file to use with generated XML documenting files. Default is logtalk_entity_to_xml.xsl, allowing the XML files to be viewed by opening them with a browser supporting XSLT (after running the lgt2xml.sh script on the output directory).
- index_xsl_file(File) option: XSLT file to use with generated XML documenting files. Default is logtalk_index_to_xml.xsl, allowing the XML files to be viewed by opening them with a browser supporting XSLT (after running the lgt2xml.sh script on the output directory).
- xml_docs_directory(Directory) option: Directory where the XML documenting files will be generated. The default value is ./xml_docs, a sub-directory of the source files directory.
- bom(Boolean) option: Defines if a BOM should be added to the generated XML documenting files.
- encoding(Encoding) option: Encoding to be used for the generated XML documenting files.
- omit_path_prefixes(Prefixes) option: List of path prefixes (atoms) to omit when writing directory paths. The default value is to omit the home directory.
- exclude_files(List) option: List of files to exclude when generating the XML documenting files.
- exclude_paths(List) option: List of relative library paths to exclude when generating the XML documenting files (default is []). All sub-directories of the excluded directories are also excluded.
- exclude_prefixes(List) option: List of path prefixes to exclude when generating the XML documenting files (default is []).
- exclude_entities(List) option: List of entities to exclude when generating the XML documenting files (default is []).
- sort_predicates(Boolean) option: Sort entity predicates (default is false).
- Known issues: Some options may depend on the used XSL processor. Most XSL processors support DTDs but only some of them support XML Schemas. Some processors (e.g., fop2) reject reference to a DTD.

Inherited public predicates:

(none)

- Public predicates
  - rlibraries/2
  - rlibraries/1
  - rlibrary/2
  - rlibrary/1
  - libraries/2
  - libraries/1
  - library/2
  - library/1
  - rdirectories/2
  - rdirectories/1
  - rdirectory/2
  - rdirectory/1
  - directories/2
  - directories/1
  - directory/2
  - directory/1
  - files/2
  - files/1
  - file/2
  - file/1
  - all/1
  - all/0
- Protected predicates
- Private predicates
- Operators

Public predicates

rlibraries/2

Creates XML documenting files for all entities in all given libraries and their sub-libraries using the specified options.

Compilation flags:
    static

Template:
    rlibraries(Libraries,Options)
Mode and number of proofs:
    rlibraries(+list(atom),+list) - one

---

rlibraries/1

Creates XML documenting files for all entities in all given libraries and their sub-libraries using default options.

Compilation flags:
    static

Template:
    rlibraries(Libraries)
Mode and number of proofs:
    rlibraries(+list(atom)) - one

---

rlibrary/2

Creates XML documenting files for all entities in a library and its sub-libraries using the specified options.

Compilation flags:
    static

Template:
    rlibrary(Library,Options)

Mode and number of proofs:
rlibrary(+atom,+list) - one

Examples:
Generate XML documenting files for all tool entities for later conversion to Markdown files
rlibrary(tools,[xslfile('lgtmd.xsl')])
yes

---

rlibrary/1

Creates XML documenting files for all entities in a library and its sub-libraries using default options.

Compilation flags:
static

Template:
rlibrary(Library)
Mode and number of proofs:
rlibrary(+atom) - one

Examples:
Generate XML documenting files for all tool entities for direct viewing in a browser (after indexing using the lgt2xml script)
rlibrary(tools)
yes

---

libraries/2

Creates XML documenting files for all entities in all given libraries using the specified options.

Compilation flags:
static

Template:
libraries(Libraries,Options)
Mode and number of proofs:
libraries(+list(atom),+list) - one

---

libraries/1

Creates XML documenting files for all entities in all given libraries using default options.

Compilation flags:
    static

Template:
    libraries(Libraries)
Mode and number of proofs:
    libraries(+list(atom)) - one

---

library/2

Creates XML documenting files for all entities in a library using the specified options.

Compilation flags:
    static

Template:
    library(Library,Options)
Mode and number of proofs:
    library(+atom,+list) - one

Examples:
    Generate XML documenting files for all library entities for later conversion to PDF A4 files
        library(library,[xslfile('logtalk_entity_to_pdf_a4.xsl')])
        yes

---

library/1

Creates XML documenting files for all entities in a library using default options.

Compilation flags:
    static

Template:
    library(Library)
Mode and number of proofs:
    library(+atom) - one

---

rdirectories/2

Creates XML documenting files for all entities in all given directories and their sub-directories using the specified options.

Compilation flags:
    static

Template:
    rdirectories(Directories,Options)
Mode and number of proofs:
    rdirectories(+list(atom),+list) - one

---

rdirectories/1

Creates XML documenting files for all entities in all given directories and their sub-directories using default options.

Compilation flags:
    static

Template:
    rdirectories(Directories)
Mode and number of proofs:
    rdirectories(+list(atom)) - one

---

rdirectory/2

Creates XML documenting files for all entities in a directory and its sub-directories using the specified options.

Compilation flags:
    static

Template:
    rdirectory(Directory,Options)
Mode and number of proofs:
    rdirectory(+atom,+list) - one

Examples:
    Generate XML documenting files for all entities in the tools directory for later conversion to Markdown files
        rdirectory('./tools',[xslfile('lgtmd.xsl')])
        yes

---

rdirectory/1

Creates XML documenting files for all entities in a directory and its sub-directories using default options.

Compilation flags:
    static

Template:
    rdirectory(Directory)
Mode and number of proofs:
    rdirectory(+atom) - one

Examples:
    Generate XML documenting files for all entities in the tools directory for direct viewing in a browser (after indexing using the lgt2xml script)
        rdirectory('./tools')
        yes

---

directories/2

Creates XML documenting files for all entities in all given directories using the specified options.

Compilation flags:
    static

Template:
    directories(Directories,Options)
Mode and number of proofs:
    directories(+list(atom),+list) - one

---

directories/1

Creates XML documenting files for all entities in all given directories using default options.

Compilation flags:
    static

Template:
    directories(Directories)
Mode and number of proofs:
    directories(+list(atom)) - one

---

directory/2

Creates XML documenting files for all entities in a directory using the specified options.

Compilation flags:
    static

Template:
    directory(Directory,Options)
Mode and number of proofs:
    directory(+atom,+list) - one

Examples:

Generate XML documenting files for all the entities in the current directory for later conversion to PDF A4 files

```
directory('.',[xslfile('logtalk_entity_to_pdf_a4.xsl')])
yes
```

---

directory/1

Creates XML documenting files for all entities in a directory using default options.

Compilation flags:

static

Template:

directory(Directory)

Mode and number of proofs:

directory(+atom) - one

---

files/2

Creates XML documenting files for all entities in loaded source files using the specified options. The files can be given by name, basename, full path, or using library notation.

Compilation flags:

static

Template:

files(Files,Options)

Mode and number of proofs:

files(+list(atom),+list) - one

---

files/1

Creates XML documenting files for all entities in loaded source files using default options. The files can be given by name, basename, full path, or using library notation.

Compilation flags:
    static

Template:
    files(Files)
Mode and number of proofs:
    files(+list(atom)) - one

---

file/2

Creates XML documenting files for all entities in a loaded source file using the specified options. The file can be given by name, basename, full path, or using library notation.

Compilation flags:
    static

Template:
    file(File,Options)
Mode and number of proofs:
    file(+atom,+list) - one

---

file/1

Creates XML documenting files for all entities in a loaded source file using default options. The file can be given by name, basename, full path, or using library notation.

Compilation flags:
    static

Template:
    file(File)
Mode and number of proofs:

file(+atom) - one

---

all/1

Creates XML documenting files for all loaded entities using the specified options.

Compilation flags:
static

Template:
all(Options)
Mode and number of proofs:
all(+list) - one

---

all/0

Creates XML documenting files for all loaded entities using default options.

Compilation flags:
static

Mode and number of proofs:
all - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

lgtdoc

1.136 lgtunit

object

1.136.1 automation_report

Intercepts unit test execution messages and generates a *.totals files for parsing by the logtalk_tester.sh automation shell script.

Availability:
logtalk_load(lgtunit(loader))

Author: Paulo Moura
Version: 6:0:1
Date: 2026-01-25

Compilation flags:
static, context_switching_calls

Provides:
logtalk::message_hook/4
Uses:
user

Remarks:

- Usage: Automatically loaded by the logtalk_tester.sh shell script.

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.136.2 cobertura_report

Intercepts unit test execution messages and generates a cobertura.xml file with code coverage results.

Availability:
    logtalk_load(lgtunit(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-13

Compilation flags:
    static, context_switching_calls

Provides:
    logtalk::message_hook/4
Uses:
    logtalk

Remarks:

- Usage: Simply load this object before running your tests using the goal logtalk_load(lgtunit(cobertura_report)).

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
  - timestamp_/6
  - class_/5
  - method_/7
  - lines_covered_/1
  - lines_total_/1
  - summary_/2
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

timestamp_/6

Test start date and time.

Compilation flags:
dynamic

Template:
timestamp_(Year,Month,Day,Hours,Minutes,Seconds)
Mode and number of proofs:

timestamp_(?integer,?integer,?integer,?integer,?integer,?integer) - zero_or_one

---

class_/5

Table of entity coverage data.

Compilation flags:
dynamic

Template:
class_(Entity,File,Line,Covered,Total)
Mode and number of proofs:
class_(?entity_identifier,?atom,?integer,?integer,?integer) - zero_or_more

---

method_/7

Table of predicate coverage data.

Compilation flags:
dynamic

Template:
method_(Entity,Predicate,Line,Covered,Total,Percentage,Clauses)
Mode and number of proofs:
method_(?entity_identifier,?predicate_indicator,?integer,?integer,?integer,?float,?list) - zero_or_more

---

lines_covered_/1

Counter for total covered lines.

Compilation flags:
dynamic

Template:
lines_covered_(Covered)
Mode and number of proofs:
lines_covered_(?integer) - zero_or_one

---

lines_total_/1

Counter for total lines.

Compilation flags:
dynamic

Template:
lines_total_(Total)
Mode and number of proofs:
lines_total_(?integer) - zero_or_one

---

summary_/2

Coverage summary with covered and total entity counts.

Compilation flags:
dynamic

Template:
summary_(Covered,Total)
Mode and number of proofs:
summary_(?integer,?integer) - zero_or_one

Operators

(none)

object

1.136.3 coverage_report

Intercepts unit test execution messages and generates a coverage_report.xml file with a test suite code coverage results.

Availability:
logtalk_load(lgtunit(loader))

Author: Paulo Moura
Version: 3:2:1
Date: 2026-02-06

Compilation flags:
static, context_switching_calls

Provides:
logtalk::message_hook/4
Uses:
logtalk
user

Remarks:

- Usage: Simply load this object before running your tests using the goal logtalk_load(lgtunit(coverage_report)).

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
  - timestamp_/6
  - object_file_/2

- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

timestamp_/6

Cache of the starting tests timestamp.

Compilation flags:
    dynamic

Template:
    timestamp_(Year,Month,Day,Hours,Minutes,Seconds)
Mode and number of proofs:
    timestamp_(-integer,-integer,-integer,-integer,-integer,-integer) - one

---

object_file_/2

Cache of test object - file pairs.

Compilation flags:
    dynamic

Template:
    object_file_(Object,File)
Mode and number of proofs:
    object_file_(?object_identifier,?atom) - zero_or_more

Operators

(none)

object

1.136.4 ctrf_output

Intercepts unit test execution messages and outputs a report using the CTRF JSON format to the current output stream.

Availability:
logtalk_load(lgtunit(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-27

Compilation flags:
static, context_switching_calls

Provides:
logtalk::message_hook/4

Remarks:

- Usage: Simply load this object before running your tests using the goal logtalk_load(lgtunit(ctrf_output)).

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
  - message_cache_/1
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

message_cache_/1

Table of messages emitted by the lgtunit tool when running tests.

Compilation flags:
dynamic

Template:
message_cache_(Message)
Mode and number of proofs:
message_cache_(?callable) - zero_or_more

Operators

(none)

object

1.136.5 ctrf_report

Intercepts unit test execution messages and generates a ctrf_report.json file using the CTRF JSON format in the same directory as the tests object file.

Availability:
logtalk_load(lgtunit(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-27

Compilation flags:

static, context_switching_calls

Provides:
logtalk::message_hook/4
Uses:
logtalk

Remarks:

- Usage: Simply load this object before running your tests using the goal logtalk_load(lgtunit(ctrf_report)).

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
  - message_cache_/1
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

message_cache_/1

Table of messages emitted by the lgtunit tool when running tests.

Compilation flags:
dynamic

Template:

message_cache_(Message)
Mode and number of proofs:
message_cache_(?callable) - zero_or_more

Operators

(none)

object

1.136.6 lcov_report

Intercepts unit test execution messages and generates a lcov.info file with code coverage results.

Availability:
logtalk_load(lgtunit(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-13

Compilation flags:
static, context_switching_calls

Provides:
logtalk::message_hook/4
Uses:
logtalk

Remarks:

- Usage: Simply load this object before running your tests using the goal logtalk_load(lgtunit(lcov_report)).

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
  - entity_file_/2
  - predicate_coverage_/7
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

entity_file_/2

Table of entity source files.

Compilation flags:
    dynamic

Template:
    entity_file_(Entity,File)
Mode and number of proofs:
    entity_file_(?entity_identifier,?atom) - zero_or_more

---

predicate_coverage_/7

Table of predicate coverage data.

Compilation flags:
    dynamic

Template:

predicate_coverage_(Entity,Predicate,Line,Covered,Total,Percentage,Clauses)

Mode and number of proofs:

predicate_coverage_(?entity_identifier,?predicate_indicator,?integer,?integer,?integer,?float,?list) - zero_or_more

---

Operators

(none)

object

1.136.7 lgtunit

A unit test framework supporting predicate clause coverage, determinism testing, input/output testing, property-based testing, and multiple test dialects.

Availability:

logtalk_load(lgtunit(loader))

Author: Paulo Moura
Version: 22:7:0
Date: 2026-04-01

Compilation flags:

static, context_switching_calls

Implements:

public expanding

Imports:

public tool_diagnostics_common
public lgtunit_messages
public tutor_explanations
public options

Provides:

logtalk::message_hook/4
logtalk::trace_event/2

Uses:

fast_random
list
logtalk
os
term_io

type
user

Remarks:

- Usage: Define test objects as extensions of the lgtunit object and compile their source files using the compiler option hook(lgtunit).
- Portability: Deterministic unit tests are currently not available when using Quintus Prolog as the backend compiler.
- Known issues: Parameter variables cannot currently be used in the definition of test options.

Inherited public predicates:
check_option/1 check_options/1 default_option/1 default_options/1 diagnostic/2 diagnostic/3 diagnostic_rule/5 diagnostic_rules/1 diagnostic_target/1 diagnostics/2 diagnostics/3 diagnostics_preflight/2 diagnostics_preflight/3 diagnostics_summary/2 diagnostics_summary/3 diagnostics_tool/5 explain//1 failed_test_reason//1 goal_expansion/2 option/2 option/3 term_expansion/2 valid_option/1 valid_options/1

- Public predicates
  - cover/1
  - run/0
  - run/1
  - run/2
  - run_test_sets/1
  - test/1
  - number_of_tests/1
  - deterministic/1
  - deterministic/2
  - assertion/1
  - assertion/2
  - quick_check/3
  - quick_check/2
  - quick_check/1
  - benchmark/2
  - benchmark_reified/3
  - benchmark/3
  - benchmark/4

  - variant/2
  - approximately_equal/2
  - approximately_equal/3
  - essentially_equal/3
  - tolerance_equal/4
  - =~= / 2
  - epsilon/1
- Protected predicates
  - run_tests/0
  - run_tests/1
  - run_test_set/0
  - run_quick_check_tests/5
  - condition/0
  - setup/0
  - cleanup/0
  - make/1
  - note/1
  - file_path/2
  - file_url/2
  - suppress_text_output/0
  - suppress_binary_output/0
  - set_text_input/3
  - set_text_input/2
  - set_text_input/1
  - check_text_input/2
  - check_text_input/1
  - text_input_assertion/3
  - text_input_assertion/2
  - clean_text_input/0
  - set_binary_input/3
  - set_binary_input/2
  - set_binary_input/1
  - check_binary_input/2
  - check_binary_input/1
  - binary_input_assertion/3

- binary_input_assertion/2
- clean_binary_input/0
- set_text_output/3
- set_text_output/2
- set_text_output/1
- check_text_output/3
- check_text_output/2
- check_text_output/1
- text_output_assertion/4
- text_output_assertion/3
- text_output_assertion/2
- text_output_contents/3
- text_output_contents/2
- text_output_contents/1
- clean_text_output/0
- set_binary_output/3
- set_binary_output/2
- set_binary_output/1
- check_binary_output/2
- check_binary_output/1
- binary_output_assertion/3
- binary_output_assertion/2
- binary_output_contents/2
- binary_output_contents/1
- clean_binary_output/0
- create_text_file/3
- create_text_file/2
- create_binary_file/2
- check_text_file/3
- check_text_file/2
- text_file_assertion/4
- text_file_assertion/3
- check_binary_file/2
- binary_file_assertion/3
- clean_file/1

  - clean_directory/1
  - closed_input_stream/2
  - closed_output_stream/2
  - stream_position/1
  - test/2
- Private predicates
  - running_test_sets_/0
  - test/3
  - auxiliary_predicate_counter_/1
  - test_/2
  - selected_test_/1
  - linter_warning_sequence_/1
  - recorded_linter_warning_/7
  - skipped_/1
  - passed_/3
  - failed_/3
  - flaky_/1
  - fired_/3
  - covered_/4
- Operators
  - op(700,xfx,=~=)

Public predicates

cover/1

Declares entities being tested for which code coverage information should be collected.

Compilation flags:
    static

Template:
    cover(Entity)
Mode and number of proofs:
    cover(?entity_identifier) - zero_or_more

run/0

Runs the unit tests, writing the results to the current output stream.

Compilation flags:
    static

Mode and number of proofs:
    run - one

---

run/1

Runs a unit test or a list of unit tests, writing the results to the current output stream. Runs the global setup and cleanup steps when defined. Fails when given a partial list of tests or when one of the test identifiers is not valid.

Compilation flags:
    static

Template:
    run(Tests)
Mode and number of proofs:
    run(++callable) - zero_or_one
    run(++list(callable)) - zero_or_one

---

run/2

Runs the unit tests, writing the results to the specified file. Mode can be either write (to create a new file) or append (to add results to an existing file).

Compilation flags:
    static

Template:
    run(File,Mode)
Mode and number of proofs:
    run(+atom,+atom) - one

run_test_sets/1

Runs two or more test sets as a unified set generating a single code coverage report if one is requested. When there is a single test set, it is equivalent to sending the message run/0 to the test set. Trivially succeeds when the argument is an empty list.

Compilation flags:
    static

Template:
    run_test_sets(TestObjects)
Mode and number of proofs:
    run_test_sets(+list(object)) - one

Exceptions:
    TestObjects is a partial list or a list with an element which is a variable:
        instantiation_error
    TestObjects is neither a partial list nor a list:
        type_error(list(object),TestObjects)
    An element TestObject of the TestObjects list is not an existing object:
        existence_error(object,TestObject)

---

test/1

Enumerates, by backtracking, the identifiers of all defined unit tests.

Compilation flags:
    static

Template:
    test(Identifier)
Mode and number of proofs:
    test(?callable) - zero_or_more

---

number_of_tests/1

Number of defined unit tests.

Compilation flags:
    static

Template:
    number_of_tests(NumerOfTests)
Mode and number of proofs:
    number_of_tests(?integer) - zero_or_one

---

deterministic/1

True if the goal succeeds once without leaving choice-points.

Compilation flags:
    static

Template:
    deterministic(Goal)
Meta-predicate template:
    deterministic(0)
Mode and number of proofs:
    deterministic(+callable) - zero_or_one

---

deterministic/2

Reified version of the deterministic/1 predicate. True if the goal succeeds. Returns a boolean value (true or false) indicating if the goal succeeded without leaving choice-points.

Compilation flags:
    static

Template:
    deterministic(Goal,Deterministic)
Meta-predicate template:

deterministic(0,*)
Mode and number of proofs:
deterministic(+callable,--atom) - zero_or_one

---

assertion/1

True if the assertion goal succeeds. Throws an error using the assertion goal as argument if the assertion goal throws an error or fails.

Compilation flags:
static

Template:
assertion(Assertion)
Meta-predicate template:
assertion(::)
Mode and number of proofs:
assertion(@callable) - one

Exceptions:
Assertion goal fails:
assertion_failure(Assertion)
Assertion goal throws Error:
assertion_error(Assertion,Error)

---

assertion/2

True if the assertion goal succeeds. Throws an error using the description as argument if the assertion goal throws an error or fails. The description argument helps to distinguish between different assertions in the same test body.

Compilation flags:
static

Template:
assertion(Description,Assertion)
Meta-predicate template:
assertion(*,0)

Mode and number of proofs:
    assertion(+nonvar,@callable) - one

Exceptions:
    Assertion goal fails:
        assertion_failure(Description)
    Assertion goal throws Error:
        assertion_error(Description,Error)

---

quick_check/3

Reified version of the quick_check/2 predicate. Reports passed(SequenceSeed,Discarded,Labels), failed(Goal,SequenceSeed,TestSeed), error(Error,Goal,SequenceSeed,TestSeed), or broken(Why,Culprit). Goal is the failed test.

Compilation flags:
    static

Template:
    quick_check(Template,Result,Options)
Meta-predicate template:
    quick_check(::,*,::)
Mode and number of proofs:
    quick_check(@callable,-callable,++list(compound)) - one

Remarks:

- SequenceSeed argument: Can be used to re-run the same exact sequence of pseudo-random tests by using the rs/1 option after changes to the code being tested.
- TestSeed argument: Can be used to re-run the test that failed by using the rs/1 option after changes to the code being tested.
- Discarded argument: Number of generated tests that were discarded for failing to comply a pre-condition specified using the pc/1 option.
- Labels argument: List of pairs Label-N where N is the number of generated tests that are classified as Label by a closure specified using the l/1 option.
- broken(Why,Culprit) result: This result signals a broken setup. For example, an invalid template, a broken pre-condition or label goal, or broken test generation.

---

quick_check/2

Generates and runs random tests for a predicate given its mode template and a set of options. Fails when a generated test fails printing the test. Also fails on an invalid option, printing the option.

Compilation flags:
    static

Template:
    quick_check(Template,Options)
Meta-predicate template:
    quick_check(::,::)
Mode and number of proofs:
    quick_check(@callable,++list(compound)) - zero_or_one

Remarks:

- Number of tests: Use the n(NumberOfTests) option to specify the number of random tests. Default is 100.
- Maximum number of shrink operations: Use the s(MaxShrinks) option to specify the number of shrink operations when a counter example is found. Default is 64.
- Type edge cases: Use the ec(Boolean) option to specify if type edge cases are tested (before generating random tests). Default is true.
- Starting seed: Use the rs(Seed) option to specify the random generator starting seed to be used when generating tests. No default. Seeds should be regarded as opaque terms.
- Test generation filtering: Use the pc/1 option to specify a pre-condition closure for filtering generated tests (extended with the test arguments; no default).
- Generated tests classification: Use the l/1 option to specify a label closure for classifying the generated tests (extended with the test arguments plus the labels argument; no default). The labelling predicate can return a single test label or a list of test labels.
- Verbose test generation: Use the v(Boolean) option to specify verbose reporting of generated random tests. Default is false.
- Progress bar: Use the pb(Boolean,Tick) option to print a progress bar for the executed tests, advancing at every Tick tests. Default is false. Only applies when the verbose option is false.

quick_check/1

Generates and runs random tests using default options for a predicate given its mode template. Fails when a generated test fails printing the test.

Compilation flags:
    static

Template:
    quick_check(Template)
Mode and number of proofs:
    quick_check(@callable) - zero_or_one

---

benchmark/2

Benchmarks a goal and returns the total execution time in seconds. Uses CPU clock. Goals that may throw an exception should be wrapped by the catch/3 control construct.

Compilation flags:
    static

Template:
    benchmark(Goal,Time)
Meta-predicate template:
    benchmark(0,*)
Mode and number of proofs:
    benchmark(+callable,-float) - one

---

benchmark_reified/3

Benchmarks a goal and returns the total execution time in seconds plus its result (success, failure, or error(Error)). Uses CPU clock.

Compilation flags:
    static

Template:

benchmark_reified(Goal,Time,Result)
Meta-predicate template:
benchmark_reified(0,*,*)
Mode and number of proofs:
benchmark_reified(+callable,-float,-callable) - one

---

benchmark/3

Benchmarks a goal by repeating it the specified number of times and returning the total execution time in seconds. Uses CPU clock. Goals that may throw an exception should be wrapped by the catch/3 control construct.

Compilation flags:
static

Template:
benchmark(Goal,Repetitions,Time)
Meta-predicate template:
benchmark(0,*,*)
Mode and number of proofs:
benchmark(@callable,+positive_integer,-float) - one

---

benchmark/4

Benchmarks a goal by repeating it the specified number of times and returning the total execution time in seconds using the given clock (cpu or wall). Goals that may throw an exception should be wrapped by the catch/3 control construct.

Compilation flags:
static

Template:
benchmark(Goal,Repetitions,Clock,Time)
Meta-predicate template:
benchmark(0,*,*,*)
Mode and number of proofs:
benchmark(@callable,+positive_integer,+atom,-float) - one

---

variant/2

True when the two arguments are a variant of each other. I.e. if is possible to rename the term variables to make them identical. Useful for checking expected test results that contain variables.

Compilation flags:
    static

Template:
    variant(Term1,Term2)
Mode and number of proofs:
    variant(@term,@term) - zero_or_one

---

approximately_equal/2

Compares two numbers for approximate equality given the epsilon arithmetic constant value using the de facto standard formula abs(Number1 - Number2) =< max(abs(Number1), abs(Number2)) * epsilon. Type-checked.

Compilation flags:
    static

Template:
    approximately_equal(Number1,Number2)
Mode and number of proofs:
    approximately_equal(+number,+number) - zero_or_one

---

approximately_equal/3

Compares two numbers for approximate equality given a user-defined epsilon value using the de facto standard formula abs(Number1 - Number2) =< max(abs(Number1), abs(Number2)) * Epsilon. Type-checked.

Compilation flags:
    static

Template:
    approximately_equal(Number1,Number2,Epsilon)

Mode and number of proofs:
approximately_equal(+number,+number,+number) - zero_or_one

Remarks:

- Epsilon range: Epsilon should be the epsilon arithmetic constant value or a small multiple of it. Only use a larger value if a greater error is expected.
- Comparison with essential equality: For the same epsilon value, approximate equality is weaker requirement than essential equality.

---

essentially_equal/3

Compares two numbers for essential equality given an epsilon value using the de facto standard formula abs(Number1 - Number2) =< min(abs(Number1), abs(Number2)) * Epsilon. Type-checked.

Compilation flags:
static

Template:
essentially_equal(Number1,Number2,Epsilon)
Mode and number of proofs:
essentially_equal(+number,+number,+number) - zero_or_one

Remarks:

- Comparison with approximate equality: For the same epsilon value, essential equality is a stronger requirement than approximate equality.

---

tolerance_equal/4

Compares two numbers for close equality given relative and absolute tolerances using the de facto standard formula abs(Number1 - Number2) =< max(RelativeTolerance * max(abs(Number1), abs(Number2)), AbsoluteTolerance). Type-checked.

Compilation flags:
static

Template:
tolerance_equal(Number1,Number2,RelativeTolerance,AbsoluteTolerance)
Mode and number of proofs:

tolerance_equal(+number,+number,+number,+number) - zero_or_one

---

=~= / 2

Compares two numbers (or lists of numbers) for approximate equality using 100*epsilon for the absolute error and, if that fails, 99.999% accuracy for the relative error. But these precision values may not be adequate for all cases. Type-checked.

Compilation flags:
static

Template:
=~=(Number1,Number2)
Mode and number of proofs:
=~=(+number,+number) - zero_or_one
=~=(+list(number),+list(number)) - zero_or_one

---

epsilon/1

Returns the value of epsilon used in the definition of the (=~=)/2 predicate.

Compilation flags:
static

Template:
epsilon(Epsilon)
Mode and number of proofs:
epsilon(-float) - one

Protected predicates

run_tests/0

Runs all defined unit tests.

Compilation flags:
    static

Mode and number of proofs:
    run_tests - one

---

run_tests/1

Runs all the tests defined in the given file.

Compilation flags:
    static

Template:
    run_tests(File)
Mode and number of proofs:
    run_tests(+atom) - one

---

run_test_set/0

Runs a test set as part of running two or more test sets as a unified set.

Compilation flags:
    static

Mode and number of proofs:
    run_test_set - one

---

run_quick_check_tests/5

Runs a QuickCheck test using the given options. Returns the starting seed used to generate the random tests, the number of discarded tests, and the test label statistics.

Compilation flags:
    static

Template:
    run_quick_check_tests(Template,Options,Seed,Discarded,Labels)
Meta-predicate template:
    run_quick_check_tests(::,::,*,*,*)
Mode and number of proofs:
    run_quick_check_tests(@callable,+list,--nonvar,--number,--list(pair)) - one_or_error

---

condition/0

Verifies conditions for running the tests. Defaults to the goal true.

Compilation flags:
    static

Mode and number of proofs:
    condition - zero_or_one

---

setup/0

Setup environment before running the test set. Defaults to the goal true.

Compilation flags:
    static

Mode and number of proofs:
    setup - zero_or_one

---

cleanup/0

Cleanup environment after running the test set. Defaults to the goal true.

Compilation flags:
    static

Mode and number of proofs:
    cleanup - zero_or_one

---

make/1

Make target for automatically running the test set when calling the logtalk_make/1 built-in predicate. No default. Possible values are all and check.

Compilation flags:
    static

Template:
    make(Target)
Mode and number of proofs:
    make(?atom) - zero_or_one

---

note/1

Note to be printed after the test results. Defaults to the empty atom.

Compilation flags:
    static

Template:
    note(Note)
Mode and number of proofs:
    note(?atom) - zero_or_one

---

file_path/2

Returns the absolute path for a file path that is relative to the tests object path. When the file path is already an absolute path, it is expanded to resolve any remaining relative file path parts.

Compilation flags:
    static

Template:
    file_path(File,Path)
Mode and number of proofs:
    file_path(+atom,-atom) - one

See also:
    clean_file/1
    clean_directory/1

---

file_url/2

Returns the URL for a file path that can be either absolute or relative to the tests object path. When the file path is an absolute path, it is expanded to resolve any remaining relative file path parts.

Compilation flags:
    static

Template:
    file_url(File,URL)
Mode and number of proofs:
    file_url(+atom,-atom) - one

---

suppress_text_output/0

Suppresses text output. Useful to avoid irrelevant text output from predicates being tested to clutter the test logs.

Compilation flags:
    static

Mode and number of proofs:
    suppress_text_output - one

---

suppress_binary_output/0

Suppresses binary output. Useful to avoid irrelevant binary output from predicates being tested to clutter the test logs.

Compilation flags:
    static

Mode and number of proofs:
    suppress_binary_output - one

---

set_text_input/3

Creates a temporary file, in the same directory as the tests object, with the given text contents, and opens it for reading referenced by the given alias and using the additional options. If no eof_action/1 option is specified, its value will be the default used by the backend compiler.

Compilation flags:
    static

Template:
    set_text_input(Alias,Contents,Options)
Mode and number of proofs:
    set_text_input(+atom,+atom,+list(stream_option)) - one
    set_text_input(+atom,+list(atom),+list(stream_option)) - one

See also:
text_input_assertion/3
check_text_input/2
clean_text_input/0

---

set_text_input/2

Creates a temporary file, in the same directory as the tests object, with the given text contents, and opens it for reading referenced by the given alias and using the default end-of-file action for the used backend compiler.

Compilation flags:
static

Template:
set_text_input(Alias,Contents)
Mode and number of proofs:
set_text_input(+atom,+atom) - one
set_text_input(+atom,+list(atom)) - one

See also:
text_input_assertion/3
check_text_input/2
clean_text_input/0

---

set_text_input/1

Creates a temporary file, in the same directory as the tests object, with the given text contents, opens it for reading using the default end-of-file action for the used backend compiler, and sets the current input stream to the file.

Compilation flags:
static

Template:
set_text_input(Contents)
Mode and number of proofs:
set_text_input(+atom) - one

set_text_input(+list(atom)) - one

See also:
text_input_assertion/2
check_text_input/1
clean_text_input/0

---

check_text_input/2

Checks that the temporary file (referenced by the given alias) being read have the expected text contents.

Compilation flags:
static

Template:
check_text_input(Alias,Contents)
Mode and number of proofs:
check_text_input(+atom,+atom) - zero_or_one

See also:
set_text_input/2
set_text_input/2
text_input_assertion/3
clean_text_input/0

---

check_text_input/1

Checks that the temporary file being read have the expected text contents.

Compilation flags:
static

Template:
check_text_input(Contents)
Mode and number of proofs:
check_text_input(+atom) - zero_or_one

See also:
set_text_input/1
text_input_assertion/2
clean_text_input/0

---

text_input_assertion/3

Returns an assertion for checking that the temporary file (referenced by the given alias) being read have the expected text contents.

Compilation flags:
static

Template:
text_input_assertion(Alias,Contents,Assertion)
Mode and number of proofs:
text_input_assertion(+atom,+atom,--callable) - one

See also:
set_text_input/3
check_text_input/2
clean_text_input/0

---

text_input_assertion/2

Returns an assertion for checking that the temporary file being read have the expected text contents.

Compilation flags:
static

Template:
text_input_assertion(Contents,Assertion)
Mode and number of proofs:
text_input_assertion(+atom,--callable) - one

See also:
set_text_input/1

check_text_input/1
clean_text_input/0

---

clean_text_input/0

Cleans the temporary file used when testing text input.

Compilation flags:
static

Mode and number of proofs:
clean_text_input - one

See also:
set_text_input/3
set_text_input/2
set_text_input/1

---

set_binary_input/3

Creates a temporary file, in the same directory as the tests object, with the given binary contents, and opens it for reading referenced by the given alias and using the additional options. If no eof_action/1 option is specified, its value will be the default used by the backend compiler.

Compilation flags:
static

Template:
set_binary_input(Alias,Bytes,Options)
Mode and number of proofs:
set_binary_input(+atom,+list(byte),+list(stream_option)) - one

See also:
binary_input_assertion/3
check_binary_input/2
clean_binary_input/0

---

set_binary_input/2

Creates a temporary file, in the same directory as the tests object, with the given binary contents, and opens it for reading referenced by the given alias and using the default end-of-file action for the used backend compiler.

Compilation flags:
    static

Template:
    set_binary_input(Alias,Bytes)
Mode and number of proofs:
    set_binary_input(+atom,+list(byte)) - one

See also:
    binary_input_assertion/3
    check_binary_input/2
    clean_binary_input/0

---

set_binary_input/1

Creates a temporary file, in the same directory as the tests object, with the given binary contents, and opens it for reading using the default end-of-file action for the used backend compiler, and sets the current input stream to the file.

Compilation flags:
    static

Template:
    set_binary_input(Bytes)
Mode and number of proofs:
    set_binary_input(+list(byte)) - one

See also:
    binary_input_assertion/2
    check_binary_input/1
    clean_binary_input/0

---

check_binary_input/2

Checks that the temporary file (referenced by the given alias) being read have the expected binary contents.

Compilation flags:
    static

Template:
    check_binary_input(Alias,Bytes)
Mode and number of proofs:
    check_binary_input(+atom,+list(byte)) - zero_or_one

See also:
    set_binary_input/3
    set_binary_input/2
    binary_input_assertion/3
    clean_binary_input/0

---

check_binary_input/1

Checks that the temporary file being read have the expected binary contents.

Compilation flags:
    static

Template:
    check_binary_input(Bytes)
Mode and number of proofs:
    check_binary_input(+list(byte)) - zero_or_one

See also:
    binary_input_assertion/2
    set_binary_input/1
    clean_binary_input/0

---

binary_input_assertion/3

Returns an assertion for checking that the temporary file (referenced by the given alias) being read have the expected binary contents.

Compilation flags:
    static

Template:
    binary_input_assertion(Alias,Bytes,Assertion)
Mode and number of proofs:
    binary_input_assertion(+atom,+list(byte),--callable) - one

See also:
    check_binary_input/2
    set_binary_input/3
    set_binary_input/2
    clean_binary_input/0

---

binary_input_assertion/2

Returns an assertion for checking that the temporary file being read have the expected binary contents.

Compilation flags:
    static

Template:
    binary_input_assertion(Bytes,Assertion)
Mode and number of proofs:
    binary_input_assertion(+list(byte),--callable) - one

See also:
    check_binary_input/1
    set_binary_input/1
    clean_binary_input/0

---

clean_binary_input/0

Cleans the temporary file used when testing binary input.

Compilation flags:
    static

Mode and number of proofs:
    clean_binary_input - one

See also:
    set_binary_input/3
    set_binary_input/2
    set_binary_input/1

---

set_text_output/3

Creates a temporary file, in the same directory as the tests object, with the given text contents, and opens it for writing referenced by the given alias and using the additional options.

Compilation flags:
    static

Template:
    set_text_output(Alias,Contents,Options)
Mode and number of proofs:
    set_text_output(+atom,+atom,+list(stream_option)) - one
    set_text_output(+atom,+list(atom),+list(stream_option)) - one

See also:
    text_output_assertion/4
    check_text_output/3
    clean_text_output/0

---

set_text_output/2

Creates a temporary file, in the same directory as the tests object, with the given text contents, and referenced by the given alias.

Compilation flags:
    static

Template:
    set_text_output(Alias,Contents)
Mode and number of proofs:
    set_text_output(+atom,+atom) - one
    set_text_output(+atom,+list(atom)) - one

See also:
    text_output_assertion/3
    check_text_output/2
    clean_text_output/0

---

set_text_output/1

Creates a temporary file, in the same directory as the tests object, with the given text contents, and sets the current output stream to the file.

Compilation flags:
    static

Template:
    set_text_output(Contents)
Mode and number of proofs:
    set_text_output(+atom) - one
    set_text_output(+list(atom)) - one

See also:
    text_output_assertion/2
    check_text_output/1
    clean_text_output/0

---

check_text_output/3

Checks that the temporary file (open with the given options and alias in the same directory as the tests object) being written have the expected text contents.

Compilation flags:
    static

Template:
    check_text_output(Alias,Contents,Options)
Mode and number of proofs:
    check_text_output(+atom,+atom,+list(stream_option)) - zero_or_one

See also:
    set_text_output/3
    text_output_assertion/4
    clean_text_output/0

---

check_text_output/2

Checks that the temporary file (open with default options and alias in the same directory as the tests object) being written have the expected text contents.

Compilation flags:
    static

Template:
    check_text_output(Alias,Contents)
Mode and number of proofs:
    check_text_output(+atom,+atom) - zero_or_one

See also:
    set_text_output/2
    text_output_assertion/3
    clean_text_output/0

---

check_text_output/1

Checks that the temporary file being written have the expected text contents.

Compilation flags:
    static

Template:
    check_text_output(Contents)
Mode and number of proofs:
    check_text_output(+atom) - zero_or_one

See also:
    set_text_output/1
    text_output_assertion/2
    clean_text_output/0

---

text_output_assertion/4

Returns an assertion for checking that the temporary file (open with the given options and alias in the same directory as the tests object) being written have the expected text contents.

Compilation flags:
    static

Template:
    text_output_assertion(Alias,Contents,Options,Assertion)
Mode and number of proofs:
    text_output_assertion(+atom,+atom,+list(stream_option),--callable) - one

See also:
    set_text_output/3
    check_text_output/3
    clean_text_output/0

---

text_output_assertion/3

Returns an assertion for checking that the temporary file (open with default options and alias in the same directory as the tests object) being written have the expected text contents.

Compilation flags:
    static

Template:
    text_output_assertion(Alias,Contents,Assertion)
Mode and number of proofs:
    text_output_assertion(+atom,+atom,--callable) - one

See also:
    set_text_output/2
    check_text_output/2
    clean_text_output/0

---

text_output_assertion/2

Returns an assertion for checking that the temporary file (open with default options in the same directory as the tests object) being written have the expected text contents.

Compilation flags:
    static

Template:
    text_output_assertion(Contents,Assertion)
Mode and number of proofs:
    text_output_assertion(+atom,--callable) - one

See also:
    set_text_output/1
    check_text_output/1
    clean_text_output/0

---

text_output_contents/3

Returns the contents of the temporary file (open with the given options and alias in the same directory as the tests object) being written.

Compilation flags:
    static

Template:
    text_output_contents(Alias,Contents,Options)
Mode and number of proofs:
    text_output_contents(+atom,-list(character),+list(stream_option)) - one

---

text_output_contents/2

Returns the contents of the temporary file (open with default options and alias in the same directory as the tests object) being written.

Compilation flags:
    static

Template:
    text_output_contents(Alias,Contents)
Mode and number of proofs:
    text_output_contents(+atom,-list(character)) - one

---

text_output_contents/1

Returns the contents of the temporary file (open with default options in the same directory as the tests object) being written.

Compilation flags:
    static

Template:
    text_output_contents(Contents)
Mode and number of proofs:

text_output_contents(-list(character)) - one

---

clean_text_output/0

Cleans the temporary file used when testing text output.

Compilation flags:
static

Mode and number of proofs:
clean_text_output - one

See also:
set_text_output/3
set_text_output/2
set_text_output/1

---

set_binary_output/3

Creates a temporary file, in the same directory as the tests object, with the given binary contents, and opens it for writing referenced by the given alias and using the additional options.

Compilation flags:
static

Template:
set_binary_output(Alias,Contents,Options)
Mode and number of proofs:
set_binary_output(+atom,+list(byte),+list(stream_option)) - one

See also:
binary_output_assertion/3
check_binary_output/2
clean_binary_output/0

---

set_binary_output/2

Creates a temporary file, in the same directory as the tests object, with the given binary contents, and opens it for writing referenced with the given alias.

Compilation flags:
    static

Template:
    set_binary_output(Alias,Bytes)
Mode and number of proofs:
    set_binary_output(+atom,+list(byte)) - one

See also:
    binary_output_assertion/3
    check_binary_output/2
    clean_binary_output/0

---

set_binary_output/1

Creates a temporary file, in the same directory as the tests object, with the given binary contents, and sets the current output stream to the file.

Compilation flags:
    static

Template:
    set_binary_output(Bytes)
Mode and number of proofs:
    set_binary_output(+list(byte)) - one

See also:
    binary_output_assertion/2
    check_binary_output/1
    clean_binary_output/0

---

check_binary_output/2

Checks that the temporary file (referenced by the given alias) have the expected binary contents.

Compilation flags:
    static

Template:
    check_binary_output(Alias,Bytes)
Mode and number of proofs:
    check_binary_output(+atom,+list(byte)) - zero_or_one

See also:
    set_binary_output/3
    set_binary_output/2
    binary_output_assertion/3
    clean_binary_output/0

---

check_binary_output/1

Checks that the temporary file (open in the same directory as the tests object) have the expected binary contents.

Compilation flags:
    static

Template:
    check_binary_output(Bytes)
Mode and number of proofs:
    check_binary_output(+list(byte)) - zero_or_one

See also:
    set_binary_output/1
    binary_output_assertion/2
    clean_binary_output/0

---

binary_output_assertion/3

Returns an assertion for checking that the temporary file (referenced by the given alias) have the expected binary contents.

Compilation flags:
    static

Template:
    binary_output_assertion(Alias,Bytes,Assertion)
Mode and number of proofs:
    binary_output_assertion(+atom,+list(byte),--callable) - one

See also:
    set_binary_output/2
    check_binary_output/2
    clean_binary_output/0

---

binary_output_assertion/2

Returns an assertion for checking that the temporary file (open in the same directory as the tests object) have the expected binary contents.

Compilation flags:
    static

Template:
    binary_output_assertion(Bytes,Assertion)
Mode and number of proofs:
    binary_output_assertion(+list(byte),--callable) - one

See also:
    set_binary_output/1
    check_binary_output/1
    clean_binary_output/0

---

binary_output_contents/2

Returns the binary contents of the temporary file (referenced by the given alias) being written.

Compilation flags:
    static

Template:
    binary_output_contents(Alias,Bytes)
Mode and number of proofs:
    binary_output_contents(+atom,-list(byte)) - one

---

binary_output_contents/1

Returns the binary contents of the temporary file being written.

Compilation flags:
    static

Template:
    binary_output_contents(Bytes)
Mode and number of proofs:
    binary_output_contents(-list(byte)) - one

---

clean_binary_output/0

Cleans the temporary file used when testing binary output.

Compilation flags:
    static

Mode and number of proofs:
    clean_binary_output - one

See also:
    set_binary_output/3

set_binary_output/2
set_binary_output/1

---

create_text_file/3

Creates a text file with the given contents. The file is open for writing using the given options. Relative file paths are interpreted as relative to the tests object path.

Compilation flags:
static

Template:
create_text_file(File,Contents,Options)
Mode and number of proofs:
create_text_file(+atom,+atom,+list(stream_option)) - one
create_text_file(+atom,+list(atom),+list(stream_option)) - one

---

create_text_file/2

Creates a text file with the given contents. The file is open for writing using default options. Relative file paths are interpreted as relative to the tests object path.

Compilation flags:
static

Template:
create_text_file(File,Contents)
Mode and number of proofs:
create_text_file(+atom,+atom) - one
create_text_file(+atom,+list(atom)) - one

---

create_binary_file/2

Creates a binary file with the given contents. Relative file paths are interpreted as relative to the tests object path.

Compilation flags:
    static

Template:
    create_binary_file(File,Bytes)
Mode and number of proofs:
    create_binary_file(+atom,+list(byte)) - one

---

check_text_file/3

Checks that the contents of a text file match the expected contents. The file is open for reading using the given options. Relative file paths are interpreted as relative to the tests object path.

Compilation flags:
    static

Template:
    check_text_file(File,Contents,Options)
Mode and number of proofs:
    check_text_file(+atom,+atom,+list(stream_option)) - zero_or_one

See also:
    text_file_assertion/4

---

check_text_file/2

Checks that the contents of a text file (open for reading using default options) match the expected contents. Relative file paths are interpreted as relative to the tests object path.

Compilation flags:
    static

Template:
check_text_file(File,Contents)
Mode and number of proofs:
check_text_file(+atom,+atom) - zero_or_one

See also:
text_file_assertion/3

---

text_file_assertion/4

Returns an assertion for checking that the given file have the expected text contents. The file is open for reading using the given options. Relative file paths are interpreted as relative to the tests object path.

Compilation flags:
static

Template:
text_file_assertion(File,Contents,Options,Assertion)
Mode and number of proofs:
text_file_assertion(+atom,+atom,+list(stream_option),--callable) - one

See also:
check_text_file/3

---

text_file_assertion/3

Returns an assertion for checking that the given file have the expected text contents. The file is open for reading using default options. Relative file paths are interpreted as relative to the tests object path.

Compilation flags:
static

Template:
text_file_assertion(File,Contents,Assertion)
Mode and number of proofs:
text_file_assertion(+atom,+atom,--callable) - one

See also:
check_text_file/2

---

check_binary_file/2

Checks the contents of a binary file match the expected contents. Relative file paths are interpreted as relative to the tests object path.

Compilation flags:
static

Template:
check_binary_file(File,Bytes)
Mode and number of proofs:
check_binary_file(+atom,+list(byte)) - zero_or_one

See also:
binary_file_assertion/3

---

binary_file_assertion/3

Returns an assertion for checking that the given file have the expected binary contents. Relative file paths are interpreted as relative to the tests object path.

Compilation flags:
static

Template:
binary_file_assertion(File,Bytes,Assertion)
Mode and number of proofs:
binary_file_assertion(+atom,+list(byte),--callable) - one

See also:
check_binary_file/2

---

clean_file/1

Closes any existing stream associated with the file and deletes the file if it exists. Relative file paths are interpreted as relative to the tests object path.

Compilation flags:
    static

Template:
    clean_file(File)
Mode and number of proofs:
    clean_file(+atom) - one

See also:
    clean_directory/1
    file_path/2

---

clean_directory/1

Deletes a directory if it exists. Relative directory paths are interpreted as relative to the tests object path.

Compilation flags:
    static

Template:
    clean_directory(Directory)
Mode and number of proofs:
    clean_directory(+atom) - one

See also:
    clean_file/1
    file_path/2

---

closed_input_stream/2

Opens a temporary file in the same directory as the tests object with the given options for reading, closes it, and returns its stream handle.

Compilation flags:
    static

Template:
    closed_input_stream(Stream,Options)
Mode and number of proofs:
    closed_input_stream(-stream,+list(stream_option)) - one

---

closed_output_stream/2

Opens a temporary file in the same directory as the tests object with the given options for writing, closes it, and returns its stream handle.

Compilation flags:
    static

Template:
    closed_output_stream(Stream,Options)
Mode and number of proofs:
    closed_output_stream(-stream,+list(stream_option)) - zero_or_one

---

stream_position/1

Returns a syntactically valid stream position by opening a temporary file in the same directory as the tests object.

Compilation flags:
    static

Template:
    stream_position(Position)
Mode and number of proofs:

stream_position(-stream_position) - one

---

test/2

Table of defined tests.

Compilation flags:
static

Template:
test(Identifier,Test)
Mode and number of proofs:
test(?callable,?compound) - zero_or_more

Private predicates

running_test_sets_/0

Internal flag used when running two or more test sets as a unified set.

Compilation flags:
dynamic

Mode and number of proofs:
running_test_sets_ - zero_or_one

---

test/3

Compiled unit tests. The list of variables is used to ensure variable sharing between a test with its test options.

Compilation flags:
static

Template:
test(Identifier,Variables,Outcome)
Mode and number of proofs:
test(?callable,?list(variable),?nonvar) - zero_or_more

---

auxiliary_predicate_counter_/1

Counter for generating unique auxiliary predicate names.

Compilation flags:
dynamic

Template:
auxiliary_predicate_counter_(Counter)
Mode and number of proofs:
auxiliary_predicate_counter_(?integer) - one_or_more

---

test_/2

Table of compiled tests.

Compilation flags:
dynamic

Template:
test_(Identifier,Test)
Mode and number of proofs:
test_(?callable,?compound) - zero_or_more

---

selected_test_/1

Table of selected tests for execution.

Compilation flags:
    dynamic

Template:
    selected_test_(Identifier)
Mode and number of proofs:
    selected_test_(?callable) - zero_or_more

---

linter_warning_sequence_/1

Counter for cached test-compilation linter warnings.

Compilation flags:
    dynamic

Template:
    linter_warning_sequence_(Sequence)
Mode and number of proofs:
    linter_warning_sequence_(?integer) - zero_or_one

---

recorded_linter_warning_/7

Cache of test-compilation linter warnings.

Compilation flags:
    dynamic

Template:
    recorded_linter_warning_(Sequence,RuleId,Message,Context,File,Lines,Properties)
Mode and number of proofs:
    recorded_linter_warning_(?integer,?atom,?atom,?compound,?atom,?compound,?list(compound)) - zero_or_more

---

skipped_/1

Counter for skipped tests.

Compilation flags:
    dynamic

Template:
    skipped_(Counter)
Mode and number of proofs:
    skipped_(?integer) - zero_or_one

---

passed_/3

Counter and total time for passed tests.

Compilation flags:
    dynamic

Template:
    passed_(Counter,CPUTime,WallTime)
Mode and number of proofs:
    passed_(?integer,-float,-float) - zero_or_one

---

failed_/3

Counter and total time for failed tests.

Compilation flags:
    dynamic

Template:
    failed_(Counter,CPUTime,WallTime)
Mode and number of proofs:

failed_(?integer,-float,-float) - zero_or_one

---

flaky_/1

Counter for failed tests that are marked as flaky.

Compilation flags:
dynamic

Template:
flaky_(Counter)
Mode and number of proofs:
flaky_(?integer) - zero_or_one

---

fired_/3

Fired clauses when running the unit tests.

Compilation flags:
dynamic

Template:
fired_(Entity,Predicate,Clause)
Mode and number of proofs:
fired_(?entity_identifier,?predicate_indicator,?integer) - zero_or_more

---

covered_/4

Auxiliary predicate for collecting statistics on clause coverage.

Compilation flags:
dynamic

Template:

covered_(Entity,Predicate,Covered,Total)
Mode and number of proofs:
covered_(?entity_identifier,?callable,?integer,?integer) - zero_or_more

---

Operators

op(700,xfx,=~=)

Scope:
public

category

1.136.8 lgtunit_messages

Logtalk unit test framework default message translations.

Availability:
logtalk_load(lgtunit(loader))

Author: Paulo Moura
Version: 12:2:0
Date: 2025-10-20

Compilation flags:
static

Provides:
logtalk::message_prefix_stream/4
logtalk::message_tokens//2

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - failed_test_reason//1
- Protected predicates
- Private predicates
- Operators

Public predicates

failed_test_reason//1

Used to rewrite a term representing the reason why a term failed into a list of tokens.

Compilation flags:
    static

Template:
    failed_test_reason(Reason)
Mode and number of proofs:
    failed_test_reason(@nonvar) - one_or_more

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.136.9 minimal_output

Intercepts unit test execution messages and outputs a minimal report.

Availability:
logtalk_load(lgtunit(loader))

Author: Paulo Moura
Version: 3:0:0
Date: 2021-05-27

Compilation flags:
static, context_switching_calls

Provides:
logtalk::message_hook/4

Remarks:

- Usage: Simply load this object before running your tests using the goal logtalk_load(lgtunit(minimal_output)).
- Limitations: Cannot be used when the test objects also intercept lgtunit messages.

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.136.10 subunit_v1_output

Intercepts unit test execution messages and outputs a Subunit v1 text stream to the current output stream.

Availability:
logtalk_load(lgtunit(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-02

Compilation flags:
static, context_switching_calls

Provides:
logtalk::message_hook/4
Uses:
os

Remarks:

- Usage: Simply load this object before running your tests using the goal logtalk_load(lgtunit(subunit_v1_output)).

Inherited public predicates:

(none)

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.136.11 subunit_v1_report

Intercepts unit test execution messages and generates a subunit_v1_report.txt file using the Subunit v1 text streaming format in the same directory as the tests object file.

Availability:
logtalk_load(lgtunit(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-02

Compilation flags:
static, context_switching_calls

Provides:

logtalk::message_hook/4

Uses:

logtalk

os

Remarks:

- Usage: Simply load this object before running your tests using the goal logtalk_load(lgtunit(subunit_v1_report)).

Inherited public predicates:

(none)

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.136.12 subunit_v2_output

Intercepts unit test execution messages and outputs a Subunit v2 binary stream to the current output stream.

Availability:
logtalk_load(lgtunit(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-02

Compilation flags:
static, context_switching_calls

Provides:
logtalk::message_hook/4
Uses:
list
os
term_io
user

Remarks:

- Usage: Simply load this object before running your tests using the goal logtalk_load([basic_types(loader),term_io(loader),lgtunit(subunit_v2_output)]).

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.136.13 subunit_v2_report

Intercepts unit test execution messages and generates a subunit_v2_report.bin file using the Subunit v2 binary streaming format in the same directory as the tests object file.

Availability:
logtalk_load(lgtunit(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-02

Compilation flags:
static, context_switching_calls

Provides:
logtalk::message_hook/4
Uses:
list
logtalk
os
term_io
user

Remarks:

- Usage: Simply load this object before running your tests using the goal logtalk_load([basic_types(loader),term_io(loader),lgtunit(subunit_v2_report)]).

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.136.14 tap_output

Intercepts unit test execution messages and outputs a report using the TAP format to the current output stream.

Availability:
logtalk_load(lgtunit(loader))

Author: Paulo Moura
Version: 5:0:0
Date: 2025-04-07

Compilation flags:
static, context_switching_calls

Provides:
logtalk::message_hook/4

Remarks:

- Usage: Simply load this object before running your tests using the goal logtalk_load(lgtunit(tap_output)).

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
  - generating_/0
  - partial_/1
  - test_count_/1
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

generating_/0

Flag to detect report in progress when processing two or more test sets as a unified set.

Compilation flags:
dynamic

Mode and number of proofs:
    generating_ - zero_or_one

---

partial_/1

Cache of total of tests per test set.

Compilation flags:
    dynamic

Template:
    partial_(Count)
Mode and number of proofs:
    partial_(?integer) - zero_or_more

---

test_count_/1

Test counter.

Compilation flags:
    dynamic

Template:
    test_count_(Count)
Mode and number of proofs:
    test_count_(?integer) - zero_or_one

Operators

(none)

object

1.136.15 tap_report

Intercepts unit test execution messages and generates a tap_report.txt file using the TAP output format in the same directory as the tests object file.

Availability:
logtalk_load(lgtunit(loader))

Author: Paulo Moura
Version: 6:0:0
Date: 2025-04-07

Compilation flags:
static, context_switching_calls

Provides:
logtalk::message_hook/4
Uses:
logtalk

Remarks:

- Usage: Simply load this object before running your tests using the goal logtalk_load(lgtunit(tap_report)).

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
  - partial_/1
  - test_count_/1
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

partial_/1

Cache of total of tests per test set.

Compilation flags:
    dynamic

Template:
    partial_(Count)
Mode and number of proofs:
    partial_(?integer) - zero_or_more

---

test_count_/1

Test counter.

Compilation flags:
    dynamic

Template:
    test_count_(Count)
Mode and number of proofs:
    test_count_(?integer) - zero_or_one

Operators

(none)

object

1.136.16 xunit_net_v2_output

Intercepts unit test execution messages and outputs a report using the xUnit.net v2 XML format to the current output stream.

Availability:
logtalk_load(lgtunit(loader))

Author: Paulo Moura
Version: 5:0:0
Date: 2025-04-07

Compilation flags:
static, context_switching_calls

Provides:
logtalk::message_hook/4
Uses:
user

Remarks:

- Usage: Simply load this object before running your tests using the goal logtalk_load(lgtunit(xunit_net_v2_output)).

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
  - message_cache_/1
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

message_cache_/1

Table of messages emitted by the lgtunit tool when running tests.

Compilation flags:
dynamic

Template:
message_cache_(Message)
Mode and number of proofs:
message_cache_(?callable) - zero_or_more

Operators

(none)

object

1.136.17 xunit_net_v2_report

Intercepts unit test execution messages and generates a xunit_report.xml file using the xUnit.net v2 XML format in the same directory as the tests object file.

Availability:
logtalk_load(lgtunit(loader))

Author: Paulo Moura
Version: 6:0:0
Date: 2025-04-07

Compilation flags:

static, context_switching_calls

Provides:
logtalk::message_hook/4
Uses:
logtalk
user

Remarks:

- Usage: Simply load this object before running your tests using the goal logtalk_load(lgtunit(xunit_net_v2_report)).

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
  - message_cache_/1
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

message_cache_/1

Table of messages emitted by the lgtunit tool when running tests.

Compilation flags:
dynamic

Template:
message_cache_(Message)
Mode and number of proofs:
message_cache_(?callable) - zero_or_more

Operators

(none)

object

1.136.18 xunit_output

Intercepts unit test execution messages and outputs a report using the xUnit XML format to the current output stream.

Availability:
logtalk_load(lgtunit(loader))

Author: Paulo Moura
Version: 5:0:1
Date: 2025-04-11

Compilation flags:
static, context_switching_calls

Provides:
logtalk::message_hook/4
Uses:
logtalk
user

Remarks:

- Usage: Simply load this object before running your tests using the goal logtalk_load(lgtunit(xunit_output)).

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
  - message_cache_/1
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

message_cache_/1

Table of messages emitted by the lgtunit tool when running tests.

Compilation flags:
    dynamic

Template:
    message_cache_(Message)
Mode and number of proofs:
    message_cache_(?callable) - zero_or_more

Operators

(none)

object

1.136.19 xunit_report

Intercepts unit test execution messages and generates a xunit_report.xml file using the xUnit XML format in the same directory as the tests object file.

Availability:
logtalk_load(lgtunit(loader))

Author: Paulo Moura
Version: 6:0:0
Date: 2025-04-07

Compilation flags:
static, context_switching_calls

Provides:
logtalk::message_hook/4
Uses:
logtalk
user

Remarks:

- Usage: Simply load this object before running your tests using the goal logtalk_load(lgtunit(xunit_report)).

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
  - message_cache_/1
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

message_cache_/1

Table of messages emitted by the lgtunit tool when running tests.

Compilation flags:
    dynamic

Template:
    message_cache_(Message)
Mode and number of proofs:
    message_cache_(?callable) - zero_or_more

Operators

(none)

1.137 library

protocol

1.137.1 cloning

Object cloning protocol.

Availability:
    logtalk_load(library(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2010-09-14

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - clone/1
- Protected predicates
- Private predicates
- Operators

Public predicates

clone/1

Clones an object, returning the identifier of the new object if none is given.

Compilation flags:
static

Template:
clone(Clone)
Mode and number of proofs:
clone(?object) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

category

1.137.2 counters

Named integer counters. Counter names can be any nonvar term.

Availability:
logtalk_load(library(loader))

Author: Paulo Moura
Version: 1:0:1
Date: 2022-02-11

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - counter/2
  - increment_counter/1

  - decrement_counter/1
  - reset_counter/1
  - reset_counters/0
- Protected predicates
- Private predicates
  - counter_/2
- Operators

Public predicates

counter/2

True if Counter is a counter with value Value.

Compilation flags:
    static

Template:
    counter(Counter,Value)
Mode and number of proofs:
    counter(?nonvar,?integer) - zero_or_more

---

increment_counter/1

Increments the named counter.

Compilation flags:
    static

Template:
    increment_counter(Counter)
Mode and number of proofs:
    increment_counter(+nonvar) - one

---

decrement_counter/1

Decrements the named counter.

Compilation flags:
    static

Template:
    decrement_counter(Counter)
Mode and number of proofs:
    decrement_counter(+nonvar) - one

---

reset_counter/1

Resets the named counter to zero. Creates the counter if it does not exist.

Compilation flags:
    static

Template:
    reset_counter(Counter)
Mode and number of proofs:
    reset_counter(+nonvar) - one

---

reset_counters/0

Resets all existing named counters to zero.

Compilation flags:
    static

Mode and number of proofs:
    reset_counters - one

Protected predicates

(none)

Private predicates

counter_/2

Table of named counters.

Compilation flags:
    dynamic

Template:
    counter_(Counter,Value)
Mode and number of proofs:
    counter_(?nonvar,?integer) - zero_or_more

Operators

(none)

object

1.137.3 streamvars

Stream variables (supporting logical, backtracable, adding and retrieving of terms).

Availability:
    logtalk_load(library(loader))

Author: Nobukuni Kino and Paulo Moura
Version: 1:3:0
Date: 2019-06-15

Compilation flags:
    static, context_switching_calls

Dependencies:
    (none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - new/1
  - new/2
  - (<=)/2
  - (=>)/2
- Protected predicates
- Private predicates
- Operators
  - op(100,xfx,<=)
  - op(100,xfx,=>)

Public predicates

new/1

Makes Variable a stream variable. Initial state will be empty.

Compilation flags:
static

Template:
new(Variable)
Mode and number of proofs:
new(--streamvar) - one

Exceptions:
Variable is not a variable:
type_error(variable,Variable)

new/2

Makes Variable a stream variable and sets its initial state to Value.

Compilation flags:
    static

Template:
    new(Variable,Value)
Mode and number of proofs:
    new(--streamvar,@nonvar) - one

Exceptions:
    Variable is not a variable:
        type_error(variable,Variable)

(<=)/2

Sets the state of the stream variable Variable to Value (initializing the variable if needed).

Compilation flags:
    static

Template:
    Variable<=Value
Mode and number of proofs:
    (?streamvar)<=(@nonvar) - one

(=>)/2

Unifies Value with the current state of the stream variable Variable.

Compilation flags:
    static

Template:

Variable=>Value
Mode and number of proofs:
+streamvar=> ?nonvar - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

op(100,xfx,<=)

Scope:
public

op(100,xfx,=>)

Scope:
public

1.138 linda

object

1.138.1 linda

Linda tuple-space implementation for process communication. Provides a server that acts as a shared blackboard where clients can write (out/1-2), read (rd/1-2), and remove (in/1-2) tuples. Uses threaded engines for the server implementation and the sockets library for network communication.

Availability:
logtalk_load(linda(loader))

Author: Paulo Moura
Version: 2:0:0
Date: 2026-03-27

Compilation flags:
static, context_switching_calls, threaded

Imports:
public linda_server
public linda_client

Remarks:

- Supported backends: SWI-Prolog, Trealla Prolog, and XVM (requires both multi-threading and sockets support).
- Linda operations: The basic operations are out/1-2 (write tuple), in/1-2 (remove tuple, blocking), rd/1-2 (read tuple, blocking), in_noblock/1-2 (remove tuple, non-blocking), and rd_noblock/1-2 (read tuple, non-blocking).
- Tuple matching: Tuples are matched using unification.
- Blocking behavior: The in/1-2 and rd/1-2 predicates block until a matching tuple is available. The in_noblock/1-2 and rd_noblock/1-2 predicates fail immediately if no matching tuple is found.
- Multiple clients: Multiple clients can connect to the same server. A tuple removed by the in/1-2 or in_noblock/1-2 predicates is only removed for one client.
- Multiple servers: A client can connect to multiple servers. The first server it connects to, uses by default the blackboard alias.
- API compatibility: The API is inspired by the SICStus Prolog Linda library.
- Network communication: Uses TCP sockets for client-server communication, allowing processes to run on different machines.

Inherited public predicates:
check_option/1 check_options/1 close_client/1 default_option/1 default_options/1 findall_in_noblock/3 findall_in_noblock/4 findall_rd_noblock/3 findall_rd_noblock/4 in/1 in/2 in_list/2 in_list/3 in_noblock/1 in_noblock/2 linda/0 linda/1 linda_client/1 linda_client/2 linda_timeout/2 option/2 option/3 out/1 out/2 rd/1 rd/2 rd_list/2 rd_list/3 rd_noblock/1 rd_noblock/2 shutdown_server/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.138.2 linda_client

Linda client predicates and client-side connection state. Import into a threaded object together with the linda_server category.

Availability:
logtalk_load(linda(loader))

Author: Paulo Moura
Version: 2:0:1
Date: 2026-04-22

Compilation flags:
static

Extends:
public options
Uses:
os
socket

Remarks:
(none)

Inherited public predicates:

check_option/1 check_options/1 default_option/1 default_options/1 option/2 option/3 valid_option/1 valid_options/1

- Public predicates
  - linda_client/2
  - linda_client/1
  - close_client/1
  - shutdown_server/1
  - linda_timeout/2
  - out/2
  - out/1
  - in/2
  - in/1
  - in_noblock/2
  - in_noblock/1
  - in_list/3
  - in_list/2
  - rd/2
  - rd/1
  - rd_noblock/2
  - rd_noblock/1
  - rd_list/3
  - rd_list/2
  - findall_rd_noblock/4
  - findall_rd_noblock/3
  - findall_in_noblock/4
  - findall_in_noblock/3
- Protected predicates
- Private predicates
  - client_connection_input_/2
  - client_connection_output_/2
  - client_connection_alias_/2
  - client_timeout_/1
- Operators

Public predicates

linda_client/2

Connects to a Linda server at the given address (Host:Port) using the given options.

Compilation flags:
    static

Template:
    linda_client(Address,Options)
Mode and number of proofs:
    linda_client(++address,++list(compound)) - one_or_error

Exceptions:
    Already connected:
        linda_error(already_connected)
    Connection failed:
        linda_error(connection_failed(Error))

Remarks:

- Option alias(Alias): Use Alias as an alias to the server address. Default is blackboard.

---

linda_client/1

Connects to a Linda server at the given address (Host:Port) using default options.

Compilation flags:
    static

Template:
    linda_client(Address)
Mode and number of proofs:
    linda_client(++address) - one_or_error

Exceptions:
    Already connected:
        linda_error(already_connected)
    Connection failed:
        linda_error(connection_failed(Error))

close_client/1

Closes the connection to the given Linda server.

Compilation flags:
    static

Template:
    close_client(Address)
Mode and number of proofs:
    close_client(++address_or_alias) - one

shutdown_server/1

Sends a shutdown signal to the given server. The server stops accepting new connections but continues serving existing clients until they all disconnect. Call close_client/1 after this predicate.

Compilation flags:
    static

Template:
    shutdown_server(Address)
Mode and number of proofs:
    shutdown_server(++address_or_alias) - one_or_error

Exceptions:
    Not connected:
        linda_error(not_connected(Address))

linda_timeout/2

Gets or sets the client timeout. OldTime is unified with the current timeout and the timeout is set to NewTime. The timeout value is either off (no timeout, wait forever) or Seconds:Milliseconds.

Compilation flags:
    static

Template:
    linda_timeout(OldTime,NewTime)
Mode and number of proofs:
    linda_timeout(?compound,+compound) - one

---

out/2

Places the tuple Tuple in the tuple-space of the given server.

Compilation flags:
    static

Template:
    out(AddressOrALias,Tuple)
Mode and number of proofs:
    out(++address_or_alias,+term) - one

---

out/1

Places the tuple Tuple in the tuple-space of the default server.

Compilation flags:
    static

Template:
    out(Tuple)
Mode and number of proofs:
    out(+term) - one

---

in/2

Removes a tuple matching Tuple from the tuple-space of the given server. Blocks if no matching tuple is available.

Compilation flags:
    static

Template:
    in(AddressOrALias,Tuple)
Mode and number of proofs:
    in(++address_or_alias,?term) - one

---

in/1

Removes a tuple matching Tuple from the tuple-space of the default server. Blocks if no matching tuple is available.

Compilation flags:
    static

Template:
    in(Tuple)
Mode and number of proofs:
    in(?term) - one

---

in_noblock/2

Removes a tuple matching Tuple from the tuple-space of the default server. Fails if no matching tuple is available.

Compilation flags:
    static

Template:

in_noblock(AddressOrALias,Tuple)
Mode and number of proofs:
in_noblock(++address_or_alias,?term) - zero_or_one

---

in_noblock/1

Removes a tuple matching Tuple from the tuple-space of the default server. Fails if no matching tuple is available.

Compilation flags:
static

Template:
in_noblock(Tuple)
Mode and number of proofs:
in_noblock(?term) - zero_or_one

---

in_list/3

Removes a tuple matching one of the patterns in TupleList from the tuple-space of the given server. Tuple is unified with the matched tuple. Blocks if no matching tuple is available.

Compilation flags:
static

Template:
in_list(AddressOrALias,TupleList,Tuple)
Mode and number of proofs:
in_list(++address_or_alias,+list,?term) - one

---

in_list/2

Removes a tuple matching one of the patterns in TupleList from the tuple-space of the default server. Tuple is unified with the matched tuple. Blocks if no matching tuple is available.

Compilation flags:
    static

Template:
    in_list(TupleList,Tuple)
Mode and number of proofs:
    in_list(+list,?term) - one

---

rd/2

Reads a tuple matching Tuple from the tuple-space of the given server without removing it. Blocks if no matching tuple is available.

Compilation flags:
    static

Template:
    rd(AddressOrALias,Tuple)
Mode and number of proofs:
    rd(++address_or_alias,?term) - one

---

rd/1

Reads a tuple matching Tuple from the tuple-space of the default server without removing it. Blocks if no matching tuple is available.

Compilation flags:
    static

Template:
    rd(Tuple)
Mode and number of proofs:

rd(?term) - one

---

rd_noblock/2

Reads a tuple matching Tuple from the tuple-space of the given server without removing it. Fails if no matching tuple is available.

Compilation flags:
static

Template:
rd_noblock(AddressOrALias,Tuple)
Mode and number of proofs:
rd_noblock(++address_or_alias,?term) - zero_or_one

---

rd_noblock/1

Reads a tuple matching Tuple from the tuple-space of the default server without removing it. Fails if no matching tuple is available.

Compilation flags:
static

Template:
rd_noblock(Tuple)
Mode and number of proofs:
rd_noblock(?term) - zero_or_one

---

rd_list/3

Reads a tuple matching one of the patterns in TupleList from the tuple-space of the given server without removing it. Tuple is unified with the matched tuple. Blocks if no matching tuple is available.

Compilation flags:
    static

Template:
    rd_list(AddressOrALias,TupleList,Tuple)
Mode and number of proofs:
    rd_list(++address_or_alias,+list,?term) - one

---

rd_list/2

Reads a tuple matching one of the patterns in TupleList from the tuple-space of the default server without removing it. Tuple is unified with the matched tuple. Blocks if no matching tuple is available.

Compilation flags:
    static

Template:
    rd_list(TupleList,Tuple)
Mode and number of proofs:
    rd_list(+list,?term) - one

---

findall_rd_noblock/4

Returns a list of all instances of Template for tuples matching Tuple in the tuple-space. The operation is atomic.

Compilation flags:
    static

Template:
    findall_rd_noblock(AddressOrALias,Template,Tuple,List)
Mode and number of proofs:

findall_rd_noblock(++address_or_alias,?term,+term,?list) - one

---

findall_rd_noblock/3

Returns a list of all instances of Template for tuples matching Tuple in the tuple-space. The operation is atomic.

Compilation flags:
static

Template:
findall_rd_noblock(Template,Tuple,List)
Mode and number of proofs:
findall_rd_noblock(?term,+term,?list) - one

---

findall_in_noblock/4

Removes and returns a list of all instances of Template for tuples matching Tuple in the tuple-space. The operation is atomic - all matching tuples are removed in one synchronized operation.

Compilation flags:
static

Template:
findall_in_noblock(AddressOrALias,Template,Tuple,List)
Mode and number of proofs:
findall_in_noblock(++address_or_alias,?term,+term,?list) - one

---

findall_in_noblock/3

Removes and returns a list of all instances of Template for tuples matching Tuple in the tuple-space. The operation is atomic - all matching tuples are removed in one synchronized operation.

Compilation flags:
    static

Template:
    findall_in_noblock(Template,Tuple,List)
Mode and number of proofs:
    findall_in_noblock(?term,+term,?list) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

client_connection_input_/2

Stores the input stream for the client connection to the server.

Compilation flags:
    dynamic

Template:
    client_connection_input_(Address,InputStream)
Mode and number of proofs:
    client_connection_input_(?address,?stream) - zero_or_more

---

client_connection_output_/2

Stores the output stream for the client connection to the server.

Compilation flags:
    dynamic

Template:
    client_connection_output_(Address,OutputStream)
Mode and number of proofs:
    client_connection_output_(?address,?stream) - zero_or_more

---

client_connection_alias_/2

Stores the client connection server aliases.

Compilation flags:
    dynamic

Template:
    client_connection_alias_(Address,Alias)
Mode and number of proofs:
    client_connection_alias_(?address,?atom) - zero_or_more

---

client_timeout_/1

Stores the timeout value for blocking client operations. Value is either off or Seconds:Milliseconds.

Compilation flags:
    dynamic

Template:
    client_timeout_(Timeout)
Mode and number of proofs:
    client_timeout_(?compound) - zero_or_one

Operators

(none)

category

1.138.3 linda_server

Linda server predicates and tuple-space state. Import into a threaded object together with the linda_client category.

Availability:
logtalk_load(linda(loader))

Author: Paulo Moura
Version: 2:0:1
Date: 2026-04-22

Compilation flags:
static

Provides:
logtalk::message_prefix_stream/4
Uses:
list
logtalk
os
socket

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - linda/1
  - linda/0
- Protected predicates
- Private predicates

    - server_socket_/1
    - client_connection_/3
    - accept_hook_/1
    - server_running_/1
    - server_shutdown_/0
    - tuple_/1
    - waiting_/3
    - client_engine_/2
    - ts_out/1
    - ts_in/4
    - ts_in_noblock/2
    - ts_in_list/4
    - ts_rd/4
    - ts_rd_noblock/2
    - ts_rd_list/4
    - ts_findall_rd_noblock/3
    - ts_findall_in_noblock/3
- Operators

Public predicates

linda/1

Starts a Linda server with the given options. The predicate succeeds when all clients have disconnected after a shutdown request.

Compilation flags:
    static

Template:
    linda(Options)
Meta-predicate template:
    linda(::)
Mode and number of proofs:
    linda(+list) - one

Remarks:

- Option port(Port): Use Port as the server port. Must be an integer and an available port.
- Option Address-Goal: Address is unified with Host:Port and Goal is called when the server starts. Useful for saving the address or starting clients.
- Option accept_hook(Client,Input,Output,Goal): When a client connects, Client is unified with the client address, Input and Output with the connection streams, and Goal is called. If Goal fails, the connection is rejected.

---

linda/0

Starts a Linda server on an automatically assigned port using default options. The server address (Host:Port) is written to the current output stream. The predicate succeeds when all clients have disconnected after a shutdown request.

Compilation flags:
    static

Mode and number of proofs:
    linda - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

server_socket_/1

Stores the server socket descriptor.

Compilation flags:
    dynamic

Template:
    server_socket_(ServerSocket)
Mode and number of proofs:
    server_socket_(?term) - zero_or_one

---

client_connection_/3

Stores active client connections. Each client has an ID, input stream, and output stream.

Compilation flags:
    dynamic

Template:
    client_connection_(ClientId,InputStream,OutputStream)
Mode and number of proofs:
    client_connection_(?term,?term,?term) - zero_or_more

---

accept_hook_/1

Stores the optional accept hook goal to call when a client connects.

Compilation flags:
    dynamic

Template:
    accept_hook_(Hook)
Mode and number of proofs:
    accept_hook_(?callable) - zero_or_one

---

server_running_/1

Flag indicating the server is running.

Compilation flags:
    dynamic

Template:
    server_running_(Address)
Mode and number of proofs:
    server_running_(?address) - zero_or_one

---

server_shutdown_/0

Flag indicating the server has received a shutdown request.

Compilation flags:
    dynamic

Mode and number of proofs:
    server_shutdown_ - zero_or_one

---

tuple_/1

Stores tuples in the Linda tuple space.

Compilation flags:
    dynamic

Template:
    tuple_(Tuple)
Mode and number of proofs:
    tuple_(?term) - zero_or_more

---

waiting_/3

Stores blocked clients waiting for tuples. Records the client ID, request pattern, and output stream.

Compilation flags:
    dynamic

Template:
    waiting_(ClientId,Request,OutputStream)
Mode and number of proofs:
    waiting_(?term,?term,?term) - zero_or_more

---

client_engine_/2

Maps client IDs to their corresponding threaded engine names.

Compilation flags:
    dynamic

Template:
    client_engine_(ClientId,EngineName)
Mode and number of proofs:
    client_engine_(?term,?atom) - zero_or_more

---

ts_out/1

Synchronized predicate to add a tuple to the tuple space and wake waiting clients.

Compilation flags:
    static, synchronized

Template:
    ts_out(Tuple)
Mode and number of proofs:
    ts_out(+term) - one

---

ts_in/4

Synchronized predicate to remove a matching tuple or register a waiting client.

Compilation flags:
    static, synchronized

Template:
    ts_in(Tuple,ClientId,OutputStream,Found)
Mode and number of proofs:
    ts_in(+term,+term,+term,-compound) - one

---

ts_in_noblock/2

Synchronized predicate to try removing a matching tuple without blocking.

Compilation flags:
    static, synchronized

Template:
    ts_in_noblock(Tuple,Found)
Mode and number of proofs:
    ts_in_noblock(+term,-compound) - zero_or_one

---

ts_in_list/4

Synchronized predicate to remove a tuple matching one of multiple patterns or register a waiting client.

Compilation flags:
    static, synchronized

Template:
    ts_in_list(TupleList,ClientId,OutputStream,Found)
Mode and number of proofs:
    ts_in_list(+list,+term,+term,-compound) - one

---

ts_rd/4

Synchronized predicate to read a matching tuple or register a waiting client.

Compilation flags:
    static, synchronized

Template:
    ts_rd(Tuple,ClientId,OutputStream,Found)
Mode and number of proofs:
    ts_rd(+term,+term,+term,-compound) - one

---

ts_rd_noblock/2

Synchronized predicate to try reading a matching tuple without blocking.

Compilation flags:
    static, synchronized

Template:
    ts_rd_noblock(Tuple,Found)
Mode and number of proofs:
    ts_rd_noblock(+term,-compound) - zero_or_one

---

ts_rd_list/4

Synchronized predicate to read a tuple matching one of multiple patterns or register a waiting client.

Compilation flags:
    static, synchronized

Template:
    ts_rd_list(TupleList,ClientId,OutputStream,Found)
Mode and number of proofs:
    ts_rd_list(+list,+term,+term,-compound) - one

---

ts_findall_rd_noblock/3

Synchronized predicate to collect all tuples matching a pattern.

Compilation flags:
    static, synchronized

Template:
    ts_findall_rd_noblock(Template,Tuple,List)
Mode and number of proofs:
    ts_findall_rd_noblock(+term,+term,-list) - zero_or_more

---

ts_findall_in_noblock/3

Synchronized predicate to collect and remove all tuples matching a pattern.

Compilation flags:
    static, synchronized

Template:
    ts_findall_in_noblock(Template,Tuple,List)
Mode and number of proofs:
    ts_findall_in_noblock(+term,+term,-list) - zero_or_more

Operators

(none)

1.139 linear_algebra

object

1.139.1 linear_algebra

Linear algebra predicates for numeric vectors and matrices implemented without dependencies on machine learning libraries.

Availability:
    logtalk_load(linear_algebra(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:
    static, context_switching_calls

Implements:
    public linear_algebra_protocol
Uses:
    list

numberlist
type

Remarks:
(none)

Inherited public predicates:
add_matrices/3 add_scaled_outer_product/4 add_scaled_vector/4 add_vectors/3
backward_substitution/3 basis_vector/3 center_gram_matrix/4 center_gram_vector/4
cholesky_decomposition/2 covariance_matrix/2 determinant/2 diagonal_matrix/2
diagonal_matrix/3 diagonal_matrix/4 difference_norm/3 dot_product/3 euclidean_norm/2
first_significant_component/2 first_significant_component/3 forward_substitution/3
frobenius_norm/2 gram_matrix/2 identity_matrix/2 inverse_matrix/2 invert_from_cholesky/2
least_squares/3 lower_triangular_part/2 lower_triangular_part/3 matrix_column/3
matrix_column_means/2 matrix_diagonal/2 matrix_diagonal/3 matrix_matrix_product/3
matrix_rank/2 matrix_rank/3 matrix_row/3 matrix_row_means/2 matrix_trace/2
matrix_value/4 matrix_vector_product/3 new_matrix/4 new_vector/3 new_vector_like/2
normalize_vector/2 normalize_vector/3 null_space/2 null_space/3 outer_product/3
pseudo_inverse/2 pseudo_inverse/3 qr_decomposition/3 scale_matrix/3 scale_vector/3
shift_matrix_diagonal/3 solve_cholesky/3 solve_linear_system/3 solve_linear_systems/3
solve_lower_triangular_matrix/3 solve_upper_triangular/3 solve_upper_triangular_matrix/3
stabilize_vector_sign/2 stabilize_vector_sign/3 subtract_matrices/3 subtract_vectors/3
symmetric_eigen/3 symmetric_eigen/4 symmetric_eigen/5 transpose_matrix/2
upper_triangular_part/2 upper_triangular_part/3 vector_norm/3 vector_value/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.139.2 linear_algebra_protocol

Linear algebra protocol.

Availability:
logtalk_load(linear_algebra(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - new_vector/3
  - new_vector_like/2
  - add_vectors/3
  - subtract_vectors/3
  - add_scaled_vector/4

- scale_vector/3
- dot_product/3
- euclidean_norm/2
- vector_norm/3
- basis_vector/3
- new_matrix/4
- identity_matrix/2
- matrix_vector_product/3
- matrix_matrix_product/3
- gram_matrix/2
- matrix_row_means/2
- matrix_column_means/2
- center_gram_matrix/4
- center_gram_vector/4
- outer_product/3
- add_matrices/3
- subtract_matrices/3
- scale_matrix/3
- frobenius_norm/2
- transpose_matrix/2
- vector_value/3
- matrix_row/3
- matrix_column/3
- matrix_value/4
- matrix_diagonal/2
- matrix_diagonal/3
- diagonal_matrix/2
- diagonal_matrix/3
- diagonal_matrix/4
- matrix_trace/2
- shift_matrix_diagonal/3
- upper_triangular_part/2
- upper_triangular_part/3
- lower_triangular_part/2
- lower_triangular_part/3

  - solve_linear_system/3
  - solve_linear_systems/3
  - determinant/2
  - inverse_matrix/2
  - qr_decomposition/3
  - least_squares/3
  - matrix_rank/2
  - matrix_rank/3
  - symmetric_eigen/3
  - symmetric_eigen/4
  - symmetric_eigen/5
  - pseudo_inverse/2
  - pseudo_inverse/3
  - null_space/2
  - null_space/3
  - normalize_vector/2
  - normalize_vector/3
  - difference_norm/3
  - stabilize_vector_sign/2
  - stabilize_vector_sign/3
  - first_significant_component/2
  - first_significant_component/3
  - add_scaled_outer_product/4
  - covariance_matrix/2
  - cholesky_decomposition/2
  - solve_cholesky/3
  - forward_substitution/3
  - solve_lower_triangular_matrix/3
  - solve_upper_triangular/3
  - solve_upper_triangular_matrix/3
  - backward_substitution/3
  - invert_from_cholesky/2
- Protected predicates
- Private predicates
- Operators

Public predicates

new_vector/3

Constructs a numeric vector with the requested length, filled with the given value.

Compilation flags:
    static

Template:
    new_vector(Count,Value,Vector)
Mode and number of proofs:
    new_vector(+integer,+number,-list(number)) - one

Exceptions:
    Count is a variable:
        instantiation_error
    Value is a variable:
        instantiation_error
    Count is neither a variable nor an integer:
        type_error(integer,Count)
    Value is neither a variable nor a number:
        type_error(number,Value)
    Count is an integer that is less than zero:
        domain_error(non_negative_integer,Count)

---

new_vector_like/2

Constructs a numeric zero vector matching the length of the reference list.

Compilation flags:
    static

Template:
    new_vector_like(Reference,Zeroes)
Mode and number of proofs:
    new_vector_like(+list,-list(number)) - one

---

add_vectors/3

Adds two numeric vectors element-wise.

Compilation flags:
    static

Template:
    add_vectors(Vector1,Vector2,Vector)
Mode and number of proofs:
    add_vectors(+list(number),+list(number),-list(number)) - one

---

subtract_vectors/3

Subtracts the second numeric vector from the first element-wise.

Compilation flags:
    static

Template:
    subtract_vectors(Vector1,Vector2,Vector)
Mode and number of proofs:
    subtract_vectors(+list(number),+list(number),-list(number)) - one

---

add_scaled_vector/4

Adds a scaled numeric vector to another numeric vector.

Compilation flags:
    static

Template:
    add_scaled_vector(Vector,Scale,Vector0,UpdatedVector)
Mode and number of proofs:
    add_scaled_vector(+list(number),+number,+list(number),-list(number)) - one

Exceptions:
    Scale is a variable:
        instantiation_error
    Scale is neither a variable nor a number:
        type_error(number,Scale)

---

scale_vector/3

Scales a numeric vector by the given factor.

Compilation flags:
    static

Template:
    scale_vector(Vector,Scale,ScaledVector)
Mode and number of proofs:
    scale_vector(+list(number),+number,-list(number)) - one

Exceptions:
    Scale is a variable:
        instantiation_error
    Scale is neither a variable nor a number:
        type_error(number,Scale)

---

dot_product/3

Computes the dot product of two numeric vectors.

Compilation flags:
    static

Template:
    dot_product(Vector1,Vector2,Product)
Mode and number of proofs:
    dot_product(+list(number),+list(number),-number) - one

---

euclidean_norm/2

Computes the Euclidean norm of a numeric vector.

Compilation flags:
    static

Template:
    euclidean_norm(Vector,Norm)
Mode and number of proofs:
    euclidean_norm(+list(number),-float) - one

---

vector_norm/3

Computes the norm of a numeric vector for a positive numeric order or the atoms inf and infinity.

Compilation flags:
    static

Template:
    vector_norm(Vector,Order,Norm)
Mode and number of proofs:
    vector_norm(+list(number),@term,-number) - one_or_error

Exceptions:
    Order is a variable:
        instantiation_error
    Order is neither a variable, a number, nor one of the atoms inf and infinity:
        type_error(number,Order)
    Order is a number but not a valid positive_number_or_infinity:
        domain_error(positive_number_or_infinity,Order)

---

basis_vector/3

Constructs a numeric basis vector with value 1.0 at the given one-based index and 0.0 elsewhere.

Compilation flags:
    static

Template:
    basis_vector(Size,Index,Vector)
Mode and number of proofs:
    basis_vector(+integer,+integer,-list(number)) - one

Exceptions:
    Size is a variable:
        instantiation_error
    Index is a variable:
        instantiation_error
    Size is neither a variable nor an integer:
        type_error(integer,Size)
    Index is neither a variable nor an integer:
        type_error(integer,Index)
    Size is an integer that is less than one:
        domain_error(positive_integer,Size)
    Index is an integer that is less than one:
        domain_error(positive_integer,Index)

---

new_matrix/4

Constructs a numeric matrix with the requested row and column counts, filled with the given value.

Compilation flags:
    static

Template:
    new_matrix(Rows,Columns,Value,Matrix)
Mode and number of proofs:
    new_matrix(+integer,+integer,+number,-list(list(number))) - one

Exceptions:
    Rows is a variable:

instantiation_error
Columns is a variable:
instantiation_error
Value is a variable:
instantiation_error
Rows is neither a variable nor an integer:
type_error(integer,Rows)
Columns is neither a variable nor an integer:
type_error(integer,Columns)
Value is neither a variable nor a number:
type_error(number,Value)
Rows is an integer that is less than zero:
domain_error(non_negative_integer,Rows)
Columns is an integer that is less than zero:
domain_error(non_negative_integer,Columns)

---

identity_matrix/2

Constructs a square identity matrix with the requested size.

Compilation flags:
static

Template:
identity_matrix(Size,Matrix)
Mode and number of proofs:
identity_matrix(+integer,-list(list(number))) - one

Exceptions:
Size is a variable:
instantiation_error
Size is neither a variable nor an integer:
type_error(integer,Size)
Size is an integer that is less than zero:
domain_error(non_negative_integer,Size)

---

matrix_vector_product/3

Computes the matrix-vector product for a numeric matrix and vector, returning the list of row-wise dot products between each matrix row and the vector.

Compilation flags:
    static

Template:
    matrix_vector_product(Matrix,Vector,Product)
Mode and number of proofs:
    matrix_vector_product(+list(list(number)),+list(number),-list(number)) - one

---

matrix_matrix_product/3

Computes the matrix-matrix product for two numeric matrices.

Compilation flags:
    static

Template:
    matrix_matrix_product(Matrix1,Matrix2,Product)
Mode and number of proofs:
    matrix_matrix_product(+list(list(number)),+list(list(number)),-list(list(number))) - one

---

gram_matrix/2

Computes the row Gram matrix of a numeric row matrix by taking all row-wise dot products.

Compilation flags:
    static

Template:
    gram_matrix(Rows,GramMatrix)
Mode and number of proofs:
    gram_matrix(+list(list(number)),-list(list(number))) - one

---

matrix_row_means/2

Computes the arithmetic mean of each row in a numeric matrix.

Compilation flags:
    static

Template:
    matrix_row_means(Matrix,Means)
Mode and number of proofs:
    matrix_row_means(+list(list(number)),-list(number)) - one

Exceptions:
    Matrix contains an empty row Row:
        domain_error(minimum_number_of_values(1),Row)

---

matrix_column_means/2

Computes the arithmetic mean of each column in a numeric matrix.

Compilation flags:
    static

Template:
    matrix_column_means(Matrix,Means)
Mode and number of proofs:
    matrix_column_means(+list(list(number)),-list(number)) - one

---

center_gram_matrix/4

Double-centers a numeric Gram matrix and also returns its row means and total mean.

Compilation flags:
    static

Template:
    center_gram_matrix(GramMatrix,CenteredGramMatrix,RowMeans,TotalMean)
Mode and number of proofs:
    center_gram_matrix(+list(list(number)),-list(list(number)),-list(number),-number) - one

Exceptions:
    GramMatrix is empty:
        domain_error(minimum_number_of_values(1),GramMatrix)
    GramMatrix contains an empty row Row:
        domain_error(minimum_number_of_values(1),Row)

---

center_gram_vector/4

Centers a numeric Gram vector using training row means and total mean.

Compilation flags:
    static

Template:
    center_gram_vector(GramVector,RowMeans,TotalMean,CenteredGramVector)
Mode and number of proofs:
    center_gram_vector(+list(number),+list(number),+number,-list(number)) - one

Exceptions:
    GramVector is empty:
        domain_error(minimum_number_of_values(1),GramVector)

---

outer_product/3

Computes the outer product of two numeric vectors.

Compilation flags:
    static

Template:
    outer_product(Vector1,Vector2,Matrix)
Mode and number of proofs:
    outer_product(+list(number),+list(number),-list(list(number))) - one

---

add_matrices/3

Adds two numeric matrices element-wise.

Compilation flags:
    static

Template:
    add_matrices(Matrix1,Matrix2,Matrix)
Mode and number of proofs:
    add_matrices(+list(list(number)),+list(list(number)),-list(list(number))) - one

---

subtract_matrices/3

Subtracts the second numeric matrix from the first element-wise.

Compilation flags:
    static

Template:
    subtract_matrices(Matrix1,Matrix2,Matrix)
Mode and number of proofs:
    subtract_matrices(+list(list(number)),+list(list(number)),-list(list(number))) - one

---

scale_matrix/3

Scales each element of a numeric matrix by the given factor.

Compilation flags:
    static

Template:
    scale_matrix(Matrix,Scale,ScaledMatrix)
Mode and number of proofs:
    scale_matrix(+list(list(number)),+number,-list(list(number))) - one

---

frobenius_norm/2

Computes the Frobenius norm of a numeric matrix.

Compilation flags:
    static

Template:
    frobenius_norm(Matrix,Norm)
Mode and number of proofs:
    frobenius_norm(+list(list(number)),-float) - one_or_error

Exceptions:
    Matrix is not a rectangular matrix:
        domain_error(rectangular_matrix,Matrix)

---

transpose_matrix/2

Transposes a numeric matrix represented as a list of row lists.

Compilation flags:
    static

Template:

transpose_matrix(Matrix,Transpose)
Mode and number of proofs:
transpose_matrix(+list(list(number)),-list(list(number))) - one

---

vector_value/3

Looks up a numeric vector element using a one-based index.

Compilation flags:
static

Template:
vector_value(Vector,Index,Value)
Mode and number of proofs:
vector_value(+list(number),+integer,-number) - one

Exceptions:
Index is a variable:
instantiation_error
Index is neither a variable nor an integer:
type_error(integer,Index)
Index is an integer that is less than one:
domain_error(positive_integer,Index)

---

matrix_row/3

Looks up a numeric matrix row using a one-based row index.

Compilation flags:
static

Template:
matrix_row(Matrix,RowIndex,Row)
Mode and number of proofs:
matrix_row(+list(list(number)),+integer,-list(number)) - one

Exceptions:

RowIndex is a variable:
instantiation_error
RowIndex is neither a variable nor an integer:
type_error(integer,RowIndex)
RowIndex is an integer that is less than one:
domain_error(positive_integer,RowIndex)

---

matrix_column/3

Looks up a numeric matrix column using a one-based column index.

Compilation flags:
static

Template:
matrix_column(Matrix,ColumnIndex,Column)
Mode and number of proofs:
matrix_column(+list(list(number)),+integer,-list(number)) - one

Exceptions:
ColumnIndex is a variable:
instantiation_error
ColumnIndex is neither a variable nor an integer:
type_error(integer,ColumnIndex)
ColumnIndex is an integer that is less than one:
domain_error(positive_integer,ColumnIndex)

---

matrix_value/4

Looks up a numeric matrix element using one-based row and column indices.

Compilation flags:
static

Template:
matrix_value(Matrix,RowIndex,ColumnIndex,Value)
Mode and number of proofs:

matrix_value(+list(list(number)),+integer,+integer,-number) - one

Exceptions:
RowIndex is a variable:
instantiation_error
ColumnIndex is a variable:
instantiation_error
RowIndex is neither a variable nor an integer:
type_error(integer,RowIndex)
ColumnIndex is neither a variable nor an integer:
type_error(integer,ColumnIndex)
RowIndex is an integer that is less than one:
domain_error(positive_integer,RowIndex)
ColumnIndex is an integer that is less than one:
domain_error(positive_integer,ColumnIndex)

---

matrix_diagonal/2

Extracts the main diagonal of a numeric matrix as a vector.

Compilation flags:
static

Template:
matrix_diagonal(Matrix,Diagonal)
Mode and number of proofs:
matrix_diagonal(+list(list(number)),-list(number)) - one

Exceptions:
Matrix is not a rectangular matrix:
domain_error(rectangular_matrix,Matrix)

---

matrix_diagonal/3

Extracts the diagonal of a numeric matrix at the given offset, where positive offsets select superdiagonals and negative offsets select subdiagonals.

Compilation flags:
    static

Template:
    matrix_diagonal(Matrix,Offset,Diagonal)
Mode and number of proofs:
    matrix_diagonal(+list(list(number)),+integer,-list(number)) - one_or_error

Exceptions:
    Offset is a variable:
        instantiation_error
    Offset is neither a variable nor an integer:
        type_error(integer,Offset)
    Matrix is not a rectangular matrix:
        domain_error(rectangular_matrix,Matrix)

---

diagonal_matrix/2

Constructs a square diagonal matrix from a numeric vector of diagonal entries.

Compilation flags:
    static

Template:
    diagonal_matrix(Diagonal,Matrix)
Mode and number of proofs:
    diagonal_matrix(+list(number),-list(list(number))) - one

---

diagonal_matrix/3

Constructs the minimal square numeric matrix whose diagonal at the given offset matches the input vector.

Compilation flags:
    static

Template:
    diagonal_matrix(Diagonal,Offset,Matrix)
Mode and number of proofs:
    diagonal_matrix(+list(number),+integer,-list(list(number))) - one_or_error

Exceptions:
    Offset is a variable:
        instantiation_error
    Offset is neither a variable nor an integer:
        type_error(integer,Offset)

---

diagonal_matrix/4

Constructs a square numeric matrix of the requested size whose diagonal at the given offset matches the input vector.

Compilation flags:
    static

Template:
    diagonal_matrix(Diagonal,Offset,Size,Matrix)
Mode and number of proofs:
    diagonal_matrix(+list(number),+integer,+integer,-list(list(number))) - one_or_error

Exceptions:
    Offset is a variable:
        instantiation_error
    Offset is neither a variable nor an integer:
        type_error(integer,Offset)
    Size is a variable:
        instantiation_error
    Size is neither a variable nor an integer:
        type_error(integer,Size)

Size is an integer that is less than zero:
domain_error(non_negative_integer,Size)
Size is smaller than the minimum size required by Diagonal and Offset:
domain_error(minimum_matrix_size(A),Size)

---

matrix_trace/2

Computes the trace of a numeric matrix by summing its main diagonal entries.

Compilation flags:
static

Template:
matrix_trace(Matrix,Trace)
Mode and number of proofs:
matrix_trace(+list(list(number)),-number) - one

Exceptions:
Matrix is not a rectangular matrix:
domain_error(rectangular_matrix,Matrix)

---

shift_matrix_diagonal/3

Shifts the main diagonal of a numeric matrix by the given value.

Compilation flags:
static

Template:
shift_matrix_diagonal(Matrix,Shift,ShiftedMatrix)
Mode and number of proofs:
shift_matrix_diagonal(+list(list(number)),+number,-list(list(number))) - one

Exceptions:
Shift is a variable:
instantiation_error
Shift is neither a variable nor a number:

type_error(number,Shift)

---

upper_triangular_part/2

Returns the upper-triangular part of a numeric matrix including the main diagonal.

Compilation flags:
static

Template:
upper_triangular_part(Matrix,UpperTriangular)
Mode and number of proofs:
upper_triangular_part(+list(list(number)),-list(list(number))) - one_or_error

Exceptions:
Matrix is not a rectangular matrix:
domain_error(rectangular_matrix,Matrix)

---

upper_triangular_part/3

Returns the upper-triangular part of a numeric matrix using the given diagonal offset, where positive offsets exclude leading diagonals and negative offsets include additional subdiagonals.

Compilation flags:
static

Template:
upper_triangular_part(Matrix,Offset,UpperTriangular)
Mode and number of proofs:
upper_triangular_part(+list(list(number)),+integer,-list(list(number))) - one_or_error

Exceptions:
Offset is a variable:
instantiation_error
Offset is neither a variable nor an integer:
type_error(integer,Offset)
Matrix is not a rectangular matrix:

domain_error(rectangular_matrix,Matrix)

---

lower_triangular_part/2

Returns the lower-triangular part of a numeric matrix including the main diagonal.

Compilation flags:
static

Template:
lower_triangular_part(Matrix,LowerTriangular)
Mode and number of proofs:
lower_triangular_part(+list(list(number)),-list(list(number))) - one_or_error

Exceptions:
Matrix is not a rectangular matrix:
domain_error(rectangular_matrix,Matrix)

---

lower_triangular_part/3

Returns the lower-triangular part of a numeric matrix using the given diagonal offset, where positive offsets include additional superdiagonals and negative offsets exclude leading diagonals below the main diagonal.

Compilation flags:
static

Template:
lower_triangular_part(Matrix,Offset,LowerTriangular)
Mode and number of proofs:
lower_triangular_part(+list(list(number)),+integer,-list(list(number))) - one_or_error

Exceptions:
Offset is a variable:
instantiation_error
Offset is neither a variable nor an integer:
type_error(integer,Offset)
Matrix is not a rectangular matrix:

domain_error(rectangular_matrix,Matrix)

---

solve_linear_system/3

Solves a non-singular square numeric linear system with a vector right-hand side using pivoted elimination.

Compilation flags:
static

Template:
solve_linear_system(Matrix,Values,Solution)
Mode and number of proofs:
solve_linear_system(+list(list(number)),+list(number),-list(float)) - one_or_error

Exceptions:
Matrix is not a square matrix:
domain_error(square_matrix,Matrix)
Values does not have the same length as the size of Matrix:
domain_error(vector_length(A),Values)
Matrix is singular:
evaluation_error(zero_divisor)

---

solve_linear_systems/3

Solves a non-singular square numeric linear system with a matrix right-hand side using pivoted elimination.

Compilation flags:
static

Template:
solve_linear_systems(Matrix,RightHandSides,Solutions)
Mode and number of proofs:
solve_linear_systems(+list(list(number)),+list(list(number)),-list(list(float))) - one_or_error

Exceptions:
Matrix is not a square matrix:
domain_error(square_matrix,Matrix)

RightHandSides does not have the same number of rows as the size of Matrix:
domain_error(matrix_row_count(A),RightHandSides)
Matrix is singular:
evaluation_error(zero_divisor)

---

determinant/2

Computes the determinant of a square numeric matrix.

Compilation flags:
static

Template:
determinant(Matrix,Determinant)
Mode and number of proofs:
determinant(+list(list(number)),-number) - one_or_error

Exceptions:
Matrix is not a square matrix:
domain_error(square_matrix,Matrix)

---

inverse_matrix/2

Computes the inverse of a non-singular square numeric matrix by solving one linear system per identity column.

Compilation flags:
static

Template:
inverse_matrix(Matrix,Inverse)
Mode and number of proofs:
inverse_matrix(+list(list(number)),-list(list(float))) - one_or_error

Exceptions:
Matrix is not a square matrix:
domain_error(square_matrix,Matrix)

Matrix is singular:
evaluation_error(zero_divisor)

---

qr_decomposition/3

Computes a thin QR decomposition of a rectangular numeric matrix, returning orthogonal columns and an upper-trapezoidal factor.

Compilation flags:
static

Template:
qr_decomposition(Matrix,Orthogonal,UpperTriangular)
Mode and number of proofs:
qr_decomposition(+list(list(number)),-list(list(float)),-list(list(float))) - one_or_error

Exceptions:
Matrix is not a rectangular matrix:
domain_error(rectangular_matrix,Matrix)

---

least_squares/3

Computes a least-squares solution for a rectangular numeric linear system using a pivoted orthogonal solve without forming normal equations.

Compilation flags:
static

Template:
least_squares(Matrix,Values,Solution)
Mode and number of proofs:
least_squares(+list(list(number)),+list(number),-list(float)) - one_or_error

Exceptions:
Matrix is not a rectangular matrix:
domain_error(rectangular_matrix,Matrix)
Values does not have the same length as the number of rows of Matrix:

domain_error(vector_length(A),Values)

---

matrix_rank/2

Estimates the rank of a numeric matrix using the default library numerical tolerance of 1.0e-12.

Compilation flags:
static

Template:
matrix_rank(Matrix,Rank)
Mode and number of proofs:
matrix_rank(+list(list(number)),-integer) - one_or_error

Exceptions:
Matrix is not a rectangular matrix:
domain_error(rectangular_matrix,Matrix)

---

matrix_rank/3

Estimates the rank of a numeric matrix using the given non-negative numerical tolerance.

Compilation flags:
static

Template:
matrix_rank(Matrix,Tolerance,Rank)
Mode and number of proofs:
matrix_rank(+list(list(number)),+number,-integer) - one_or_error

Exceptions:
Tolerance is a variable:
instantiation_error
Tolerance is neither a variable nor a number:
type_error(number,Tolerance)
Tolerance is a number but not a non-negative number:
domain_error(non_negative_number,Tolerance)

Matrix is not a rectangular matrix:
domain_error(rectangular_matrix,Matrix)

---

symmetric_eigen/3

Computes an orthonormal eigendecomposition of a real symmetric dense matrix using the default library numerical tolerance of 1.0e-12.

Compilation flags:
static

Template:
symmetric_eigen(Matrix,Eigenvectors,Eigenvalues)
Mode and number of proofs:
symmetric_eigen(+list(list(number)),-list(list(float)),-list(float)) - one_or_error

Exceptions:
Matrix is not a square matrix:
domain_error(square_matrix,Matrix)
Matrix is not symmetric within the default numerical tolerance:
domain_error(symmetric_matrix,Matrix)

---

symmetric_eigen/4

Computes an orthonormal eigendecomposition of a real symmetric dense matrix using the given non-negative numerical tolerance.

Compilation flags:
static

Template:
symmetric_eigen(Matrix,Tolerance,Eigenvectors,Eigenvalues)
Mode and number of proofs:
symmetric_eigen(+list(list(number)),+number,-list(list(float)),-list(float)) - one_or_error

Exceptions:
Tolerance is a variable:

instantiation_error
Tolerance is neither a variable nor a number:
type_error(number,Tolerance)
Tolerance is a number but not a non-negative number:
domain_error(non_negative_number,Tolerance)
Matrix is not a square matrix:
domain_error(square_matrix,Matrix)
Matrix is not symmetric within Tolerance:
domain_error(symmetric_matrix,Matrix)

---

symmetric_eigen/5

Computes an orthonormal eigendecomposition of a real symmetric dense matrix using the given non-negative numerical tolerance and positive iteration bound.

Compilation flags:
static

Template:
symmetric_eigen(Matrix,Tolerance,MaximumIterations,Eigenvectors,Eigenvalues)
Mode and number of proofs:
symmetric_eigen(+list(list(number)),+number,+integer,-list(list(float)),-list(float)) - one_or_error

Exceptions:
Tolerance is a variable:
instantiation_error
Tolerance is neither a variable nor a number:
type_error(number,Tolerance)
Tolerance is a number but not a non-negative number:
domain_error(non_negative_number,Tolerance)
MaximumIterations is a variable:
instantiation_error
MaximumIterations is neither a variable nor an integer:
type_error(integer,MaximumIterations)
MaximumIterations is an integer that is less than one:
domain_error(positive_integer,MaximumIterations)
Matrix is not a square matrix:
domain_error(square_matrix,Matrix)
Matrix is not symmetric within Tolerance:
domain_error(symmetric_matrix,Matrix)

---

pseudo_inverse/2

Computes the Moore-Penrose pseudo-inverse of a rectangular numeric matrix using the default library numerical tolerance of 1.0e-12.

Compilation flags:
    static

Template:
    pseudo_inverse(Matrix,PseudoInverse)
Mode and number of proofs:
    pseudo_inverse(+list(list(number)),-list(list(float))) - one_or_error

Exceptions:
    Matrix is not a rectangular matrix:
        domain_error(rectangular_matrix,Matrix)

---

pseudo_inverse/3

Computes the Moore-Penrose pseudo-inverse of a rectangular numeric matrix using the given non-negative numerical tolerance.

Compilation flags:
    static

Template:
    pseudo_inverse(Matrix,Tolerance,PseudoInverse)
Mode and number of proofs:
    pseudo_inverse(+list(list(number)),+number,-list(list(float))) - one_or_error

Exceptions:
    Tolerance is a variable:
        instantiation_error
    Tolerance is neither a variable nor a number:
        type_error(number,Tolerance)
    Tolerance is a number but not a non-negative number:
        domain_error(non_negative_number,Tolerance)
    Matrix is not a rectangular matrix:
        domain_error(rectangular_matrix,Matrix)

---

null_space/2

Computes an orthonormal basis for the right null space of a rectangular numeric matrix using the default library numerical tolerance of 1.0e-12.

Compilation flags:
    static

Template:
    null_space(Matrix,Basis)
Mode and number of proofs:
    null_space(+list(list(number)),-list(list(float))) - one_or_error

Exceptions:
    Matrix is not a rectangular matrix:
        domain_error(rectangular_matrix,Matrix)

---

null_space/3

Computes an orthonormal basis for the right null space of a rectangular numeric matrix using the given non-negative numerical tolerance.

Compilation flags:
    static

Template:
    null_space(Matrix,Tolerance,Basis)
Mode and number of proofs:
    null_space(+list(list(number)),+number,-list(list(float))) - one_or_error

Exceptions:
    Tolerance is a variable:
        instantiation_error
    Tolerance is neither a variable nor a number:
        type_error(number,Tolerance)
    Tolerance is a number but not a non-negative number:
        domain_error(non_negative_number,Tolerance)

Matrix is not a rectangular matrix:
domain_error(rectangular_matrix,Matrix)

---

normalize_vector/2

Normalizes a numeric vector to unit length when its norm is above the default library numerical tolerance of 1.0e-12.

Compilation flags:
static

Template:
normalize_vector(Vector,NormalizedVector)
Mode and number of proofs:
normalize_vector(+list(number),-list(number)) - one

---

normalize_vector/3

Normalizes a numeric vector to unit length when its norm is above the given non-negative numerical tolerance.

Compilation flags:
static

Template:
normalize_vector(Vector,Tolerance,NormalizedVector)
Mode and number of proofs:
normalize_vector(+list(number),+number,-list(number)) - one_or_error

Exceptions:
Tolerance is a variable:
instantiation_error
Tolerance is neither a variable nor a number:
type_error(number,Tolerance)
Tolerance is a number but not a non-negative number:
domain_error(non_negative_number,Tolerance)

---

difference_norm/3

Computes the Euclidean norm of the difference between two numeric vectors.

Compilation flags:
    static

Template:
    difference_norm(Vector1,Vector2,Norm)
Mode and number of proofs:
    difference_norm(+list(number),+list(number),-float) - one

---

stabilize_vector_sign/2

Normalizes a vector sign convention by flipping vectors whose first significant component is negative.

Compilation flags:
    static

Template:
    stabilize_vector_sign(Vector,StableVector)
Mode and number of proofs:
    stabilize_vector_sign(+list(number),-list(number)) - one

---

stabilize_vector_sign/3

Normalizes a vector sign convention by flipping vectors whose first significant component under the given non-negative numerical tolerance is negative.

Compilation flags:
    static

Template:
    stabilize_vector_sign(Vector,Tolerance,StableVector)
Mode and number of proofs:
    stabilize_vector_sign(+list(number),+number,-list(number)) - one_or_error

Exceptions:
Tolerance is a variable:
instantiation_error
Tolerance is neither a variable nor a number:
type_error(number,Tolerance)
Tolerance is a number but not a non-negative number:
domain_error(non_negative_number,Tolerance)

---

first_significant_component/2

Returns the first component whose absolute value exceeds the default library numerical tolerance of 1.0e-12, defaulting to zero when no such component exists.

Compilation flags:
static

Template:
first_significant_component(Vector,First)
Mode and number of proofs:
first_significant_component(+list(number),-number) - one

---

first_significant_component/3

Returns the first component whose absolute value exceeds the given non-negative numerical tolerance, defaulting to zero when no such component exists.

Compilation flags:
static

Template:
first_significant_component(Vector,Tolerance,First)
Mode and number of proofs:
first_significant_component(+list(number),+number,-number) - one_or_error

Exceptions:
Tolerance is a variable:
instantiation_error
Tolerance is neither a variable nor a number:

type_error(number,Tolerance)
Tolerance is a number but not a non-negative number:
domain_error(non_negative_number,Tolerance)

---

add_scaled_outer_product/4

Adds a scaled outer product of a vector with itself to an accumulated matrix.

Compilation flags:
static

Template:
add_scaled_outer_product(Vector,Scale,Matrix0,Matrix)
Mode and number of proofs:
add_scaled_outer_product(+list(number),+number,+list(list(number)),-list(list(number))) - one

Exceptions:
Scale is a variable:
instantiation_error
Scale is neither a variable nor a number:
type_error(number,Scale)

---

covariance_matrix/2

Computes the sample covariance matrix from a list of centered numeric row vectors.

Compilation flags:
static

Template:
covariance_matrix(Rows,CovarianceMatrix)
Mode and number of proofs:
covariance_matrix(+list(list(number)),-list(list(number))) - one_or_error

Exceptions:
Rows contains fewer than two rows:
domain_error(minimum_number_of_rows(2),Rows)

---

cholesky_decomposition/2

Computes the lower-triangular Cholesky factor of a positive definite numeric matrix.

Compilation flags:
    static

Template:
    cholesky_decomposition(Matrix,CholeskyFactor)
Mode and number of proofs:
    cholesky_decomposition(+list(list(number)),-list(list(float))) - one_or_error

Exceptions:
    Matrix is not numerically positive definite:
        domain_error(positive_definite_matrix,Matrix)

---

solve_cholesky/3

Solves a positive definite linear system from a lower-triangular Cholesky factor and a right-hand-side vector.

Compilation flags:
    static

Template:
    solve_cholesky(CholeskyFactor,Values,Solution)
Mode and number of proofs:
    solve_cholesky(+list(list(number)),+list(number),-list(float)) - one

---

forward_substitution/3

Solves a lower-triangular system by forward substitution.

Compilation flags:
    static

Template:
    forward_substitution(LowerTriangular,Values,Solution)
Mode and number of proofs:
    forward_substitution(+list(list(number)),+list(number),-list(float)) - one

---

solve_lower_triangular_matrix/3

Solves a lower-triangular system with a matrix right-hand side by forward substitution on each column.

Compilation flags:
    static

Template:
    solve_lower_triangular_matrix(LowerTriangular,Matrix,Solution)
Mode and number of proofs:
    solve_lower_triangular_matrix(+list(list(number)),+list(list(number)),-list(list(float))) - one

---

solve_upper_triangular/3

Solves an upper-triangular system by backward substitution.

Compilation flags:
    static

Template:
    solve_upper_triangular(UpperTriangular,Values,Solution)
Mode and number of proofs:
    solve_upper_triangular(+list(list(number)),+list(number),-list(float)) - one

---

solve_upper_triangular_matrix/3

Solves an upper-triangular system with a matrix right-hand side by backward substitution on each column.

Compilation flags:
    static

Template:
    solve_upper_triangular_matrix(UpperTriangular,Matrix,Solution)
Mode and number of proofs:
    solve_upper_triangular_matrix(+list(list(number)),+list(list(number)),-list(list(float))) - one

---

backward_substitution/3

Solves the transposed upper-triangular system induced by a lower-triangular Cholesky factor using backward substitution.

Compilation flags:
    static

Template:
    backward_substitution(CholeskyFactor,Values,Solution)
Mode and number of proofs:
    backward_substitution(+list(list(number)),+list(number),-list(float)) - one

---

invert_from_cholesky/2

Computes a matrix inverse from a Cholesky factor by solving one linear system per basis column.

Compilation flags:
    static

Template:
    invert_from_cholesky(CholeskyFactor,Inverse)
Mode and number of proofs:
    invert_from_cholesky(+list(list(number)),-list(list(float))) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.140 linear_regression

object

1.140.1 linear_regression

Linear regression regressor supporting continuous and mixed-feature datasets using a direct ordinary least-squares solve. Learns from a dataset object implementing the regression_dataset_protocol protocol and returns a regressor term that can be used for prediction and exported as predicate clauses.

Availability:
    logtalk_load(linear_regression(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:
    static, context_switching_calls

Imports:
    public regressor_common
Uses:
    format
    list
    numberlist
    type

Remarks:

(none)

Inherited public predicates:

check_option/1 check_options/1 check_regressor/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 print_regressor/1 regressor_options/2 valid_option/1 valid_options/1 valid_regressor/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

ridge_regression, knn_regression, regression_tree, random_forest_regression, gradient_boosting_regression

1.141 linear_svm_classifier

object

1.141.1 linear_svm_classifier

Linear support vector machine classifier supporting binary and multiclass classification using one-vs-rest margin models. Learns from a dataset object implementing the dataset_protocol protocol and returns a classifier term that can be used for prediction and exported as predicate clauses.

Availability:
logtalk_load(linear_svm_classifier(loader))

Author: Paulo Moura
Version: 2:0:0
Date: 2026-05-11

Compilation flags:
static, context_switching_calls

Imports:
public classifier_common
Uses:
format
linear_algebra
list
numberlist
pairs
population
type

Remarks:
(none)

Inherited public predicates:
check_classifier/1 check_option/1 check_options/1 classifier_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 print_classifier/1 valid_classifier/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

dataset_protocol, logistic_regression_classifier, c45_classifier, knn_classifier, naive_bayes_classifier, nearest_centroid_classifier, random_forest_classifier, adaptive_boosting_classifier

1.142 linter_reporter

object

1.142.1 linter_reporter

Intercepts compiler linter warnings and caches them as machine-readable diagnostics.

Availability:
logtalk_load(linter_reporter(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-30

Compilation flags:
static, context_switching_calls

Imports:
public tool_diagnostics_common
public tutor_explanations
public options
Provides:
logtalk::message_hook/4
Uses:
list
logtalk
os
term_io
type
user

Remarks:

- Usage: Load this tool before compiling code to be checked by the built-in linter. Call enable/0-1 before compiling code, disable/0 when finished collecting warnings, and then query the cached warnings using either the legacy warning predicates or the diagnostics protocol predicates. The standalone sarif tool can generate SARIF reports by querying these diagnostics.
- Diagnostics targets: The diagnostics predicates accept the targets all, entity(Entity), file(File), directory(Directory), rdirectory(Directory), library(Library), and rlibrary(Library). These targets simply filter the cached diagnostics collected in the current warning collection session.

Inherited public predicates:
check_option/1 check_options/1 default_option/1 default_options/1 diagnostic/2 diagnostic/3 diagnostic_rule/5 diagnostic_rules/1 diagnostic_target/1 diagnostics/2 diagnostics/3 diagnostics_preflight/2 diagnostics_preflight/3 diagnostics_summary/2 diagnostics_summary/3 diagnostics_tool/5 explain//1 option/2 option/3 valid_option/1 valid_options/1

- Public predicates
  - enable/0
  - enable/1
  - disable/0
  - reset/0
  - warning/1
  - warnings/1
  - summary/1

- Protected predicates
- Private predicates
  - enabled_/0
  - warning_sequence_/1
  - recorded_warning_/4
  - collection_options_/1
- Operators

Public predicates

enable/0

Enables warning collection and starts a fresh warning collection session using the default options.

Compilation flags:
    static

Mode and number of proofs:
    enable - one

---

enable/1

Enables warning collection and starts a fresh warning collection session using the given options.

Compilation flags:
    static

Template:
    enable(Options)
Mode and number of proofs:
    enable(+list(compound)) - one_or_error

---

disable/0

Disables warning collection while preserving the cached warnings for later querying.

Compilation flags:
    static

Mode and number of proofs:
    disable - one

---

reset/0

Clears all cached warnings collected in the current session.

Compilation flags:
    static

Mode and number of proofs:
    reset - one

---

warning/1

Enumerates normalized linter warnings collected in the current session.

Compilation flags:
    static

Template:
    warning(Warning)
Mode and number of proofs:
    warning(-compound) - zero_or_more

---

warnings/1

Returns the collected normalized linter warnings.

Compilation flags:
    static

Template:
    warnings(Warnings)
Mode and number of proofs:
    warnings(-list(compound)) - one

---

summary/1

Returns a machine-readable summary for the collected linter warnings.

Compilation flags:
    static

Template:
    summary(Summary)
Mode and number of proofs:
    summary(-compound) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

enabled_/0

True when warning collection is enabled.

Compilation flags:
    dynamic

Mode and number of proofs:
enabled_ - zero_or_one

---

warning_sequence_/1

Stores the last assigned warning sequence number.

Compilation flags:
dynamic

Template:
warning_sequence_(Sequence)
Mode and number of proofs:
warning_sequence_(-integer) - zero_or_one

---

recorded_warning_/4

Caches collected warnings together with their sequence number, flag, normalized message term, and printed message tokens.

Compilation flags:
dynamic

Template:
recorded_warning_(Sequence,Flag,Message,Tokens)
Mode and number of proofs:
recorded_warning_(?integer,?atom,?compound,?list(compound)) - zero_or_more

---

collection_options_/1

Stores the merged options for the current warning collection session.

Compilation flags:
dynamic

Template:
collection_options_(Options)
Mode and number of proofs:
collection_options_(-list(compound)) - zero_or_one

Operators

(none)

1.143 listing

category

1.143.1 listing

Listing predicates.

Availability:
logtalk_load(listing(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2024-01-26

Compilation flags:
static

Dependencies:
(none)

Remarks:

(none)

Inherited public predicates:
(none)

- Public predicates
  - listing/0
  - listing/1
  - portray_clause/1
- Protected predicates
- Private predicates
- Operators

Public predicates

listing/0

Lists all clauses of all visible dynamic predicates to the current output stream.

Compilation flags:
static

Mode and number of proofs:
listing - one

---

listing/1

Lists all clauses of a visible dynamic predicate or non-terminal to the current output stream. When the argument is a clause head, lists all matching clauses.

Compilation flags:
static

Template:
listing(Spec)
Mode and number of proofs:

listing(+predicate_indicator) - one_or_error
listing(+non_terminal_indicator) - one_or_error
listing(+callable) - one_or_error

Exceptions:
Spec is not ground:
instantiation_error
Spec is ground but not a valid predicate indicator:
type_error(predicate_indicator,Spec)
Spec is ground but not a valid non-terminal indicator:
type_error(non_terminal_indicator,Spec)
Spec is a predicate indicator but not a visible predicate:
existence_error(predicate,Spec)
Spec is a non-terminal indicator but not a visible non-terminal:
existence_error(non_terminal,Spec)
Spec is a callable term with a Functor/Arity indicator but not a visible predicate:
existence_error(predicate,Functor/Arity)
Spec is a predicate indicator of a visible predicate but not a dynamic predicate:
permission_error(access,predicate,Spec)
Spec is a non-terminal indicator of a visible non-terminal but not a dynamic non-terminal:
permission_error(access,non_terminal,Spec)
Spec is a callable term for a visible predicate with a Functor/Arity indicator but not a dynamic predicate:
permission_error(access,predicate,Functor/Arity)

---

portray_clause/1

Pretty prints a clause to the current output stream.

Compilation flags:
static

Template:
portray_clause(Clause)
Mode and number of proofs:
portray_clause(+clause) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.144 lof_anomaly_detector

object

1.144.1 lof_anomaly_detector

Local Outlier Factor anomaly detector with multiple distance metrics, mixed-feature support, and missing-value handling. Learns from baseline training examples selected from a dataset object implementing the anomaly_dataset_protocol protocol and returns a detector term that can be used for scoring, prediction, and export.

Availability:
    logtalk_load(lof_anomaly_detector(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-07

Compilation flags:
    static, context_switching_calls

Imports:
    public anomaly_detector_common
Uses:
    format
    list
    numberlist
    pairs
    type

Remarks:

(none)

Inherited public predicates:

anomaly_detector_options/2 check_anomaly_detector/1 check_option/1 check_options/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 predict/4 print_anomaly_detector/1 score/3 score_all/3 valid_anomaly_detector/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

anomaly_dataset_protocol, anomaly_detector_protocol, knn_distance_anomaly_detector, isolation_forest_anomaly_detector

1.145 logging

object

1.145.1 logger

Global logger object for logging events to files.

Availability:
logtalk_load(logging(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2011-01-06

Compilation flags:
static, context_switching_calls

Implements:
public loggingp

Remarks:
(none)

Inherited public predicates:
define_log_file/2 disable_logging/1 enable_logging/1 init_log_file/2 log_event/2 log_file/2 logging/1

- Public predicates
- Protected predicates
- Private predicates
  - log_file_/2
  - logging_to_file_/2
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

log_file_/2

Table of log files.

Compilation flags:
    dynamic

Template:
    log_file_(Alias,File)
Mode and number of proofs:
    log_file_(?atom,?nonvar) - zero_or_more

---

logging_to_file_/2

Table of logging file status for log files.

Compilation flags:
    dynamic

Template:
    logging_to_file_(Alias,Status)
Mode and number of proofs:
    logging_to_file_(?atom,?atom) - zero_or_more

Operators

(none)

category

1.145.2 logging

Logging events to files category.

Availability:
logtalk_load(logging(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2011-01-06

Compilation flags:
static

Implements:
public loggingp

Remarks:
(none)

Inherited public predicates:
define_log_file/2 disable_logging/1 enable_logging/1 init_log_file/2 log_event/2 log_file/2 logging/1

- Public predicates
- Protected predicates
- Private predicates
  - log_file_/2
  - logging_to_file_/2
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

log_file_/2

Table of log files.

Compilation flags:
    dynamic

Template:
    log_file_(Alias,File)
Mode and number of proofs:
    log_file_(?atom,?nonvar) - zero_or_more

---

logging_to_file_/2

Table of logging file status for log files.

Compilation flags:
    dynamic

Template:
    logging_to_file_(Alias,Status)
Mode and number of proofs:
    logging_to_file_(?atom,?atom) - zero_or_more

Operators

(none)

protocol

1.145.3 loggingp

Logging events to files protocol.

Availability:
logtalk_load(logging(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2011-01-06

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - log_file/2
  - define_log_file/2
  - init_log_file/2
  - log_event/2
  - logging/1
  - enable_logging/1
  - disable_logging/1
- Protected predicates
- Private predicates

- Operators

Public predicates

log_file/2

Access to the table of log files.

Compilation flags:
    static

Template:
    log_file(Alias,File)
Mode and number of proofs:
    log_file(?atom,?atom) - zero_or_more

---

define_log_file/2

Defines a log file with alias Alias and file name File. If the log file already exists, its contents are kept. Logging is enabled by default.

Compilation flags:
    static

Template:
    define_log_file(Alias,File)
Mode and number of proofs:
    define_log_file(+atom,+atom) - one

---

init_log_file/2

Initializes a new log file with alias Alias and file name File. If the log file already exists, its contents are erased. Logging is enabled by default.

Compilation flags:
    static

Template:
    init_log_file(Alias,File)
Mode and number of proofs:
    init_log_file(+atom,+atom) - one

---

log_event/2

Logs an event Event to a log file with alias Alias. Fails if a log file with alias Alias is not defined.

Compilation flags:
    static

Template:
    log_event(Alias,Event)
Mode and number of proofs:
    log_event(+atom,+nonvar) - zero_or_one

---

logging/1

True if logging to file with alias Alias is enabled.

Compilation flags:
    static

Template:
    logging(Alias)
Mode and number of proofs:
    logging(+atom) - zero_or_one

---

enable_logging/1

Enables logging to file with alias Alias. Fails if a log file with alias Alias is not defined.

Compilation flags:
    static

Template:
    enable_logging(Alias)
Mode and number of proofs:
    enable_logging(+atom) - zero_or_one

---

disable_logging/1

Disables logging to file with alias Alias. Fails if a log file with alias Alias is not defined.

Compilation flags:
    static

Template:
    disable_logging(Alias)
Mode and number of proofs:
    disable_logging(+atom) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

logging

1.146 logistic_regression_classifier

object

1.146.1 logistic_regression_classifier

Logistic regression classifier supporting binary and multiclass classification using joint softmax training. Learns from a dataset object implementing the dataset_protocol protocol and returns a classifier term that can be used for prediction and exported as predicate clauses.

Availability:
logtalk_load(logistic_regression_classifier(loader))

Author: Paulo Moura
Version: 2:0:0
Date: 2026-05-11

Compilation flags:
static, context_switching_calls

Imports:
public probabilistic_classifier_common
Uses:
format
linear_algebra
list
numberlist
pairs
population
type

Remarks:
(none)

Inherited public predicates:
check_classifier/1 check_option/1 check_options/1 classifier_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 predict_probabilities/3 print_classifier/1 valid_classifier/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

dataset_protocol, c45_classifier, knn_classifier, naive_bayes_classifier, nearest_centroid_classifier, random_forest_classifier, adaptive_boosting_classifier

1.147 loops

object

1.147.1 loop

Loop control structures predicates.

Availability:
logtalk_load(loops(loader))

Author: Paulo Moura
Version: 1:4:1
Date: 2020-12-20

Compilation flags:
static, context_switching_calls

Implements:
public loopp

Remarks:
(none)

Inherited public predicates:
dowhile/2 fordownto/3 fordownto/4 fordownto/5 foreach/3 foreach/4 forto/3 forto/4 forto/5 whiledo/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.147.2 loopp

Loop control constructs protocol.

Availability:
logtalk_load(loops(loader))

Author: Paulo Moura
Version: 1:3:0
Date: 2017-03-20

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - whiledo/2
  - dowhile/2
  - foreach/3
  - foreach/4
  - forto/3
  - forto/4
  - forto/5
  - fordownto/3
  - fordownto/4
  - fordownto/5
- Protected predicates
- Private predicates
- Operators

Public predicates

whiledo/2

While Condition is true do Action.

Compilation flags:
    static

Template:
    whiledo(Condition,Action)
Meta-predicate template:
    whiledo(0,0)
Mode and number of proofs:
    whiledo(+callable,@callable) - zero_or_one

dowhile/2

Do Action while Condition is true.

Compilation flags:
    static

Template:
    dowhile(Action,Condition)
Meta-predicate template:
    dowhile(0,0)
Mode and number of proofs:
    dowhile(@callable,+callable) - zero_or_one

---

foreach/3

For each Element in List call Goal.

Compilation flags:
    static

Template:
    foreach(Element,List,Goal)
Meta-predicate template:
    foreach(*,*,0)
Mode and number of proofs:
    foreach(@var,+list(term),@callable) - zero_or_one

---

foreach/4

For each Element in List at position Index call Goal. Index starts at 1.

Compilation flags:
    static

Template:
    foreach(Element,Index,List,Goal)

Meta-predicate template:
    foreach(*,*,*,0)
Mode and number of proofs:
    foreach(@var,@var,+list(term),@callable) - zero_or_one

---

forto/3

Calls Goal counting up from First to Last. Increment is 1. For convenience, First and Last can be arithmetic expressions. Fails iff Goal fails.

Compilation flags:
    static

Template:
    forto(First,Last,Goal)
Meta-predicate template:
    forto(*,*,0)
Mode and number of proofs:
    forto(+number,+number,@callable) - zero_or_one

---

forto/4

Calls Goal counting up from First to Last and binding Count to each successive value. Increment is 1. For convenience, First and Last can be arithmetic expressions. Fails iff Goal fails.

Compilation flags:
    static

Template:
    forto(Count,First,Last,Goal)
Meta-predicate template:
    forto(*,*,*,0)
Mode and number of proofs:
    forto(@var,+number,+number,@callable) - zero_or_one

---

forto/5

Calls Goal counting up from First to Last and binding Count to each successive value. For convenience, First, Last, and Increment can be arithmetic expressions (uses Increment absolute value). Fails iff Goal fails.

Compilation flags:
    static

Template:
    forto(Count,First,Last,Increment,Goal)
Meta-predicate template:
    forto(*,*,*,*,0)
Mode and number of proofs:
    forto(@var,+number,+number,+number,@callable) - zero_or_one

---

fordownto/3

Calls Goal counting down from First to Last. Decrement is 1. For convenience, First and Last can be arithmetic expressions. Fails iff Goal fails.

Compilation flags:
    static

Template:
    fordownto(First,Last,Goal)
Meta-predicate template:
    fordownto(*,*,0)
Mode and number of proofs:
    fordownto(+number,+number,@callable) - zero_or_one

---

fordownto/4

Calls Goal counting down from First to Last and binding Count to each successive value. Decrement is 1. For convenience, First and Last can be arithmetic expressions. Fails iff Goal fails.

Compilation flags:
    static

Template:
fordownto(Count,First,Last,Goal)
Meta-predicate template:
fordownto(*,*,*,0)
Mode and number of proofs:
fordownto(@var,+number,+number,@callable) - zero_or_one

---

fordownto/5

Calls Goal counting down from First to Last and binding Count to each successive value. For convenience, First, Last, and Decrement can be arithmetic expressions (uses Decrement absolute value). Fails iff Goal fails.

Compilation flags:
static

Template:
fordownto(Count,First,Last,Decrement,Goal)
Meta-predicate template:
fordownto(*,*,*,*,0)
Mode and number of proofs:
fordownto(@var,+number,+number,+number,@callable) - zero_or_one

---

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

loop

1.148 massey_ranker

object

1.148.1 massey_ranker

Massey pairwise preference ranker. Learns one deterministic rating per item from a dataset object implementing the pairwise_ranking_dataset_protocol protocol by solving the Massey linear system built from aggregated pairwise outcomes and returns a self-describing ranker term with diagnostics that can be used for ranking and export.

Availability:
logtalk_load(massey_ranker(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public ranking_dataset_common
public score_ranker_model_common
public pairwise_strength_ranker_common
Uses:
avltree
list
numberlist

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_ranker/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 grouped_dataset_groups/2 grouped_dataset_items/2 grouped_dataset_items/3 grouped_dataset_summary/2 learn/2 learn/3 option/2 option/3 pairwise_dataset_connected_components/2 pairwise_dataset_declared_items/2 pairwise_dataset_items/2 pairwise_dataset_matchups/2 pairwise_dataset_preferences/2 pairwise_dataset_summary/2 pairwise_dataset_win_totals/2 pairwise_measurement_dataset_connected_components/2 pairwise_measurement_dataset_declared_items/2 pairwise_measurement_dataset_items/2 pairwise_measurement_dataset_measurements/2 pairwise_measurement_dataset_summary/2 print_ranker/1 rank/3 ranker_options/2 scores/2

temporal_pairwise_dataset_connected_components/2
temporal_pairwise_dataset_declared_items/2 temporal_pairwise_dataset_games/2
temporal_pairwise_dataset_games/3 temporal_pairwise_dataset_items/2
temporal_pairwise_dataset_periods/2 temporal_pairwise_dataset_summary/2 valid_option/1
valid_options/1 valid_ranker/1 validate_grouped_dataset/1 validate_grouped_dataset/2
validate_pairwise_dataset/1 validate_pairwise_dataset/2
validate_pairwise_measurement_dataset/1 validate_pairwise_measurement_dataset/2
validate_temporal_pairwise_dataset/1 validate_temporal_pairwise_dataset/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

pairwise_ranking_dataset_protocol, ranker_protocol, colley_ranker, elo_ranker

1.149 mcp_server

protocol

1.149.1 mcp_prompt_protocol

Protocol for Logtalk objects that provide prompts to be exposed via an MCP (Model Context Protocol) server. Implements the MCP 2025-06-18 specification. Implementing objects must define the set of prompts available and handle prompt get requests. Prompts are templates for structured LLM interactions that can accept arguments to customize their behavior.

Availability:
logtalk_load(mcp_server(loader))

Author: Paulo Moura
Version: 0:2:0
Date: 2026-02-24

Compilation flags:
static

Dependencies:
(none)

Remarks:

- Capabilities: Objects providing prompts must declare prompts in their capabilities/1 predicate (from the mcp_tool_protocol protocol). The server will then advertise the prompts capability and handle prompts/list and prompts/get requests.
- Prompt descriptors: Each prompt is described by a prompt(Name, Description, Arguments) or prompt(Name, Title, Description, Arguments) term where Name is an atom, Title is an optional human-friendly display name (an atom), Description is a human-readable atom, and Arguments is a list of argument(ArgName, ArgDescription, Required) terms.
- Prompt messages: The prompt_get/3 predicate must return a result term containing a list of messages. Each message is a message(Role, Content) term where Role is user or assistant and Content is text(Text) where Text is an atom.

Inherited public predicates:
(none)

- Public predicates
  - prompts/1
  - prompt_get/3
- Protected predicates

- Private predicates
- Operators

Public predicates

prompts/1

Returns a list of prompt descriptors available from this object. Each descriptor is a compound term prompt(Name, Description, Arguments) or prompt(Name, Title, Description, Arguments) where Name is the MCP prompt name (an atom), Title is an optional human-friendly display name (an atom), Description is a human-readable description (an atom), and Arguments is a list of argument(ArgName, ArgDescription, Required) terms describing the prompt arguments. ArgName and ArgDescription are atoms, and Required is the boolean true or false.

Compilation flags:
    static

Template:
    prompts(Prompts)
Mode and number of proofs:
    prompts(-list(compound)) - one

---

prompt_get/3

Handles a prompt get request. Name is the MCP prompt name (as declared in prompts/1), Arguments is a list of ArgumentName-Value pairs provided by the client, and Result is unified with the prompt result. The result must be one of: messages(MessageList) for a list of prompt messages, or messages(Description, MessageList) to also include a description. Each message in the list must be a message(Role, Content) term where Role is user or assistant and Content is text(Text) where Text is an atom.

Compilation flags:
    static

Template:
    prompt_get(Name,Arguments,Result)
Mode and number of proofs:
    prompt_get(+atom,+list(pair),--compound) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

protocol

1.149.2 mcp_resource_protocol

Protocol for Logtalk objects that provide resources to be exposed via an MCP (Model Context Protocol) server. Implements the MCP 2025-06-18 specification. Implementing objects must define the set of resources available and handle resource read requests. Resources expose data and content from the application that MCP clients can access.

Availability:
logtalk_load(mcp_server(loader))

Author: Paulo Moura
Version: 0:2:0
Date: 2026-02-24

Compilation flags:
static

Dependencies:
(none)

Remarks:

- Capabilities: Objects providing resources must declare resources in their capabilities/1 predicate (from the mcp_tool_protocol protocol). The server will then advertise the resources capability and handle resources/list and resources/read requests.
- Resource descriptors: Each resource is described by a resource(URI, Name, Description, MimeType) or resource(URI, Name, Title, Description, MimeType) term where URI is the resource identifier (an atom), Name is a human-readable name (an atom), Title is an optional human-friendly display name (an atom), Description is a human-readable description (an atom), and MimeType is the MIME type of the resource content (an atom, e.g. `'text/plain'`).

- Resource contents: The resource_read/3 predicate must return a result term. The result must be contents(ContentList) where each item is text_content(URI, MimeType, Text) for text resources or blob_content(URI, MimeType, Base64Data) for binary resources encoded as base64.

Inherited public predicates:
(none)

- Public predicates
  - resources/1
  - resource_read/3
- Protected predicates
- Private predicates
- Operators

Public predicates

resources/1

Returns a list of resource descriptors available from this object. Each descriptor is a compound term resource(URI, Name, Description, MimeType) or resource(URI, Name, Title, Description, MimeType) where URI is the resource identifier (an atom, typically a URI like logtalk://my-app/data), Name is a human-readable name (an atom), Title is an optional human-friendly display name (an atom), Description is a human-readable description (an atom), and MimeType is the MIME type (an atom, e.g. 'text/plain', 'application/json').

Compilation flags:
static

Template:
resources(Resources)
Mode and number of proofs:
resources(-list(compound)) - one

resource_read/3

Handles a resource read request. URI is the resource identifier (as declared in resources/1), Arguments is a list of ArgumentName-Value pairs (currently unused but reserved for future use), and Result is unified with the resource result. The result must be contents(ContentList) where each content item is either text_content(URI, MimeType, Text) for text resources or blob_content(URI, MimeType, Base64Data) for binary resources encoded as base64. Text and Base64Data must be atoms.

Compilation flags:
    static

Template:
    resource_read(URI,Arguments,Result)
Mode and number of proofs:
    resource_read(+atom,+list(pair),--compound) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

object

1.149.3 mcp_server

MCP (Model Context Protocol) server for Logtalk applications. Exposes Logtalk objects implementing the mcp_tool_protocol and optionally mcp_prompt_protocol and mcp_resource_protocol protocols as MCP tool, prompt, and resource providers over stdio transport with Content-Length framing. Implements the MCP 2025-03-26 specification. Uses the json_rpc library for JSON-RPC 2.0 message handling.

Availability:
    logtalk_load(mcp_server(loader))

Author: Paulo Moura
Version: 0:5:1

Date: 2026-03-13

Compilation flags:
static, context_switching_calls

Imports:
public options
Uses:
json_rpc
list
term_io
user

Remarks:

- MCP specification: Implements the Model Context Protocol 2025-03-26: https://spec.modelcontextprotocol.io/specification/2025-03-26/
- Transport: Uses stdio (standard input/output) with Content-Length header framing as defined by the MCP specification.
- Version negotiation: During initialization, the server performs protocol version negotiation. The client sends its highest supported version in the initialize request. The server selects the highest version it supports that is compatible (i.e., less than or equal to the client version). If no compatible version exists, the server responds with error -32602 ("Unsupported protocol version") including a data field with the list of supported versions.
- Capabilities: Supports the tools capability (tools/list and tools/call). Optionally supports the prompts capability (prompts/list and prompts/get) when the application declares it via capabilities/1 and implements mcp_prompt_protocol. Optionally supports the resources capability (resources/list and resources/read) when the application declares it via capabilities/1 and implements mcp_resource_protocol. Optionally supports the elicitation capability (elicitation/create) when the application declares it via capabilities/1.
- Tool discovery: Tools are discovered from objects implementing the mcp_tool_protocol. Tool metadata (descriptions, parameter schemas) is derived from info/2 and mode/2 directives.
- Prompt discovery: Prompts are discovered from objects implementing the mcp_prompt_protocol. Prompt metadata (names, descriptions, arguments) is declared via the prompts/1 predicate.
- Resource discovery: Resources are discovered from objects implementing the mcp_resource_protocol. Resource metadata (URIs, names, descriptions, MIME types) is declared via the resources/1 predicate.
- Auto-dispatch: When an object does not define tool_call/3 or tool_call/4 for a tool, the server auto-dispatches: it calls the predicate as a message to the object, collects output-mode arguments, and returns them as text content.
- Elicitation: When the application declares elicitation in its capabilities/1, the server constructs an elicit closure and passes it to tool_call/4. The closure sends elicitation/create requests to the client and reads back the user response. This enables interactive tools that can ask the user questions during execution.
- Error handling: Predicate failures and exceptions both result in MCP tool-level errors with isError set to true.

Inherited public predicates:
check_option/1 check_options/1 default_option/1 default_options/1 option/2 option/3 valid_option/1 valid_options/1

- Public predicates
  - start/2
  - start/3
  - start/4
  - start/5
  - elicit_request/5
- Protected predicates
- Private predicates
  - initialized_/0
  - application_/1
  - server_name_/1
  - server_version_/1
  - server_title_/1
  - elicit_counter_/1
  - application_capabilities_/1
- Operators

Public predicates

start/2

Starts the MCP server with the given server name and application object. The application object must implement the mcp_tool_protocol protocol. Reads JSON-RPC messages from standard input and writes responses to standard output using Content-Length framing. Blocks until the client disconnects or an exit signal is received.

Compilation flags:
static

Template:
start(Name,Application)
Mode and number of proofs:
start(+atom,+object_identifier) - one

start/3

Starts the MCP server with the given server name, application object, and options. Currently supported options: server_version(Version) to set the server version (default '1.0.0'), server_title(Title) to set the server display title (default '', omitted when empty).

Compilation flags:
    static

Template:
    start(Name,Application,Options)
Mode and number of proofs:
    start(+atom,+object_identifier,+list) - one

---

start/4

Starts the MCP server with custom input and output streams. Useful for testing or non-stdio transports. Uses default options.

Compilation flags:
    static

Template:
    start(Name,Application,Input,Output)
Mode and number of proofs:
    start(+atom,+object_identifier,+stream,+stream) - one

---

start/5

Starts the MCP server with custom input and output streams and options.

Compilation flags:
    static

Template:
    start(Name,Application,Input,Output,Options)
Mode and number of proofs:

start(+atom,+object_identifier,+stream,+stream,+list) - one

---

elicit_request/5

Sends an elicitation/create request to the MCP client and reads the response. Input and Output are the I/O streams. Message is the prompt text (an atom). RequestedSchema is a curly-term JSON Schema describing the input to request from the user (e.g., {type-object, properties-{answer-{type-string, enum-[yes, no]}}, required-[answer]}). Answer is unified with accept(Content) (where Content is the user response as a curly-term), decline, or cancel. This predicate is typically not called directly but through the elicit closure passed to tool_call/4.

Compilation flags:
static

Template:
elicit_request(Input,Output,Message,RequestedSchema,Answer)
Mode and number of proofs:
elicit_request(+stream,+stream,+atom,+compound,--compound) - one

---

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

initialized_/0

Flag indicating that the server has been initialized.

Compilation flags:
dynamic

Mode and number of proofs:
initialized_ - zero_or_one

---

application_/1

Current application object.

Compilation flags:
    dynamic

Template:
    application_(Application)
Mode and number of proofs:
    application_(?object_identifier) - zero_or_one

---

server_name_/1

Current server name.

Compilation flags:
    dynamic

Template:
    server_name_(Name)
Mode and number of proofs:
    server_name_(?atom) - zero_or_one

---

server_version_/1

Current server version.

Compilation flags:
    dynamic

Template:
    server_version_(Version)
Mode and number of proofs:
    server_version_(?atom) - zero_or_one

---

server_title_/1

Current server display title.

Compilation flags:
    dynamic

Template:
    server_title_(Title)
Mode and number of proofs:
    server_title_(?atom) - zero_or_one

---

elicit_counter_/1

Counter for elicitation request identifiers.

Compilation flags:
    dynamic

Template:
    elicit_counter_(Counter)
Mode and number of proofs:
    elicit_counter_(?integer) - zero_or_one

---

application_capabilities_/1

Cached application capabilities list.

Compilation flags:
    dynamic

Template:
    application_capabilities_(Capabilities)
Mode and number of proofs:
    application_capabilities_(?list) - zero_or_one

Operators

(none)

protocol

1.149.4 mcp_tool_protocol

Protocol for Logtalk objects that provide tools to be exposed via an MCP (Model Context Protocol) server. Implements the MCP 2025-06-18 specification. Implementing objects must define the set of tools available and handle tool calls. Tool metadata (names, titles, descriptions, parameter schemas) can be derived automatically from info/2 and mode/2 directives on the tool predicates, or can be specified explicitly via the tool/1 predicate.

Availability:
logtalk_load(mcp_server(loader))

Author: Paulo Moura
Version: 0:3:0
Date: 2026-02-24

Compilation flags:
static

Dependencies:
(none)

Remarks:

- Capabilities: Objects can optionally define capabilities/1 to declare which MCP capabilities they require. Currently supported: elicitation (allows the server to ask the user questions during tool execution via MCP elicitation). If capabilities/1 is not defined, only the tools capability is advertised.
- Tool title: The tool title (human-friendly display name) is derived from a title key in the predicate info/2 directive. If not specified, the predicate name (functor) is used as the title.
- Structured output: Tools can declare an output schema via output_schema/2 and return structured(StructuredContent) or structured(ContentItems, StructuredContent) results. The structured content is included in the structuredContent field of the tool call response alongside the content array.
- Resource links: Tool results can include resource_link(URI, Name) or resource_link(URI, Name, Description, MimeType) content items in results/1 lists.
- Elicitation: Tools that need user interaction during execution should define tool_call/4 instead of tool_call/3. The extra argument is an elicitation closure that can be called as call(Elicit, Message, Schema, Answer) where Message is the prompt text (an atom), Schema is a JSON Schema curly-term for the requested input, and Answer is unified with accept(Content), decline, or cancel.

Inherited public predicates:
(none)

- Public predicates
  - capabilities/1
  - tools/1
  - tool_call/3
  - tool_call/4
  - output_schema/2
- Protected predicates
- Private predicates
- Operators

Public predicates

capabilities/1

Returns a list of MCP capabilities required by this tool provider. Currently supported capabilities: elicitation (server can ask the user questions during tool execution). If not defined, only the tools capability is advertised. The server uses this to build the capabilities field in the initialize response.

Compilation flags:
static

Template:
capabilities(Capabilities)
Mode and number of proofs:
capabilities(-list(atom)) - one

---

tools/1

Returns a list of tool descriptors available from this object. Each descriptor is a compound term tool(Name, Functor, Arity) where Name is the MCP tool name (an atom), Functor is the predicate functor, and Arity is the predicate arity. By default, the tool name is the predicate functor. Tool titles are derived from info/2 title keys (falling back to the predicate name). Tool descriptions and parameter schemas are derived from the info/2 and mode/2 directives of the corresponding predicates.

Compilation flags:

static

Template:
tools(Tools)
Mode and number of proofs:
tools(-list(compound)) - one

---

tool_call/3

Handles a tool call. Name is the MCP tool name (as declared in tools/1), Arguments is a list of ArgumentName-Value pairs, and Result is unified with the tool result. The result must be one of: text(Atom) for a text result, error(Atom) for an error result, results(List) for a list of content items where each item is text(Atom), error(Atom), resource_link(URI, Name), or resource_link(URI, Name, Description, MimeType), structured(StructuredContent) for structured output with auto-generated text, or structured(ContentItems, StructuredContent) for structured output with explicit content items. If this predicate is not defined for a tool, the MCP server will use auto-dispatch: it calls the tool predicate as a message to the implementing object, collects output arguments, and returns them as a text result.

Compilation flags:
static

Template:
tool_call(Name,Arguments,Result)
Mode and number of proofs:
tool_call(+atom,+list(pair),--compound) - one

---

tool_call/4

Handles a tool call with elicitation support. Same as tool_call/3 but receives an Elicit closure as the third argument. The closure can be called as call(Elicit, Message, Schema, Answer) where Message is a prompt text (an atom), Schema is a curly-term JSON Schema describing the requested input (e.g., {type-object, properties-{answer-{type-string, enum-[yes, no]}}, required-[answer]}), and Answer is unified with accept(Content) (where Content is the user response as a curly-term), decline, or cancel. Only available when the application declares elicitation in its capabilities/1. Falls back to tool_call/3 and then auto-dispatch if not defined.

Compilation flags:
static

Template:
tool_call(Name,Arguments,Elicit,Result)
Mode and number of proofs:
tool_call(+atom,+list(pair),+callable,--compound) - one

---

output_schema/2

Returns the JSON output schema for the given tool name. Optional; when defined, the schema is included in the tool descriptor as outputSchema and the tool can return structured(StructuredContent) or structured(ContentItems, StructuredContent) results. The schema must be a curly-term following JSON Schema format (e.g. {type-object, properties-{temperature-{type-number}}, required-[temperature]}).

Compilation flags:
static

Template:
output_schema(Name,Schema)
Mode and number of proofs:
output_schema(+atom,-compound) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.150 memcached

object

1.150.1 memcached

Portable Memcached client implementing the text (ASCII) protocol. Uses the sockets library for TCP communication.

Availability:
logtalk_load(memcached(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-09

Compilation flags:
static, context_switching_calls

Uses:
list
socket
user

Remarks:

- Supported backends: ECLiPSe, GNU Prolog, SICStus Prolog, SWI-Prolog, and Trealla Prolog (same as the sockets library).
- Protocol version: Implements the Memcached text (ASCII) protocol as documented in the official protocol.txt specification.
- Connection: The default Memcached port is 11211. Connections are represented as opaque handles that should be passed to all other predicates.
- Keys: Keys are atoms up to 250 characters. They must not include control characters or whitespace.
- Flags: Flags are 32-bit unsigned integers stored alongside the data, opaque to the server.
- Expiration: Expiration time in seconds. 0 means never expire. Values over 30 days (2592000) are treated as Unix timestamps.
- Storage commands: Supports set, add, replace, append, prepend, and cas (check-and-set).
- Retrieval commands: Supports get, gets (with CAS token), gat, and gats (get-and-touch).
- Other commands: Supports delete, incr/decr, touch, flush_all, version, stats, and quit.

Inherited public predicates:

(none)

- Public predicates
  - connect/3
  - connect/2
  - disconnect/1
  - set/5
  - set/3
  - add/5
  - replace/5
  - append/3
  - prepend/3
  - cas/6
  - get/3
  - get/4
  - gets/4
  - gets/5
  - mget/3
  - delete/2
  - incr/4
  - decr/4
  - touch/3
  - gat/4
  - gats/5
  - flush_all/1
  - flush_all/2
  - version/2
  - stats/2
  - stats/3
- Protected predicates
- Private predicates
- Operators

Public predicates

connect/3

Connects to a Memcached server at the given host and port. Returns a connection handle for subsequent operations.

Compilation flags:
    static

Template:
    connect(Host,Port,Connection)
Mode and number of proofs:
    connect(+atom,+integer,--compound) - one_or_error

Exceptions:
    Connection refused or network error:
        memcached_error(connection_failed)

---

connect/2

Connects to a Memcached server at the given host on the default port (11211). Returns a connection handle for subsequent operations.

Compilation flags:
    static

Template:
    connect(Host,Connection)
Mode and number of proofs:
    connect(+atom,--compound) - one_or_error

Exceptions:
    Connection refused or network error:
        memcached_error(connection_failed)

---

disconnect/1

Disconnects from the Memcached server. Sends a quit command before closing.

Compilation flags:
    static

Template:
    disconnect(Connection)
Mode and number of proofs:
    disconnect(+compound) - one

---

set/5

Stores the data unconditionally. Overwrites any existing data for the key.

Compilation flags:
    static

Template:
    set(Connection,Key,Value,Flags,ExpTime)
Mode and number of proofs:
    set(+compound,+atom,+atom,+integer,+integer) - one_or_error

Exceptions:
    Storage failed:
        memcached_error(not_stored)
    Network error:
        memcached_error(Error)

---

set/3

Stores the data unconditionally with default flags (0) and no expiration (0).

Compilation flags:
    static

Template:
    set(Connection,Key,Value)
Mode and number of proofs:
    set(+compound,+atom,+atom) - one_or_error

Exceptions:
    Storage failed:
        memcached_error(not_stored)
    Network error:
        memcached_error(Error)

---

add/5

Stores the data only if the key does not already exist.

Compilation flags:
    static

Template:
    add(Connection,Key,Value,Flags,ExpTime)
Mode and number of proofs:
    add(+compound,+atom,+atom,+integer,+integer) - one_or_error

Exceptions:
    Key already exists:
        memcached_error(not_stored)
    Network error:
        memcached_error(Error)

---

replace/5

Stores the data only if the key already exists.

Compilation flags:
    static

Template:
    replace(Connection,Key,Value,Flags,ExpTime)
Mode and number of proofs:
    replace(+compound,+atom,+atom,+integer,+integer) - one_or_error

Exceptions:
    Key does not exist:
        memcached_error(not_stored)
    Network error:
        memcached_error(Error)

---

append/3

Appends the data to the end of an existing item's data.

Compilation flags:
    static

Template:
    append(Connection,Key,Value)
Mode and number of proofs:
    append(+compound,+atom,+atom) - one_or_error

Exceptions:
    Key does not exist:
        memcached_error(not_stored)
    Network error:
        memcached_error(Error)

---

prepend/3

Prepends the data to the beginning of an existing item's data.

Compilation flags:
    static

Template:
    prepend(Connection,Key,Value)
Mode and number of proofs:
    prepend(+compound,+atom,+atom) - one_or_error

Exceptions:
    Key does not exist:
        memcached_error(not_stored)
    Network error:
        memcached_error(Error)

---

cas/6

Stores the data only if no one else has updated it since the given CAS unique value was obtained (via gets/3).

Compilation flags:
    static

Template:
    cas(Connection,Key,Value,Flags,ExpTime,CasUnique)
Mode and number of proofs:
    cas(+compound,+atom,+atom,+integer,+integer,+integer) - one_or_error

Exceptions:
    CAS value mismatch (item modified by another client):
        memcached_error(exists)
    Key does not exist:
        memcached_error(not_found)
    Network error:
        memcached_error(Error)

---

get/3

Retrieves the value associated with the key. Fails if the key is not found.

Compilation flags:
    static

Template:
    get(Connection,Key,Value)
Mode and number of proofs:
    get(+compound,+atom,-atom) - zero_or_one_or_error

---

get/4

Retrieves the value and flags associated with the key. Fails if the key is not found.

Compilation flags:
    static

Template:
    get(Connection,Key,Value,Flags)
Mode and number of proofs:
    get(+compound,+atom,-atom,-integer) - zero_or_one_or_error

---

gets/4

Retrieves the value and CAS unique token for the key. Fails if the key is not found. The CAS value is used with the cas/6 predicate.

Compilation flags:
    static

Template:
    gets(Connection,Key,Value,CasUnique)
Mode and number of proofs:
    gets(+compound,+atom,-atom,-integer) - zero_or_one_or_error

---

gets/5

Retrieves the value, flags, and CAS unique token for the key. Fails if the key is not found.

Compilation flags:
    static

Template:
    gets(Connection,Key,Value,Flags,CasUnique)
Mode and number of proofs:
    gets(+compound,+atom,-atom,-integer,-integer) - zero_or_one_or_error

---

mget/3

Retrieves multiple keys at once. Returns a list of item(Key, Value, Flags) terms for found keys.

Compilation flags:
    static

Template:
    mget(Connection,Keys,Items)
Mode and number of proofs:
    mget(+compound,+list(atom),-list(compound)) - one_or_error

---

delete/2

Deletes the item with the given key. Fails if the key is not found.

Compilation flags:
    static

Template:
    delete(Connection,Key)
Mode and number of proofs:

delete(+compound,+atom) - zero_or_one_or_error

---

incr/4

Increments the numeric value of the given key by the specified amount. Returns the new value. The item must already exist and contain a decimal representation of a 64-bit unsigned integer. Fails if the key is not found.

Compilation flags:
static

Template:
incr(Connection,Key,Amount,NewValue)
Mode and number of proofs:
incr(+compound,+atom,+integer,-integer) - zero_or_one_or_error

---

decr/4

Decrements the numeric value of the given key by the specified amount. Returns the new value. Underflow is caught: decrementing below 0 yields 0. Fails if the key is not found.

Compilation flags:
static

Template:
decr(Connection,Key,Amount,NewValue)
Mode and number of proofs:
decr(+compound,+atom,+integer,-integer) - zero_or_one_or_error

---

touch/3

Updates the expiration time of the given key without fetching the data. Fails if the key is not found.

Compilation flags:
    static

Template:
    touch(Connection,Key,ExpTime)
Mode and number of proofs:
    touch(+compound,+atom,+integer) - zero_or_one_or_error

---

gat/4

Gets the value of the key and updates its expiration time. Fails if the key is not found.

Compilation flags:
    static

Template:
    gat(Connection,Key,ExpTime,Value)
Mode and number of proofs:
    gat(+compound,+atom,+integer,-atom) - zero_or_one_or_error

---

gats/5

Gets the value and CAS unique token of the key and updates its expiration time. Fails if the key is not found.

Compilation flags:
    static

Template:
    gats(Connection,Key,ExpTime,Value,CasUnique)
Mode and number of proofs:
    gats(+compound,+atom,+integer,-atom,-integer) - zero_or_one_or_error

---

flush_all/1

Invalidates all existing items immediately.

Compilation flags:
    static

Template:
    flush_all(Connection)
Mode and number of proofs:
    flush_all(+compound) - one_or_error

---

flush_all/2

Invalidates all existing items after the specified number of seconds.

Compilation flags:
    static

Template:
    flush_all(Connection,Delay)
Mode and number of proofs:
    flush_all(+compound,+integer) - one_or_error

---

version/2

Returns the version string of the Memcached server.

Compilation flags:
    static

Template:
    version(Connection,Version)
Mode and number of proofs:

version(+compound,-atom) - one_or_error

---

stats/2

Returns general-purpose statistics as a list of stat(Name, Value) terms.

Compilation flags:
static

Template:
stats(Connection,Stats)
Mode and number of proofs:
stats(+compound,-list(compound)) - one_or_error

---

stats/3

Returns statistics for the given argument (e.g. items, slabs, sizes) as a list of stat(Name, Value) terms.

Compilation flags:
static

Template:
stats(Connection,Argument,Stats)
Mode and number of proofs:
stats(+compound,+atom,-list(compound)) - one_or_error

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.151 message_pack

object

1.151.1 message_pack(StringRepresentation)

- StringRepresentation - Text representation to be used when decoding MessagePack strings. Possible values are atom (default), chars, and codes.

MessagePack format exporter and importer.

Availability:
logtalk_load(message_pack(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-09

Compilation flags:
static, context_switching_calls

Uses:
ieee_754(Precision,ByteOrder,NaNRepresentation)
list
type
utf_8

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - parse/2
  - generate/2
- Protected predicates
- Private predicates
- Operators

Public predicates

parse/2

Parses a list of bytes in the MessagePack format returning the corresponding term representation. Throws an error when parsing is not possible (usually due to an invalid byte sequence).

Compilation flags:
    static

Template:
    parse(Bytes,Term)
Mode and number of proofs:
    parse(@list(byte),-ground) - one_or_error

---

generate/2

Generates a list of bytes in the MessagePack format representing the given term. Throws an error when generating is not possible (usually due to a term with no corresponding MessagePack representation).

Compilation flags:
    static

Template:
    generate(Term,Bytes)
Mode and number of proofs:
    generate(@ground,-list(byte)) - one_or_error

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.152 meta

object

1.152.1 meta

Some useful meta-predicates.

Availability:
logtalk_load(meta(loader))

Author: Paulo Moura
Version: 5:2:0
Date: 2016-10-06

Compilation flags:
static, context_switching_calls

Implements:
public metap
Aliases:
metap map/2 as succeeds/2
metap map/2 as maplist/2
metap map/3 as maplist/3
metap map/4 as maplist/4
metap map/5 as maplist/5
metap map/6 as maplist/6
metap map/7 as maplist/7
metap map/8 as maplist/8
metap include/3 as filter/3
metap fold_left/4 as foldl/4

metap fold_left_1/3 as foldl1/3
metap fold_right/4 as foldr/4
metap fold_right_1/3 as foldr1/3
metap scan_left/4 as scanl/4
metap scan_left_1/3 as scanl1/3
metap scan_right/4 as scanr/4
metap scan_right_1/3 as scanr1/3

Remarks:
(none)

Inherited public predicates:
exclude/3 findall_member/4 findall_member/5 fold_left/4 fold_left_1/3 fold_right/4 fold_right_1/3 include/3 map/2 map/3 map/4 map/5 map/6 map/7 map/8 map_reduce/5 partition/4 partition/6 scan_left/4 scan_left_1/3 scan_right/4 scan_right_1/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

meta_compiler

protocol

1.152.2 metap

Useful meta-predicates protocol.

Availability:
logtalk_load(meta(loader))

Author: Paulo Moura
Version: 6:1:0
Date: 2015-12-23

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - include/3
  - exclude/3
  - findall_member/4
  - findall_member/5
  - partition/4

      - partition/6
      - fold_left/4
      - fold_left_1/3
      - scan_left/4
      - scan_left_1/3
      - fold_right/4
      - fold_right_1/3
      - scan_right/4
      - scan_right_1/3
      - map/2
      - map/3
      - map/4
      - map/5
      - map/6
      - map/7
      - map/8
      - map_reduce/5
  - Protected predicates
  - Private predicates
  - Operators

Public predicates

include/3

Returns a list of all list elements that satisfy a predicate.

Compilation flags:
    static

Template:
    include(Closure,List,Included)
Meta-predicate template:
    include(1,*,*)
Mode and number of proofs:
    include(+callable,+list,-list) - one

exclude/3

Returns a list of all list elements that fail to satisfy a predicate.

Compilation flags:
    static

Template:
    exclude(Closure,List,Excluded)
Meta-predicate template:
    exclude(1,*,*)
Mode and number of proofs:
    exclude(+callable,+list,-list) - one

---

findall_member/4

Finds all members of a list that satisfy a given test.

Compilation flags:
    static

Template:
    findall_member(Member,List,Test,Result)
Meta-predicate template:
    findall_member(*,*,0,*)
Mode and number of proofs:
    findall_member(@term,+list,@callable,-list) - one

---

findall_member/5

Finds all members of a list that satisfy a given test appending the given tail to the result.

Compilation flags:
    static

Template:
    findall_member(Member,List,Test,Result,Tail)

Meta-predicate template:
    findall_member(*,*,0,*,*)
Mode and number of proofs:
    findall_member(@term,+list,@callable,-list,+list) - one

---

partition/4

Partition a list of elements in two lists using a predicate.

Compilation flags:
    static

Template:
    partition(Closure,List,Included,Excluded)
Meta-predicate template:
    partition(1,*,*,*)
Mode and number of proofs:
    partition(+callable,+list,-list,-list) - one

---

partition/6

Partitions a list in lists with values less, equal, and greater than a given value using a comparison predicate with the same argument order as compare/3.

Compilation flags:
    static

Template:
    partition(Closure,List,Value,Less,Equal,Greater)
Meta-predicate template:
    partition(3,*,*,*,*,*)
Mode and number of proofs:
    partition(+callable,+list,@term,-list,-list,-list) - one

---

fold_left/4

List folding (left associative). Closure is extended with three arguments: accumulator, list element, and updated accumulator.

Compilation flags:
    static

Template:
    fold_left(Closure,Accumulator,List,Result)
Meta-predicate template:
    fold_left(3,*,*,*)
Mode and number of proofs:
    fold_left(+callable,?term,+list,?term) - zero_or_more

---

fold_left_1/3

List folding (left associative). Closure is extended with three arguments: accumulator, list element, and updated accumulator. The initial value of the accumulator is the list first element. Fails for empty lists.

Compilation flags:
    static

Template:
    fold_left_1(Closure,List,Result)
Meta-predicate template:
    fold_left_1(3,*,*)
Mode and number of proofs:
    fold_left_1(+callable,+list,?term) - zero_or_more

---

scan_left/4

List scanning (left associative). Closure is extended with three arguments: accumulator, list element, and updated accumulator.

Compilation flags:
    static

Template:
scan_left(Closure,Accumulator,List,Results)
Meta-predicate template:
scan_left(3,*,*,*)
Mode and number of proofs:
scan_left(+callable,?term,+list,?list) - zero_or_more

---

scan_left_1/3

List scanning (left associative). Closure is extended with three arguments: accumulator, list element, and updated accumulator. The accumulator is initialized with the list first element. Fails for empty lists.

Compilation flags:
static

Template:
scan_left_1(Closure,List,Results)
Meta-predicate template:
scan_left_1(3,*,*)
Mode and number of proofs:
scan_left_1(+callable,+list,?list) - zero_or_more

---

fold_right/4

List folding (right associative). Closure is extended with three arguments: list element, accumulator, and updated accumulator.

Compilation flags:
static

Template:
fold_right(Closure,Accumulator,List,Result)
Meta-predicate template:
fold_right(3,*,*,*)
Mode and number of proofs:
fold_right(+callable,?term,+list,?term) - zero_or_more

---

fold_right_1/3

List folding (right associative). Closure is extended with three arguments: list element, accumulator, and updated accumulator. The initial value of the accumulator is the list first element. Fails for empty lists.

Compilation flags:
    static

Template:
    fold_right_1(Closure,List,Result)
Meta-predicate template:
    fold_right_1(3,*,*)
Mode and number of proofs:
    fold_right_1(+callable,+list,?term) - zero_or_more

---

scan_right/4

List scanning (right associative). Closure is extended with three arguments: list element, accumulator, and updated accumulator.

Compilation flags:
    static

Template:
    scan_right(Closure,Accumulator,List,Results)
Meta-predicate template:
    scan_right(3,*,*,*)
Mode and number of proofs:
    scan_right(+callable,?term,+list,?list) - zero_or_more

---

scan_right_1/3

List scanning (right associative). Closure is extended with three arguments: list element, accumulator, and updated accumulator. The accumulator is initialized with the list first element. Fails for empty lists.

Compilation flags:
    static

Template:
    scan_right_1(Closure,List,Results)
Meta-predicate template:
    scan_right_1(3,*,*)
Mode and number of proofs:
    scan_right_1(+callable,+list,?list) - zero_or_more

---

map/2

True if the predicate succeeds for each list element.

Compilation flags:
    static

Template:
    map(Closure,List)
Meta-predicate template:
    map(1,*)
Mode and number of proofs:
    map(+callable,?list) - zero_or_more

---

map/3

List mapping predicate taken arguments from two lists of elements.

Compilation flags:
    static

Template:
    map(Closure,List1,List2)
Meta-predicate template:
    map(2,*,*)
Mode and number of proofs:
    map(+callable,?list,?list) - zero_or_more

---

map/4

List mapping predicate taken arguments from three lists of elements.

Compilation flags:
    static

Template:
    map(Closure,List1,List2,List3)
Meta-predicate template:
    map(3,*,*,*)
Mode and number of proofs:
    map(+callable,?list,?list,?list) - zero_or_more

---

map/5

List mapping predicate taken arguments from four lists of elements.

Compilation flags:
    static

Template:
    map(Closure,List1,List2,List3,List4)
Meta-predicate template:
    map(4,*,*,*,*)
Mode and number of proofs:
    map(+callable,?list,?list,?list,?list) - zero_or_more

---

map/6

List mapping predicate taken arguments from five lists of elements.

Compilation flags:
    static

Template:
    map(Closure,List1,List2,List3,List4,List5)

Meta-predicate template:
    map(5,*,*,*,*,*)
Mode and number of proofs:
    map(+callable,?list,?list,?list,?list,?list) - zero_or_more

---

map/7

List mapping predicate taken arguments from six lists of elements.

Compilation flags:
    static

Template:
    map(Closure,List1,List2,List3,List4,List5,List6)
Meta-predicate template:
    map(6,*,*,*,*,*,*)
Mode and number of proofs:
    map(+callable,?list,?list,?list,?list,?list,?list) - zero_or_more

---

map/8

List mapping predicate taken arguments from seven lists of elements.

Compilation flags:
    static

Template:
    map(Closure,List1,List2,List3,List4,List5,List6,List7)
Meta-predicate template:
    map(7,*,*,*,*,*,*,*)
Mode and number of proofs:
    map(+callable,?list,?list,?list,?list,?list,?list,?list) - zero_or_more

---

map_reduce/5

Map a list and apply a fold left (reduce) to the resulting list.

Compilation flags:
    static

Template:
    map_reduce(Map,Reduce,Accumulator,List,Result)
Meta-predicate template:
    map_reduce(2,3,*,*,*)
Mode and number of proofs:
    map_reduce(+callable,+callable,+term,?list,?term) - zero_or_more

---

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

meta

1.153 meta_compiler

object

1.153.1 meta_compiler

Compiler for the meta object meta-predicates. Generates auxiliary predicates in order to avoid meta-call overheads.

Availability:
logtalk_load(meta_compiler(loader))

Author: Paulo Moura
Version: 0:16:1
Date: 2026-04-22

Compilation flags:
static, context_switching_calls

Implements:
public expanding
Uses:
gensym
list
logtalk
user

Remarks:

- Usage: Compile source files with calls to the meta object meta-predicates using the compiler option hook(meta_compiler).

Inherited public predicates:
goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
  - generated_predicate_/1
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

generated_predicate_/1

Table of generated auxiliary predicates.

Compilation flags:
    dynamic

Template:
    generated_predicate_(Predicate)
Mode and number of proofs:
    generated_predicate_(?predicate_indicator) - zero_or_more

---

Operators

(none)

See also

meta

1.154 metagol

object

1.154.1 metagol

Inductive logic programming (ILP) system based on meta-interpretive learning.

Availability:
logtalk_load(metagol(loader))

Author: Metagol authors; adapted to Logtalk by Paulo Moura.
Version: 0:24:4
Date: 2024-03-15

Copyright: Copyright 2016 Metagol authors; Copyright 2018-2024 Paulo Moura
License: BSD-3-Clause

Compilation flags:
static, context_switching_calls

Implements:
public expanding
Provides:
logtalk::message_tokens//2
logtalk::message_prefix_stream/4
Uses:
coroutining
integer
list
logtalk
meta
timeout

Remarks:
(none)

Inherited public predicates:
goal_expansion/2 term_expansion/2

- Public predicates
  - learn/3
  - learn/2

  - learn_seq/2
  - learn_with_timeout/4
  - program_to_clauses/2
  - pprint/1
  - metarule/6
  - head_pred/1
  - body_pred/1
  - ibk/3
  - func_test/3
  - functional/0
  - min_clauses/1
  - max_clauses/1
  - max_inv_preds/1
  - metarule_next_id/1
  - timeout/1
- Protected predicates
  - pprint_clause/1
  - pprint_clauses/1
  - compiled_pred_call/2
  - body_pred_call/2
  - type/3
- Private predicates
- Operators

Public predicates

learn/3

Learns from a set of positive examples and a set of negative examples and returns the learned program.

Compilation flags:
    static

Template:
    learn(PositiveExamples,NegativeExamples,Program)
Mode and number of proofs:
    learn(@list(example),@list(example),-list(term)) - zero_or_more

---

learn/2

Learns from a set of positive examples and a set of negative examples and pretty prints the learned program.

Compilation flags:
    static

Template:
    learn(PositiveExamples,NegativeExamples)
Mode and number of proofs:
    learn(@list(example),@list(example)) - zero_or_more

---

learn_seq/2

Learns from a sequence of examples represented as a list of PositiveExamples/NegativeExamples elements and returns the learned program.

Compilation flags:
    static

Template:
    learn_seq(Examples,Program)
Mode and number of proofs:
    learn_seq(@list(example),-list(clause)) - zero_or_one

---

learn_with_timeout/4

Learns from a set of positive examples and a set of negative examples and returns the learned program constrained by the given timeout or its default value.

Compilation flags:
    static

Template:

learn_with_timeout(PositiveExamples,NegativeExamples,Program,Timeout)
Mode and number of proofs:
learn_with_timeout(@list(example),@list(example),-list(term),+number) - zero_or_one_or_error
learn_with_timeout(@list(example),@list(example),-list(term),-number) - zero_or_one_or_error

Exceptions:
Learning does not complete in the allowed time:
timeout(learn(PositiveExamples,NegativeExamples,Program))

---

program_to_clauses/2

Converts a learned program into a list of clauses.

Compilation flags:
static

Template:
program_to_clauses(Program,Clauses)
Mode and number of proofs:
program_to_clauses(@list(term),-list(clause)) - one

---

pprint/1

Pretty prints a learned program.

Compilation flags:
static

Template:
pprint(Program)
Mode and number of proofs:
pprint(@list(term)) - one

---

metarule/6

Compilation flags:
    static

---

head_pred/1

Compilation flags:
    static

---

body_pred/1

Compilation flags:
    dynamic

---

ibk/3

Compilation flags:
    static

---

func_test/3

Compilation flags:
    static

---

functional/0

Compilation flags:
    dynamic

---

min_clauses/1

Compilation flags:
dynamic

---

max_clauses/1

Compilation flags:
dynamic

---

max_inv_preds/1

Compilation flags:
dynamic

---

metarule_next_id/1

Compilation flags:
dynamic

---

timeout/1

Compilation flags:
dynamic

Protected predicates

pprint_clause/1

Compilation flags:
    static

---

pprint_clauses/1

Compilation flags:
    static

---

compiled_pred_call/2

Compilation flags:
    dynamic

---

body_pred_call/2

Compilation flags:
    dynamic

---

type/3

Compilation flags:
    dynamic

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.154.2 metagol_example_protocol

Convenient learning predicates for use in examples and unit tests.

Availability:
logtalk_load(metagol(loader))

Author: Paulo Moura.
Version: 0:1:1
Date: 2024-03-15

License: BSD-3-Clause

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - learn/1
  - learn/0
- Protected predicates

- Private predicates
- Operators

Public predicates

learn/1

Learns and returns set of clauses.

Compilation flags:
    static

Template:
    learn(Clauses)
Mode and number of proofs:
    learn(-list(clause)) - zero_or_more

---

learn/0

Learns and prints a set of clauses.

Compilation flags:
    static

Mode and number of proofs:
    learn - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.155 mime_types

object

1.155.1 mime_types

MIME type registry and convenience predicates for mapping file names and URLs to media types.

Availability:
logtalk_load(mime_types(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-04-17

Compilation flags:
static, context_switching_calls

Uses:
list
os
reader
string(Representation)

Remarks:

- Default behavior: Convenience predicates default to lenient lookup by consulting both built-in standard mappings and built-in common mappings.
- Path decomposition: File path decomposition uses the os library decompose_file_name/4 predicate.
- External overlays: Additional mime.types-style files can be loaded into the in-memory registry using the load/1-2 predicates.

Inherited public predicates:

(none)

- Public predicates
  - guess_type/3
  - guess_type/4
  - guess_file_type/3
  - guess_file_type/4
  - extension_type/2
  - extension_type/3
  - guess_extension/2
  - guess_extension/3
  - guess_all_extensions/2
  - guess_all_extensions/3
  - add_type/2
  - add_type/3
  - read_mime_types/2
  - load/1
  - load/2
  - reset/0
  - suffix_alias/2
  - encoding_suffix/2
- Protected predicates
- Private predicates
  - runtime_type_/3
- Operators

Public predicates

guess_type/3

Guesses the MIME type and content encoding for a URL or file name using lenient lookup. Unknown values are returned as the empty atom.

Compilation flags:
static

Template:
guess_type(Resource,Type,Encoding)
Mode and number of proofs:
guess_type(+atom,-atom,-atom) - one

---

guess_type/4

Guesses the MIME type and content encoding for a URL or file name. When Strict is true, only built-in standard mappings and strict runtime overlays are consulted.

Compilation flags:
static

Template:
guess_type(Resource,Type,Encoding,Strict)
Mode and number of proofs:
guess_type(+atom,-atom,-atom,+boolean) - one

---

guess_file_type/3

Guesses the MIME type and content encoding for a file path using lenient lookup. Unknown values are returned as the empty atom.

Compilation flags:
static

Template:
guess_file_type(Path,Type,Encoding)
Mode and number of proofs:
guess_file_type(+atom,-atom,-atom) - one

---

guess_file_type/4

Guesses the MIME type and content encoding for a file path.

Compilation flags:
    static

Template:
    guess_file_type(Path,Type,Encoding,Strict)
Mode and number of proofs:
    guess_file_type(+atom,-atom,-atom,+boolean) - one

---

extension_type/2

Returns the MIME type associated with a file extension using lenient lookup.

Compilation flags:
    static

Template:
    extension_type(Extension,Type)
Mode and number of proofs:
    extension_type(+atom,-atom) - zero_or_one

---

extension_type/3

Returns the MIME type associated with a file extension.

Compilation flags:
    static

Template:
    extension_type(Extension,Type,Strict)
Mode and number of proofs:
    extension_type(+atom,-atom,+boolean) - zero_or_one

---

guess_extension/2

Returns the preferred file extension associated with a MIME type using lenient lookup.

Compilation flags:
    static

Template:
    guess_extension(Type,Extension)
Mode and number of proofs:
    guess_extension(+atom,-atom) - zero_or_one

---

guess_extension/3

Returns the preferred file extension associated with a MIME type.

Compilation flags:
    static

Template:
    guess_extension(Type,Extension,Strict)
Mode and number of proofs:
    guess_extension(+atom,-atom,+boolean) - zero_or_one

---

guess_all_extensions/2

Returns all known file extensions associated with a MIME type using lenient lookup.

Compilation flags:
    static

Template:
    guess_all_extensions(Type,Extensions)
Mode and number of proofs:
    guess_all_extensions(+atom,-list(atom)) - one
    guess_all_extensions(+atom,-list(atom)) - one

---

guess_all_extensions/3

Returns all known file extensions associated with a MIME type.

Compilation flags:
    static

Template:
    guess_all_extensions(Type,Extensions,Strict)

---

add_type/2

Adds a lenient runtime mapping from a MIME type to a file extension.

Compilation flags:
    static

Template:
    add_type(Type,Extension)
Mode and number of proofs:
    add_type(+atom,+atom) - one

---

add_type/3

Adds a runtime mapping from a MIME type to a file extension.

Compilation flags:
    static

Template:
    add_type(Type,Extension,Strict)
Mode and number of proofs:
    add_type(+atom,+atom,+boolean) - one

---

read_mime_types/2

Reads a mime.types-style file returning a list of Extension-Type pairs.

Compilation flags:
    static

Template:
    read_mime_types(File,Pairs)
Mode and number of proofs:
    read_mime_types(+atom,-list(pair(atom,atom))) - one

---

load/1

Loads a mime.types-style file as a lenient runtime overlay.

Compilation flags:
    static

Template:
    load(File)
Mode and number of proofs:
    load(+atom) - one

---

load/2

Loads a mime.types-style file as a strict or lenient runtime overlay.

Compilation flags:
    static

Template:
    load(File,Strict)
Mode and number of proofs:

load(+atom,+boolean) - one

---

reset/0

Clears all runtime MIME type overlays.

Compilation flags:
static

Mode and number of proofs:
reset - one

---

suffix_alias/2

Returns suffix aliases used before splitting encoding and type suffixes.

Compilation flags:
static

Template:
suffix_alias(Alias,Expanded)
Mode and number of proofs:
suffix_alias(?atom,?atom) - zero_or_more

---

encoding_suffix/2

Returns content-encoding suffix mappings.

Compilation flags:
static

Template:
encoding_suffix(Extension,Encoding)
Mode and number of proofs:

encoding_suffix(?atom,?atom) - zero_or_more

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

runtime_type_/3

Table of runtime MIME type overlays.

Compilation flags:
dynamic

Template:
runtime_type_(Strict,Extension,Type)
Mode and number of proofs:
runtime_type_(?boolean,?atom,?atom) - zero_or_more

Operators

(none)

1.156 modified_z_score_anomaly_detector

object

1.156.1 modified_z_score_anomaly_detector

Statistical modified Z-score anomaly detector for continuous datasets. Learns per-attribute sample median and median absolute deviation from baseline training examples selected from a dataset object implementing the anomaly_dataset_protocol protocol and returns a detector term that can be used for scoring, prediction, and export.

Availability:
logtalk_load(modified_z_score_anomaly_detector(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public anomaly_detector_common
Uses:
format
list
numberlist
pairs
sample
type

Remarks:
(none)

Inherited public predicates:
anomaly_detector_options/2 check_anomaly_detector/1 check_option/1 check_options/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 predict/4 print_anomaly_detector/1 score/3 score_all/3 valid_anomaly_detector/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

anomaly_dataset_protocol, anomaly_detector_protocol, z_score_anomaly_detector, isolation_forest_anomaly_detector, knn_distance_anomaly_detector, lof_anomaly_detector

1.157 multisets

object

1.157.1 multisets

Implementation of multiset operations with repetition over lists.

Availability:
logtalk_load(multisets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-12

Compilation flags:
static, context_switching_calls

Implements:
public multisets_protocol
Uses:
fast_random(Algorithm)
list
natural

Remarks:
(none)

Inherited public predicates:
count_distinct_multisets/3 count_multisets/3 distinct_multiset/3 distinct_multiset/4 distinct_multiset_index/4 distinct_multisets/3 distinct_multisets/4 multiset/3 multiset/4 multiset_index/4 multiset_index/5 multisets/3 multisets/4 next_multiset/3 nth_distinct_multiset/4 nth_multiset/4 nth_multiset/5 previous_multiset/3 random_distinct_multiset/3 random_multiset/3 sample_distinct_multisets/4 sample_multisets/4

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.157.2 multisets_protocol

Protocol for multiset operations over lists.

Availability:
logtalk_load(multisets(loader))

Author: Paulo Moura
Version: 1:0:0

Date: 2026-05-12

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - multisets/3
  - multiset/3
  - multisets/4
  - multiset/4
  - distinct_multisets/3
  - distinct_multiset/3
  - distinct_multisets/4
  - distinct_multiset/4
  - nth_multiset/4
  - nth_multiset/5
  - multiset_index/4
  - multiset_index/5
  - count_multisets/3
  - count_distinct_multisets/3
  - nth_distinct_multiset/4
  - distinct_multiset_index/4
  - random_multiset/3
  - sample_multisets/4
  - random_distinct_multiset/3
  - sample_distinct_multisets/4

  - next_multiset/3
  - previous_multiset/3
- Protected predicates
- Private predicates
- Operators

Public predicates

multisets/3

Generates all K-multisets (unordered K-element selections with replacement) of a list using default order.

Compilation flags:
    static

Template:
    multisets(K,List,Multisets)
Mode and number of proofs:
    multisets(+integer,+list,-list) - one

---

multiset/3

True iff the third argument is a K-multiset with replacement of a list using default order.

Compilation flags:
    static

Template:
    multiset(K,List,Multiset)
Mode and number of proofs:
    multiset(+integer,+list,-list) - one_or_more

---

multisets/4

Generates all K-multisets with the given order: default or lexicographic.

Compilation flags:
    static

Template:
    multisets(K,List,Order,Multisets)
Mode and number of proofs:
    multisets(+integer,+list,+atom,-list) - one

---

multiset/4

True iff the fourth argument is a K-multiset with replacement with the given order: default or lexicographic.

Compilation flags:
    static

Template:
    multiset(K,List,Order,Multiset)
Mode and number of proofs:
    multiset(+integer,+list,+atom,-list) - one_or_more

---

distinct_multisets/3

Generates all distinct K-multisets (deduplicating repeated values in the input list) using default order.

Compilation flags:
    static

Template:
    distinct_multisets(K,List,Multisets)
Mode and number of proofs:
    distinct_multisets(+integer,+list,-list) - one

---

distinct_multiset/3

True iff the third argument is a distinct K-multiset with replacement of a list using default order.

Compilation flags:
    static

Template:
    distinct_multiset(K,List,Multiset)
Mode and number of proofs:
    distinct_multiset(+integer,+list,-list) - one_or_more

---

distinct_multisets/4

Generates all distinct K-multisets with the given order: default or lexicographic.

Compilation flags:
    static

Template:
    distinct_multisets(K,List,Order,Multisets)
Mode and number of proofs:
    distinct_multisets(+integer,+list,+atom,-list) - one

---

distinct_multiset/4

True iff the fourth argument is a distinct K-multiset with replacement with the given order: default or lexicographic.

Compilation flags:
    static

Template:
    distinct_multiset(K,List,Order,Multiset)
Mode and number of proofs:
    distinct_multiset(+integer,+list,+atom,-list) - one_or_more

---

nth_multiset/4

Returns the K-multiset at a given zero-based index using default order.

Compilation flags:
    static

Template:
    nth_multiset(K,List,Index,Multiset)
Mode and number of proofs:
    nth_multiset(+integer,+list,+integer,-list) - zero_or_one

---

nth_multiset/5

Returns the K-multiset at a given zero-based index in the given order: default or lexicographic.

Compilation flags:
    static

Template:
    nth_multiset(K,List,Order,Index,Multiset)
Mode and number of proofs:
    nth_multiset(+integer,+list,+atom,+integer,-list) - zero_or_one

---

multiset_index/4

Returns the zero-based index of a K-multiset using default order.

Compilation flags:
    static

Template:
    multiset_index(K,List,Multiset,Index)
Mode and number of proofs:

multiset_index(+integer,+list,+list,-integer) - zero_or_one

---

multiset_index/5

Returns the zero-based index of a K-multiset in the given order: default or lexicographic.

Compilation flags:
static

Template:
multiset_index(K,List,Order,Multiset,Index)
Mode and number of proofs:
multiset_index(+integer,+list,+atom,+list,-integer) - zero_or_one

---

count_multisets/3

Counts the number of K-multisets with replacement of a list.

Compilation flags:
static

Template:
count_multisets(K,List,Count)
Mode and number of proofs:
count_multisets(+integer,+list,-integer) - one

---

count_distinct_multisets/3

Counts the number of distinct K-multisets with replacement of a list (deduplicating repeated values in the input list).

Compilation flags:
static

Template:
count_distinct_multisets(K,List,Count)
Mode and number of proofs:
count_distinct_multisets(+integer,+list,-integer) - one

---

nth_distinct_multiset/4

Returns the distinct K-multiset at a given zero-based index in default generation order.

Compilation flags:
static

Template:
nth_distinct_multiset(K,List,Index,Multiset)
Mode and number of proofs:
nth_distinct_multiset(+integer,+list,+integer,-list) - zero_or_one

---

distinct_multiset_index/4

Returns the zero-based index of a distinct K-multiset in default generation order.

Compilation flags:
static

Template:
distinct_multiset_index(K,List,Multiset,Index)
Mode and number of proofs:
distinct_multiset_index(+integer,+list,+list,-integer) - zero_or_one

---

random_multiset/3

Returns a random K-multiset with replacement of a list.

Compilation flags:
    static

Template:
    random_multiset(K,List,Multiset)
Mode and number of proofs:
    random_multiset(+integer,+list,-list) - zero_or_one

---

sample_multisets/4

Returns SampleCount random K-multisets with replacement of a list, sampled with replacement.

Compilation flags:
    static

Template:
    sample_multisets(K,List,SampleCount,Samples)
Mode and number of proofs:
    sample_multisets(+integer,+list,+integer,-list) - zero_or_one

---

random_distinct_multiset/3

Returns a random distinct K-multiset with replacement of a list (deduplicating repeated values in the input list).

Compilation flags:
    static

Template:
    random_distinct_multiset(K,List,Multiset)
Mode and number of proofs:
    random_distinct_multiset(+integer,+list,-list) - zero_or_one

---

sample_distinct_multisets/4

Returns SampleCount random distinct K-multisets with replacement of a list, sampled with replacement after deduplicating repeated values in the input list.

Compilation flags:
    static

Template:
    sample_distinct_multisets(K,List,SampleCount,Samples)
Mode and number of proofs:
    sample_distinct_multisets(+integer,+list,+integer,-list) - zero_or_one

---

next_multiset/3

Returns the next distinct multiset value in lexicographic order induced by the first argument.

Compilation flags:
    static

Template:
    next_multiset(List,Multiset,Next)
Mode and number of proofs:
    next_multiset(+list,+list,-list) - zero_or_one

---

previous_multiset/3

Returns the previous distinct multiset value in lexicographic order induced by the first argument.

Compilation flags:
    static

Template:
    previous_multiset(List,Multiset,Previous)

Mode and number of proofs:
previous_multiset(+list,+list,-list) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.158 mutation_testing

object

1.158.1 arithmetic_operator_replacement(Entity,Predicate,ClauseIndex,Occurrence,PrintMutation)

- Entity - Identifier of the entity being mutated.
- Predicate - Predicate or non-terminal indicator selecting clauses to mutate.
- ClauseIndex - 1-based clause index for the selected mutation (equal to Occurrence for this mutator).
- Occurrence - 1-based mutation occurrence index to target within selected predicate clauses.
- PrintMutation - Boolean flag to print the original and mutated term plus source location.

Hook object implementing the arithmetic_operator_replacement mutator for matching predicate clauses.

Availability:
logtalk_load(mutation_testing(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-20

Compilation flags:
static, context_switching_calls

Implements:
public expanding
public clause_mutator_protocol
Imports:
public mutator_common

Remarks:
(none)

Inherited public predicates:
coverage_clause_mutator/0 goal_expansion/2 mutation/2 reset/0 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.158.2 body_goal_negation(Entity,Predicate,ClauseIndex,Occurrence,PrintMutation)

- Entity - Identifier of the entity being mutated.
- Predicate - Predicate or non-terminal indicator selecting clauses to mutate.
- ClauseIndex - 1-based clause index for the selected mutation (equal to Occurrence for this mutator).
- Occurrence - 1-based mutation occurrence index to target within selected predicate clauses.
- PrintMutation - Boolean flag to print the original and mutated term plus source location.

Hook object implementing the body_goal_negation mutator by negating matching predicate clause bodies.

Availability:
logtalk_load(mutation_testing(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-20

Compilation flags:
static, context_switching_calls

Implements:
public expanding
public clause_mutator_protocol
Imports:
public mutator_common

Remarks:
(none)

Inherited public predicates:
coverage_clause_mutator/0 goal_expansion/2 mutation/2 reset/0 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.158.3 clause_mutator_protocol

Marker protocol for clause and grammar-rule mutators.

Availability:
logtalk_load(mutation_testing(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-20

Compilation flags:
static

Extends:
public mutator_protocol

Remarks:
(none)

Inherited public predicates:
coverage_clause_mutator/0 mutation/2 reset/0

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.158.4 clauses_reordering(Entity,Predicate,ClauseIndex,Occurrence,PrintMutation)

- Entity - Identifier of the entity being mutated.
- Predicate - Predicate or non-terminal indicator selecting clauses to mutate.
- ClauseIndex - 1-based clause index selecting which clause is swapped with its successor (last swaps with first).
- Occurrence - 1-based clause index selecting which clause is swapped with its successor (last swaps with first).
- PrintMutation - Boolean flag to print the original and mutated term plus source location.

Hook object implementing the clauses_reordering mutator by reordering the clauses of a non-discontiguous predicate or non-terminal definition.

Availability:
logtalk_load(mutation_testing(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-20

Compilation flags:
    static, context_switching_calls

Implements:
    public expanding
    public clause_mutator_protocol
Imports:
    public mutator_common
Uses:
    list

Remarks:
    (none)

Inherited public predicates:
    coverage_clause_mutator/0 goal_expansion/2 mutation/2 reset/0 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
  - clauses_/1
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

clauses_/1

Table of parsed clauses.

Compilation flags:
    dynamic

Template:
clauses_(Clause)
Mode and number of proofs:
clauses_(?clause) - zero_or_more

Operators

(none)

protocol

1.158.5 directive_mutator_protocol

Marker protocol for directive mutators.

Availability:
logtalk_load(mutation_testing(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-20

Compilation flags:
static

Extends:
public mutator_protocol

Remarks:
(none)

Inherited public predicates:
coverage_clause_mutator/0 mutation/2 reset/0

- Public predicates
- Protected predicates
- Private predicates

- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.158.6 fail_insertion(Entity,Predicate,ClauseIndex,Occurrence,PrintMutation)

- Entity - Identifier of the entity being mutated.
- Predicate - Predicate or non-terminal indicator selecting clauses to mutate.
- ClauseIndex - 1-based clause index for the selected mutation (equal to Occurrence for this mutator).
- Occurrence - 1-based mutation occurrence index to target within selected predicate clauses.
- PrintMutation - Boolean flag to print the original and mutated term plus source location.

Hook object implementing the fail_insertion mutator by inserting fail at deterministic body positions for matching predicate clauses.

Availability:
    logtalk_load(mutation_testing(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-20

Compilation flags:
    static, context_switching_calls

Implements:

public expanding
public clause_mutator_protocol

Imports:

public mutator_common

Remarks:

(none)

Inherited public predicates:

coverage_clause_mutator/0 goal_expansion/2 mutation/2 reset/0 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.158.7 head_arguments_mutation(Entity,Predicate,ClauseIndex,Occurrence,PrintMutation)

- Entity - Identifier of the entity being mutated.
- Predicate - Predicate or non-terminal indicator selecting clauses to mutate.
- ClauseIndex - 1-based clause index for the selected mutation (equal to Occurrence for this mutator).
- Occurrence - 1-based mutation occurrence index selecting one compile-time bound head argument to mutate.
- PrintMutation - Boolean flag to print the original and mutated term plus source location.

Hook object implementing the head_arguments_mutation mutator by mutating one compile-time bound predicate/non-terminal head argument using the type::mutation/3 predicate.

Availability:
logtalk_load(mutation_testing(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-20

Compilation flags:
static, context_switching_calls

Implements:
public expanding
public clause_mutator_protocol
Imports:
public mutator_common
Uses:
type

Remarks:
(none)

Inherited public predicates:
coverage_clause_mutator/0 goal_expansion/2 mutation/2 reset/0 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates

- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.158.8 head_arguments_reordering(Entity,Predicate,ClauseIndex,Occurrence,PrintMutation)

- Entity - Identifier of the entity being mutated.
- Predicate - Predicate or non-terminal indicator selecting clauses to mutate.
- ClauseIndex - 1-based clause index for the selected mutation (equal to Occurrence for this mutator).
- Occurrence - 1-based mutation occurrence index to target within selected predicate clauses.
- PrintMutation - Boolean flag to print the original and mutated term plus source location.

Hook object implementing the head_arguments_reordering mutator by swapping the first two arguments in matching rule or grammar rule heads.

Availability:
logtalk_load(mutation_testing(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-20

Compilation flags:
static, context_switching_calls

Implements:
public expanding
public clause_mutator_protocol
Imports:
public mutator_common
Uses:
list

Remarks:
(none)

Inherited public predicates:
coverage_clause_mutator/0 goal_expansion/2 mutation/2 reset/0 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.158.9 mutation_testing

Mutation testing tool.

Availability:
logtalk_load(mutation_testing(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-03

Compilation flags:
static, context_switching_calls

Imports:
public options
Provides:
logtalk::message_hook/4
Uses:
fast_random(Algorithm)
json(ObjectRepresentation,PairRepresentation,StringRepresentation)
list
logtalk
os
term_io
type
user

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 default_option/1 default_options/1 option/2 option/3 valid_option/1 valid_options/1

- Public predicates
  - library/1
  - library/2
  - directory/1
  - directory/2

    - entity/1
    - entity/2
    - predicate/2
    - predicate/3
    - report_entity/3
    - report_predicate/4
    - report_library/3
    - report_directory/3
    - format_report/3
    - format_report/2
    - format_report/1
    - entity_mutants/2
    - entity_mutants/3
    - predicate_mutants/3
    - predicate_mutants/4
    - library_mutants/2
    - library_mutants/3
    - directory_mutants/2
    - directory_mutants/3
- Protected predicates
- Private predicates
    - probe_mutation_happened_/0
    - probing_/0
    - capturing_mutated_terms_/0
    - captured_mutated_terms_/5
    - baseline_coverage_/4
- Operators

Public predicates

library/1

Runs mutation testing for all loaded entities from a given library using default options.

Compilation flags:
    static

Template:
library(Library)
Mode and number of proofs:
library(+atom) - zero_or_one

---

library/2

Runs mutation testing for all loaded entities from a given library using the given options.

Compilation flags:
static

Template:
library(Library,Options)
Mode and number of proofs:
library(+atom,+list(compound)) - zero_or_one

---

directory/1

Runs mutation testing for all loaded entities from a given directory using default options.

Compilation flags:
static

Template:
directory(Directory)
Mode and number of proofs:
directory(+atom) - zero_or_one

---

directory/2

Runs mutation testing for all loaded entities from a given directory using the given options.

Compilation flags:
    static

Template:
    directory(Directory,Options)
Mode and number of proofs:
    directory(+atom,+list(compound)) - zero_or_one

---

entity/1

Runs mutation testing for a loaded entity using default options.

Compilation flags:
    static

Template:
    entity(Entity)
Mode and number of proofs:
    entity(+entity_identifier) - zero_or_one

---

entity/2

Runs mutation testing for a loaded entity using the given options.

Compilation flags:
    static

Template:
    entity(Entity,Options)
Mode and number of proofs:
    entity(+entity_identifier,+list(compound)) - zero_or_one

---

predicate/2

Runs mutation testing for a loaded entity predicate using default options.

Compilation flags:
    static

Template:
    predicate(Entity,Predicate)
Mode and number of proofs:
    predicate(+entity_identifier,+predicate_indicator) - zero_or_one

---

predicate/3

Runs mutation testing for a loaded entity predicate using the given options.

Compilation flags:
    static

Template:
    predicate(Entity,Predicate,Options)
Mode and number of proofs:
    predicate(+entity_identifier,+predicate_indicator,+list(compound)) - zero_or_one

---

report_entity/3

Runs mutation testing for an entity and returns a structured report term.

Compilation flags:
    static

Template:
    report_entity(Entity,Report,Options)
Mode and number of proofs:
    report_entity(+entity_identifier,-compound,+list(compound)) - zero_or_one

---

report_predicate/4

Runs mutation testing for an entity predicate and returns a structured report term.

Compilation flags:
    static

Template:
    report_predicate(Entity,Predicate,Report,Options)
Mode and number of proofs:
    report_predicate(+entity_identifier,+predicate_indicator,-compound,+list(compound)) - zero_or_one

---

report_library/3

Runs mutation testing for loaded entities from a library and returns structured report terms.

Compilation flags:
    static

Template:
    report_library(Library,Report,Options)
Mode and number of proofs:
    report_library(+atom,-compound,+list(compound)) - zero_or_one

---

report_directory/3

Runs mutation testing for loaded entities from a directory and returns structured report terms.

Compilation flags:
    static

Template:
    report_directory(Directory,Report,Options)
Mode and number of proofs:
    report_directory(+atom,-compound,+list(compound)) - zero_or_one

---

format_report/3

Formats a mutation testing report term to the given stream using the given format.

Compilation flags:
    static

Template:
    format_report(Stream,Format,Report)
Mode and number of proofs:
    format_report(+stream_or_alias,+atom,+compound) - one

---

format_report/2

Formats a mutation testing report term to the current output stream using the given format.

Compilation flags:
    static

Template:
    format_report(Format,Report)
Mode and number of proofs:
    format_report(+atom,+compound) - one

---

format_report/1

Formats a mutation testing report term to the current output stream using the text format.

Compilation flags:
    static

Template:
    format_report(Report)
Mode and number of proofs:

format_report(+compound) - one

---

entity_mutants/2

Returns the deterministic list of mutants for an entity using default options.

Compilation flags:
static

Template:
entity_mutants(Entity,Mutants)
Mode and number of proofs:
entity_mutants(+entity_identifier,-list(compound)) - zero_or_one

---

entity_mutants/3

Returns the deterministic list of mutants for an entity using the given options.

Compilation flags:
static

Template:
entity_mutants(Entity,Mutants,Options)
Mode and number of proofs:
entity_mutants(+entity_identifier,-list(compound),+list(compound)) - zero_or_one

---

predicate_mutants/3

Returns the deterministic list of mutants for an entity predicate using default options.

Compilation flags:
static

Template:

predicate_mutants(Entity,Predicate,Mutants)
Mode and number of proofs:
predicate_mutants(+entity_identifier,+predicate_indicator,-list(compound)) - zero_or_one

---

predicate_mutants/4

Returns the deterministic list of mutants for an entity predicate using the given options.

Compilation flags:
static

Template:
predicate_mutants(Entity,Predicate,Mutants,Options)
Mode and number of proofs:
predicate_mutants(+entity_identifier,+predicate_indicator,-list(compound),+list(compound)) - zero_or_one

---

library_mutants/2

Returns the deterministic list of mutants for loaded entities from a given library using default options.

Compilation flags:
static

Template:
library_mutants(Library,Mutants)
Mode and number of proofs:
library_mutants(+atom,-list(compound)) - zero_or_one

---

library_mutants/3

Returns the deterministic list of mutants for loaded entities from a given library using the given options.

Compilation flags:
    static

Template:
    library_mutants(Library,Mutants,Options)
Mode and number of proofs:
    library_mutants(+atom,-list(compound),+list(compound)) - zero_or_one

---

directory_mutants/2

Returns the deterministic list of mutants for loaded entities from a given directory using default options.

Compilation flags:
    static

Template:
    directory_mutants(Directory,Mutants)
Mode and number of proofs:
    directory_mutants(+atom,-list(compound)) - zero_or_one

---

directory_mutants/3

Returns the deterministic list of mutants for loaded entities from a given directory using the given options.

Compilation flags:
    static

Template:
    directory_mutants(Directory,Mutants,Options)
Mode and number of proofs:
    directory_mutants(+atom,-list(compound),+list(compound)) - zero_or_one

---

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

probe_mutation_happened_/0

True iff a mutation happened.

Compilation flags:
    dynamic

Mode and number of proofs:
    probe_mutation_happened_ - zero_or_one

---

probing_/0

True iff we are currently probing for mutations (suppresses printing).

Compilation flags:
    dynamic

Mode and number of proofs:
    probing_ - zero_or_one

---

capturing_mutated_terms_/0

True iff we are capturing original and mutated terms while formatting reports.

Compilation flags:
    dynamic

Mode and number of proofs:
    capturing_mutated_terms_ - zero_or_one

---

captured_mutated_terms_/5

Captured original and mutated terms, variable names, and source location for one mutant.

Compilation flags:
dynamic

Template:
captured_mutated_terms_(Original,Mutation,Variables,File,Lines)
Mode and number of proofs:
captured_mutated_terms_(-callable,-callable,-list,-atom,-compound) - zero_or_one

---

baseline_coverage_/4

Coverage baseline cache for an entity and predicate: covered clauses and total clauses.

Compilation flags:
dynamic

Template:
baseline_coverage_(Entity,Predicate,CoveredClauses,TotalClauses)
Mode and number of proofs:
baseline_coverage_(?entity_identifier,?predicate_indicator,?list(integer),?integer) - zero_or_more

---

Operators

(none)

category

1.158.10 mutation_testing_messages

Default message translations for the mutation_testing tool.

Availability:
logtalk_load(mutation_testing(loader))

Author: Paulo Moura

Version: 1:0:0
Date: 2026-03-06

Compilation flags:
static

Provides:
logtalk::message_prefix_stream/4
logtalk::message_tokens//2

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.158.11 mutator_common

Mutator common predicate utilities.

Availability:
logtalk_load(mutation_testing(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-20

Compilation flags:
static

Implements:
public mutator_protocol
Uses:
logtalk

Remarks:
(none)

Inherited public predicates:
coverage_clause_mutator/0 mutation/2 reset/0

- Public predicates
- Protected predicates
  - print_mutation/3
  - target_predicate/3
  - target_predicate_clause_index/4
  - target_scope_directive/3
  - target_scope_directive_index/4
  - target_predicate_directive/3

    - target_predicate_directive_index/4
    - target_uses_directive/3
    - target_uses_directive_index/4
    - next_occurrence/1
- Private predicates
    - current_predicate_clause_index_/2
    - update_target_predicate_clause_index_/2
    - current_scope_directive_index_/2
    - update_target_scope_directive_index_/2
    - current_predicate_directive_index_/2
    - update_target_predicate_directive_index_/2
    - current_uses_directive_index_/2
    - update_target_uses_directive_index_/2
    - seen_/1
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

print_mutation/3

Prints a term and its mutation when Flag is true. Succeeds otherwise.

Compilation flags:
    static

Template:
    print_mutation(Flag,Original,Mutation)
Mode and number of proofs:
    print_mutation(+boolean,@callable,@callable) - one

target_predicate/3

True iff Term is a candidate for mutation.

Compilation flags:
    static

Template:
    target_predicate(Term,Entity,Predicate)
Mode and number of proofs:
    target_predicate(@callable,@entity_identifier,@predicate_indicator) - one

---

target_predicate_clause_index/4

True iff Term is a candidate for mutation while also returning its current 1-based contiguous clause index for the matching predicate or non-terminal.

Compilation flags:
    static

Template:
    target_predicate_clause_index(Term,Entity,Predicate,ClauseIndex)
Mode and number of proofs:
    target_predicate_clause_index(@callable,@entity_identifier,@predicate_indicator,-integer) - zero_or_one

---

target_scope_directive/3

True iff Term is a matching predicate or non-terminal scope directive candidate for mutation.

Compilation flags:
    static

Template:
    target_scope_directive(Term,Entity,Predicate)
Mode and number of proofs:
    target_scope_directive(@callable,@entity_identifier,@predicate_indicator) - one

---

target_scope_directive_index/4

True iff Term is a matching predicate or non-terminal scope directive candidate for mutation while also returning its 1-based index among matching scope directives for the selected predicate or non-terminal.

Compilation flags:
    static

Template:
    target_scope_directive_index(Term,Entity,Predicate,DirectiveIndex)
Mode and number of proofs:
    target_scope_directive_index(@callable,@entity_identifier,@predicate_indicator,-integer) - zero_or_one

---

target_predicate_directive/3

True iff Term is a matching predicate or non-terminal directive candidate for mutation.

Compilation flags:
    static

Template:
    target_predicate_directive(Term,Entity,Predicate)
Mode and number of proofs:
    target_predicate_directive(@callable,@entity_identifier,@predicate_indicator) - one

---

target_predicate_directive_index/4

True iff Term is a matching predicate or non-terminal directive candidate for mutation while also returning its 1-based index among matching directives for the selected predicate or non-terminal.

Compilation flags:
    static

Template:

target_predicate_directive_index(Term,Entity,Predicate,DirectiveIndex)
Mode and number of proofs:
target_predicate_directive_index(@callable,@entity_identifier,@predicate_indicator,-integer) - zero_or_one

---

target_uses_directive/3

True iff Term is a matching uses/2 directive candidate for mutation.

Compilation flags:
static

Template:
target_uses_directive(Term,Entity,Predicate)
Mode and number of proofs:
target_uses_directive(@callable,@entity_identifier,@predicate_indicator) - one

---

target_uses_directive_index/4

True iff Term is a matching uses/2 directive candidate for mutation while also returning its 1-based index among matching directives for the selected predicate or non-terminal.

Compilation flags:
static

Template:
target_uses_directive_index(Term,Entity,Predicate,DirectiveIndex)
Mode and number of proofs:
target_uses_directive_index(@callable,@entity_identifier,@predicate_indicator,-integer) - zero_or_one

---

next_occurrence/1

Next mutation occurrence.

Compilation flags:
    static

Template:
    next_occurrence(Occurrence)
Mode and number of proofs:
    next_occurrence(-integer) - one

Private predicates

current_predicate_clause_index_/2

Table of current clause indexes per predicate.

Compilation flags:
    dynamic

Template:
    current_predicate_clause_index_(Predicate,ClauseIndex)
Mode and number of proofs:
    current_predicate_clause_index_(?predicate_indicator,?integer) - zero_or_one

---

update_target_predicate_clause_index_/2

Updates and returns the next clause index for the given predicate.

Compilation flags:
    static

Template:
    update_target_predicate_clause_index_(Predicate,ClauseIndex)
Mode and number of proofs:
    update_target_predicate_clause_index_(@predicate_indicator,-integer) - one

---

current_scope_directive_index_/2

Table of current scope directive indexes per predicate.

Compilation flags:
    dynamic

Template:
    current_scope_directive_index_(Predicate,DirectiveIndex)
Mode and number of proofs:
    current_scope_directive_index_(?predicate_indicator,?integer) - zero_or_one

---

update_target_scope_directive_index_/2

Updates and returns the next scope directive index for the given predicate.

Compilation flags:
    static

Template:
    update_target_scope_directive_index_(Predicate,DirectiveIndex)
Mode and number of proofs:
    update_target_scope_directive_index_(@predicate_indicator,-integer) - one

---

current_predicate_directive_index_/2

Table of current predicate directive indexes per predicate.

Compilation flags:
    dynamic

Template:
    current_predicate_directive_index_(Predicate,DirectiveIndex)
Mode and number of proofs:

current_predicate_directive_index_(?predicate_indicator,?integer) - zero_or_one

---

update_target_predicate_directive_index_/2

Updates and returns the next predicate directive index for the given predicate.

Compilation flags:
static

Template:
update_target_predicate_directive_index_(Predicate,DirectiveIndex)
Mode and number of proofs:
update_target_predicate_directive_index_(@predicate_indicator,-integer) - one

---

current_uses_directive_index_/2

Table of current uses directive indexes per predicate.

Compilation flags:
dynamic

Template:
current_uses_directive_index_(Predicate,DirectiveIndex)
Mode and number of proofs:
current_uses_directive_index_(?predicate_indicator,?integer) - zero_or_one

---

update_target_uses_directive_index_/2

Updates and returns the next uses directive index for the given predicate.

Compilation flags:
static

Template:

update_target_uses_directive_index_(Predicate,DirectiveIndex)
Mode and number of proofs:
update_target_uses_directive_index_(@predicate_indicator,-integer) - one

---

seen_/1

Table of last seen mutation occurrence.

Compilation flags:
dynamic

Template:
seen_(Occurrence)
Mode and number of proofs:
seen_(?integer) - zero_or_one

Operators

(none)

protocol

1.158.12 mutator_protocol

Mutator protocol.

Availability:
logtalk_load(mutation_testing(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-07

Compilation flags:
static

Dependencies:

(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - reset/0
  - mutation/2
  - coverage_clause_mutator/0
- Protected predicates
- Private predicates
- Operators

Public predicates

reset/0

Resets any mutator internal state used while expanding terms.

Compilation flags:
static

Mode and number of proofs:
reset - one

---

mutation/2

Generates by backtracking zero or more mutations for a given term.

Compilation flags:
static

Template:

mutation(Term,Mutation)
Mode and number of proofs:
mutation(@callable,@callable) - zero_or_more

---

coverage_clause_mutator/0

True when mutation occurrences map directly to predicate clause numbers and can use baseline clause coverage for skipping uncovered mutants.

Compilation flags:
static

Mode and number of proofs:
coverage_clause_mutator - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

object

1.158.13 predicate_directive_suppression(Entity,Predicate,DirectiveIndex,Occurrence,PrintMutation)

- Entity - Identifier of the entity being mutated.
- Predicate - Predicate or non-terminal indicator selecting directives.
- DirectiveIndex - 1-based index for the selected matching directive.
- Occurrence - 1-based mutation occurrence index to target within selected matching directives.
- PrintMutation - Boolean flag to print the original and mutated term plus source location.

Hook object implementing the predicate_directive_suppression mutator by suppressing matching predicate or non-terminal directives.

Availability:
logtalk_load(mutation_testing(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-20

Compilation flags:
static, context_switching_calls

Implements:
public expanding
public directive_mutator_protocol
Imports:
public mutator_common

Remarks:
(none)

Inherited public predicates:
coverage_clause_mutator/0 goal_expansion/2 mutation/2 reset/0 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.158.14 relational_operator_replacement(Entity,Predicate,ClauseIndex,Occurrence,PrintMutation)

- Entity - Identifier of the entity being mutated.
- Predicate - Predicate or non-terminal indicator selecting clauses to mutate.
- ClauseIndex - 1-based clause index for the selected mutation.
- Occurrence - 1-based mutation occurrence index to target within selected predicate clauses.
- PrintMutation - Boolean flag to print the original and mutated term plus source location.

Hook object implementing the relational_operator_replacement mutator for matching predicate clauses.

Availability:
logtalk_load(mutation_testing(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-20

Compilation flags:
static, context_switching_calls

Implements:
public expanding
public clause_mutator_protocol
Imports:
public mutator_common

Remarks:
(none)

Inherited public predicates:
coverage_clause_mutator/0 goal_expansion/2 mutation/2 reset/0 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.158.15 scope_directive_replacement(Entity,Predicate,DirectiveIndex,Occurrence,PrintMutation)

- Entity - Identifier of the entity being mutated.
- Predicate - Predicate or non-terminal indicator selecting directives.
- DirectiveIndex - 1-based index for the selected matching scope directive.
- Occurrence - 1-based mutation occurrence index selecting an alternative visibility for the directive.
- PrintMutation - Boolean flag to print the original and mutated term plus source location.

Hook object implementing the scope_directive_replacement mutator by replacing matching scope directives with an alternative visibility.

Availability:
logtalk_load(mutation_testing(loader))

Author: Paulo Moura

Version: 1:0:0
Date: 2026-03-20

Compilation flags:
static, context_switching_calls

Implements:
public expanding
public directive_mutator_protocol
Imports:
public mutator_common

Remarks:
(none)

Inherited public predicates:
coverage_clause_mutator/0 goal_expansion/2 mutation/2 reset/0 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.158.16 subprocess_coverage_hook

Subprocess coverage hook object. Loaded in baseline coverage subprocesses to capture lgtunit coverage and persist it to a file.

Availability:
logtalk_load(mutation_testing(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-07

Compilation flags:
static, context_switching_calls

Provides:
logtalk::message_hook/4

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - load_coverage_config/1
- Protected predicates
- Private predicates
  - coverage_file_/1
  - coverage_entry_/4
- Operators

Public predicates

load_coverage_config/1

Loads a baseline coverage config file and enables coverage collection in the subprocess.

Compilation flags:
    static

Template:
    load_coverage_config(ConfigFile)
Mode and number of proofs:
    load_coverage_config(+atom) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

coverage_file_/1

Holds the baseline coverage output file path for the current subprocess.

Compilation flags:
    dynamic

Template:
    coverage_file_(CoverageFile)
Mode and number of proofs:
    coverage_file_(?atom) - zero_or_more

---

coverage_entry_/4

Captured lgtunit coverage for one entity predicate: covered clauses and total clauses.

Compilation flags:
dynamic

Template:
coverage_entry_(Entity,Predicate,CoveredClauses,TotalClauses)
Mode and number of proofs:
coverage_entry_(?entity_identifier,?predicate_indicator,?list(integer),?integer) - zero_or_more

---

Operators

(none)

object

1.158.17 subprocess_mutation_hook

Subprocess mutation hook object. Loaded in mutation testing subprocesses to set up the mutator hook and record test results to a status file.

Availability:
logtalk_load(mutation_testing(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-20

Compilation flags:
static, context_switching_calls

Provides:
logtalk::message_hook/4
Uses:
fast_random(Algorithm)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - load_config/1
- Protected predicates
- Private predicates
  - status_file_/1
  - failed_count_/1
- Operators

Public predicates

load_config/1

Loads a mutation config file and sets up the mutator hook for the subprocess.

Compilation flags:
static

Template:
load_config(ConfigFile)
Mode and number of proofs:
load_config(+atom) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

status_file_/1

Holds the status file path for the current subprocess.

Compilation flags:
dynamic

Template:
status_file_(StatusFile)
Mode and number of proofs:
status_file_(?atom) - zero_or_more

---

failed_count_/1

Accumulates the number of failed tests reported by lgtunit.

Compilation flags:
dynamic

Template:
failed_count_(Failed)
Mode and number of proofs:
failed_count_(?integer) - zero_or_more

Operators

(none)

object

1.158.18 truth_literal_flip(Entity,Predicate,ClauseIndex,Occurrence,PrintMutation)

- Entity - Identifier of the entity being mutated.
- Predicate - Predicate or non-terminal indicator selecting clauses to mutate.
- ClauseIndex - 1-based clause index for the selected mutation (equal to Occurrence for this mutator).
- Occurrence - 1-based mutation occurrence index to target within selected predicate clauses.
- PrintMutation - Boolean flag to print the original and mutated term plus source location.

Hook object implementing the truth_literal_flip mutator for matching predicate clauses.

Availability:
logtalk_load(mutation_testing(loader))

Author: Paulo Moura
Version: 1:0:0

Date: 2026-03-20

Compilation flags:
    static, context_switching_calls

Implements:
    public expanding
    public clause_mutator_protocol
Imports:
    public mutator_common

Remarks:
    (none)

Inherited public predicates:
    coverage_clause_mutator/0 goal_expansion/2 mutation/2 reset/0 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.158.19 uses_directive_resource_deletion(Entity,Predicate,DirectiveIndex,Occurrence,PrintMutation)

- Entity - Identifier of the entity being mutated.
- Predicate - Predicate or non-terminal indicator selecting resources.
- DirectiveIndex - 1-based index for the selected matching uses/2 directive.
- Occurrence - 1-based mutation occurrence index selecting which matching resource to delete.
- PrintMutation - Boolean flag to print the original and mutated term plus source location.

Hook object implementing the uses_directive_resource_deletion mutator by deleting one matching resource from a uses/2 directive.

Availability:
logtalk_load(mutation_testing(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-20

Compilation flags:
static, context_switching_calls

Implements:
public expanding
public directive_mutator_protocol
Imports:
public mutator_common

Remarks:
(none)

Inherited public predicates:
coverage_clause_mutator/0 goal_expansion/2 mutation/2 reset/0 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.159 mutations

object

1.159.1 default_atom_mutations

Default atom mutations.

Availability:
logtalk_load(mutations(loader))

Author: Paulo Moura
Version: 0:1:0
Date: 2023-11-24

Compilation flags:
static, context_switching_calls

Provides:

mutations_store::mutation/4

Uses:

fast_random
list
type

Remarks:

(none)

Inherited public predicates:

(none)

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

type

object

1.159.2 default_compound_mutations

Default compound mutations.

Availability:
logtalk_load(mutations(loader))

Author: Paulo Moura
Version: 0:1:0
Date: 2023-11-23

Compilation flags:
static, context_switching_calls

Provides:
mutations_store::mutation/4
Uses:
mutations_store

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

type

object

1.159.3 default_float_mutations

Default float mutations.

Availability:
logtalk_load(mutations(loader))

Author: Paulo Moura
Version: 0:1:0
Date: 2023-11-23

Compilation flags:
static, context_switching_calls

Provides:
mutations_store::mutation/4

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

type

object

1.159.4 default_integer_mutations

Default integer mutations.

Availability:
logtalk_load(mutations(loader))

Author: Paulo Moura
Version: 0:1:0
Date: 2023-11-24

Compilation flags:
static, context_switching_calls

Provides:
mutations_store::mutation/4
Uses:
fast_random
list

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

type

object

1.159.5 default_list_mutations

Default list mutations.

Availability:
logtalk_load(mutations(loader))

Author: Paulo Moura
Version: 0:1:0
Date: 2023-11-24

Compilation flags:
static, context_switching_calls

Provides:
mutations_store::mutation/4
Uses:
fast_random
list
mutations_store

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

type

category

1.159.6 mutations

Adds mutations support to the library type object.

Availability:
logtalk_load(mutations(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2023-11-23

Compilation flags:
static

Complements:
type
Uses:
mutations_store

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - mutation/3
- Protected predicates
- Private predicates
- Operators

Public predicates

mutation/3

Returns a random mutation of a term into another term of the same type. The input Term is assume to be valid for the given Type.

Compilation flags:
static

Template:
mutation(Type,Term,Mutation)
Mode and number of proofs:
mutation(@callable,@term,-term) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.159.7 mutations_store

Stores mutation definitions for selected types. User extensible by defining objects or categories defining clauses for the mutation/3 predicate and using this object as a hook object for their compilation.

Availability:
logtalk_load(mutations(loader))

Author: Paulo Moura
Version: 0:1:0
Date: 2023-11-23

Compilation flags:
static, context_switching_calls

Implements:
public expanding
Uses:
fast_random

Remarks:
(none)

Inherited public predicates:
goal_expansion/2 term_expansion/2

- Public predicates
  - mutation/3
  - counter/2
- Protected predicates
- Private predicates
  - mutation/4
  - counter_/2

- Operators

Public predicates

mutation/3

Returns a random mutation of a term into another term of the same type. The input Term is assumed to be valid for the given Type.

Compilation flags:
    static

Template:
    mutation(Type,Term,Mutation)
Mode and number of proofs:
    mutation(@callable,@term,-term) - one

---

counter/2

Table of the number of mutations available per type.

Compilation flags:
    static

Template:
    counter(Type,N)
Mode and number of proofs:
    counter(?callable,?positive_integer) - zero_or_more

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

mutation/4

Returns a random mutation of a term into another term of the same type using mutator N. The input Term is assume to be valid for the given Type.

Compilation flags:
    static, multifile

Template:
    mutation(Type,N,Term,Mutation)
Mode and number of proofs:
    mutation(?callable,?positive_integer,@term,-term) - zero_or_more

---

counter_/2

Internal counter for the number of mutations available for a given type.

Compilation flags:
    dynamic

Template:
    counter_(Type,N)
Mode and number of proofs:
    counter_(?callable,?positive_integer) - zero_or_more

Operators

(none)

See also

type

1.160 naive_bayes_classifier

object

1.160.1 naive_bayes_classifier

Naive Bayes classifier with Laplace smoothing and Gaussian distribution support. Learns from a dataset object implementing the dataset_protocol protocol and returns a classifier term that can be used for prediction and exported as predicate clauses.

Availability:
logtalk_load(naive_bayes_classifier(loader))

Author: Paulo Moura
Version: 2:0:0
Date: 2026-05-11

Compilation flags:
static, context_switching_calls

Imports:
public probabilistic_classifier_common
Uses:
format
list
pairs
population
type

Remarks:
(none)

Inherited public predicates:

check_classifier/1 check_option/1 check_options/1 classifier_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 predict_probabilities/3 print_classifier/1 valid_classifier/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

dataset_protocol, isolation_forest_anomaly_detector, c45_classifier, knn_classifier, nearest_centroid_classifier, random_forest_classifier, adaptive_boosting_classifier

1.161 nanoid

object

1.161.1 nanoid

NanoID generator using atom representation, 21 symbols, and the standard URL alphabet.

Availability:
logtalk_load(nanoid(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-26

Compilation flags:
static, context_switching_calls

Extends:
public nanoid(atom,21,_-0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ)

Remarks:
(none)

Inherited public predicates:
generate/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

nanoid(Representation,Size,Alphabet), ids, ulid, uuid

object

1.161.2 nanoid(Representation,Size,Alphabet)

- Representation - Text representation for the NanoID. Possible values are atom, chars, and codes.
- Size - Number of symbols in the NanoID.
- Alphabet - Alphabet used for generating NanoIDs represented as an atom, list of characters, or list of character codes.

NanoID generator.

Availability:
logtalk_load(nanoid(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-26

Compilation flags:
static, context_switching_calls

Implements:
public nanoid_protocol
Uses:
fast_random(Algorithm)
list
os

Remarks:
(none)

Inherited public predicates:
generate/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

nanoid, ids(Representation,Bytes), ulid(Representation), uuid(Representation)

protocol

1.161.3 nanoid_protocol

NanoID generator protocol.

Availability:
logtalk_load(nanoid(loader))

Author: Paulo Moura
Version: 0:1:0
Date: 2026-02-26

Compilation flags:
    static

Dependencies:
    (none)

Remarks:
    (none)

Inherited public predicates:
    (none)

- Public predicates
  - generate/1
- Protected predicates
- Private predicates
- Operators

Public predicates

generate/1

Returns a NanoID.

Compilation flags:
    static

Template:
    generate(NanoID)
Mode and number of proofs:
    generate(--ground) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.162 nearest_centroid_classifier

object

1.162.1 nearest_centroid_classifier

Nearest Centroid classifier with multiple distance metrics. Learns from a dataset object implementing the dataset_protocol protocol and returns a classifier term that can be used for prediction and exported as predicate clauses.

Availability:
logtalk_load(nearest_centroid_classifier(loader))

Author: Paulo Moura
Version: 2:0:0
Date: 2026-05-11

Compilation flags:
static, context_switching_calls

Imports:
public probabilistic_classifier_common
Uses:
format
linear_algebra
list
numberlist
pairs
type

Remarks:

(none)

Inherited public predicates:

check_classifier/1 check_option/1 check_options/1 classifier_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 predict_probabilities/3 print_classifier/1 valid_classifier/1 valid_option/1 valid_options/1

- Public predicates
  - predict/4
  - predict_probabilities/4
- Protected predicates
- Private predicates
- Operators

Public predicates

predict/4

Predicts the class label for a new instance using the learned classifier and the given options. The instance is a list of Attribute-Value pairs.

Compilation flags:

static

Template:

predict(Classifier,Instance,Class,Options)

Mode and number of proofs:

predict(+compound,+list,-atom,+list(compound)) - one

predict_probabilities/4

Predicts class probabilities for a new instance using the learned classifier and the given options. Returns a list of Class-Probability pairs. The instance is a list of Attribute-Value pairs.

Compilation flags:
    static

Template:
    predict_probabilities(Classifier,Instance,Probabilities,Options)
Mode and number of proofs:
    predict_probabilities(+compound,+list,-list,+list(compound)) - one

---

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

dataset_protocol, isolation_forest_anomaly_detector, c45_classifier, knn_classifier, naive_bayes_classifier, random_forest_classifier, adaptive_boosting_classifier

1.163 nested_dictionaries

object

1.163.1 navltree

Nested dictionary implementation based on the AVL tree implementation. Uses standard order to compare keys.

Availability:
logtalk_load(nested_dictionaries(loader))

Author: Paul Brown and Paulo Moura.
Version: 0:1:0
Date: 2021-04-09

Compilation flags:
static, context_switching_calls

Implements:
public nested_dictionary_protocol
Extends:
private avltree

Remarks:
(none)

Inherited public predicates:
as_curly_bracketed/2 as_nested_dictionary/2 delete_in/4 empty/1 insert_in/4 lookup_in/3 new/1 update_in/4 update_in/5

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

nrbtree, nbintree

object

1.163.2 nbintree

Nested dictionary implementation based on the simple binary tree implementation. Uses standard order to compare keys.

Availability:
logtalk_load(nested_dictionaries(loader))

Author: Paul Brown and Paulo Moura.
Version: 0:1:0
Date: 2021-04-09

Compilation flags:
static, context_switching_calls

Implements:
public nested_dictionary_protocol
Extends:
private bintree

Remarks:

(none)

Inherited public predicates:
as_curly_bracketed/2 as_nested_dictionary/2 delete_in/4 empty/1 insert_in/4 lookup_in/3 new/1 update_in/4 update_in/5

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

nrbtree, navltree

protocol

1.163.3 nested_dictionary_protocol

Nested dictionary protocol.

Availability:
logtalk_load(nested_dictionaries(loader))

Author: Paul Brown and Paulo Moura

Version: 0:1:0
Date: 2021-04-07

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - new/1
  - empty/1
  - as_nested_dictionary/2
  - as_curly_bracketed/2
  - lookup_in/3
  - update_in/4
  - update_in/5
  - insert_in/4
  - delete_in/4
- Protected predicates
- Private predicates
- Operators

Public predicates

new/1

Create an empty (nested) dictionary.

Compilation flags:
static

Template:
    new(Dictionary)
Mode and number of proofs:
    new(--dictionary) - one

---

empty/1

True iff the dictionary is empty.

Compilation flags:
    static

Template:
    empty(Dictionary)
Mode and number of proofs:
    empty(@dictionary) - zero_or_one

---

as_nested_dictionary/2

Creates a (nested) dictionary term from a curly-brackted term representation.

Compilation flags:
    static

Template:
    as_nested_dictionary(Term,Dictionary)
Mode and number of proofs:
    as_nested_dictionary(++term,--dictionary) - one_or_error

---

as_curly_bracketed/2

Creates a a curly-brackted term representation from a (nested) dictionary.

Compilation flags:
    static

Template:
    as_curly_bracketed(Dictionary,Term)
Mode and number of proofs:
    as_curly_bracketed(+dictionary,--term) - one_or_error

---

lookup_in/3

Lookup a chain of keys in a nested dictionary. Unifies Value with Dictionary when Keys is the empty list.

Compilation flags:
    static

Template:
    lookup_in(Keys,Value,Dictionary)
Mode and number of proofs:
    lookup_in(++list(ground),?term,+dictionary) - zero_or_more

---

update_in/4

Updates the value found by traversing through the nested keys.

Compilation flags:
    static

Template:
    update_in(OldDictionary,Keys,Value,NewDictionary)
Mode and number of proofs:
    update_in(+dictionary,++list(ground),++term,--dictionary) - zero_or_one

---

update_in/5

Updates the value found by traversing through the nested keys, only succeeding if the value found after traversal matches the old value.

Compilation flags:
    static

Template:
    update_in(OldDictionary,Keys,OldValue,NewValue,NewDictionary)
Mode and number of proofs:
    update_in(+dictionary,++list(ground),?term,++term,--dictionary) - zero_or_one

---

insert_in/4

Inserts a key-value pair into a dictionary by traversing through the nested keys. When the key already exists, the associated value is updated.

Compilation flags:
    static

Template:
    insert_in(OldDictionary,Keys,Value,NewDictionary)
Mode and number of proofs:
    insert_in(+dictionary,++list(ground),++term,--dictionary) - zero_or_one

---

delete_in/4

Deletes a matching key-value pair from a dictionary by traversing through the nested keys, returning the updated dictionary.

Compilation flags:
    static

Template:
    delete_in(OldDictionary,Keys,Value,NewDictionary)
Mode and number of proofs:

delete_in(+dictionary,++list(ground),?term,--dictionary) - zero_or_one

---

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

navltree, nbintree, nrbtree

object

1.163.4 nrbtree

Nested dictionary implementation based on the red-black tree implementation. Uses standard order to compare keys.

Availability:
logtalk_load(nested_dictionaries(loader))

Author: Paul Brown and Paulo Moura.
Version: 0:1:0
Date: 2021-04-09

Compilation flags:
static, context_switching_calls

Implements:
public nested_dictionary_protocol
Extends:
private rbtree

Remarks:

(none)

Inherited public predicates:

as_curly_bracketed/2 as_nested_dictionary/2 delete_in/4 empty/1 insert_in/4 lookup_in/3 new/1 update_in/4 update_in/5

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

navltree, nbintree

1.164 nmea

object

1.164.1 nmea

Parser for NMEA 0183 sentences with typed semantic decoding for selected GPS/GNSS sentence types.

Availability:
logtalk_load(nmea(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-10

Compilation flags:
static, context_switching_calls

Implements:
public nmea_protocol
Imports:
public options
Uses:
date
list
reader
time

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 checksum/2 data/2 default_option/1 default_options/1 fields/2 option/2 option/3 parse/2 parse/3 sentence_type/2 talker/2 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.164.2 nmea_protocol

Protocol for parsing NMEA 0183 sentences and projecting supported sentence types into typed semantic terms.

Availability:
logtalk_load(nmea(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-10

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - parse/2
  - parse/3
  - talker/2
  - sentence_type/2
  - fields/2
  - checksum/2
  - data/2
- Protected predicates
- Private predicates
- Operators

Public predicates

parse/2

Parses NMEA sentences from a source specification into canonical raw sentence terms. Supported source specifications are atom(Atom), chars(List), codes(List), stream(Stream), and file(Path).

Compilation flags:
    static

Template:
    parse(Source,Sentences)
Mode and number of proofs:
    parse(++compound,--list(compound)) - one_or_error

---

parse/3

Parses NMEA sentences from a source specification using parser options. Supported options are checksum(required), checksum(optional), checksum(ignore), unknown_type(keep), and unknown_type(error).

Compilation flags:
    static

Template:

parse(Source,Options,Sentences)
Mode and number of proofs:
parse(++compound,++list(compound),--list(compound)) - one_or_error

---

talker/2

Returns the normalized talker identifier for a parsed sentence. Proprietary sentences use the atom proprietary.

Compilation flags:
static

Template:
talker(Sentence,Talker)
Mode and number of proofs:
talker(+compound,-atom) - one

---

sentence_type/2

Returns the normalized sentence type identifier for a parsed sentence.

Compilation flags:
static

Template:
sentence_type(Sentence,Type)
Mode and number of proofs:
sentence_type(+compound,-atom) - one

---

fields/2

Returns the lossless ordered list of sentence fields after the sentence identifier.

Compilation flags:
    static

Template:
    fields(Sentence,Fields)
Mode and number of proofs:
    fields(+compound,-list(atom)) - one

---

checksum/2

Returns the checksum information as checksum(Provided,Computed) where Provided is either a normalized two-digit hexadecimal atom or missing.

Compilation flags:
    static

Template:
    checksum(Sentence,Checksum)
Mode and number of proofs:
    checksum(+compound,-compound) - one

---

data/2

Projects a parsed sentence into a typed semantic term for the supported sentence types gga, rmc, gsa, gsv, vtg, and gll.

Compilation flags:
    static

Template:
    data(Sentence,Data)
Mode and number of proofs:
    data(+compound,-compound) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.165 nmf_projection

object

1.165.1 nmf_projection

Non-negative Matrix Factorization dimension reduction for non-negative continuous datasets.

Availability:
logtalk_load(nmf_projection(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public dimension_reducer_common
Uses:
format
linear_algebra
list
numberlist
type
user

Remarks:
(none)

Inherited public predicates:
check_dimension_reducer/1 check_option/1 check_options/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 dimension_reducer_options/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 print_dimension_reducer/1 transform/3 valid_dimension_reducer/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.166 optics_clusterer

object

1.166.1 optics_clusterer

OPTICS clusterer for continuous datasets. Learns an ordering from a dataset object implementing the clustering_dataset_protocol protocol and extracts clusters using an epsilon threshold so the result can be used with the standard clusterer protocol.

Availability:
logtalk_load(optics_clusterer(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public clusterer_common
public search_indexing
Uses:
avltree
binary_heap_min
format
list
numberlist
pairs
type

Remarks:
(none)

Inherited public predicates:
check_clusterer/1 check_option/1 check_options/1 cluster/3 clusterer_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 print_clusterer/1 valid_clusterer/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

clusterer_protocol, clustering_dataset_protocol, dbscan_clusterer

1.167 optionals

object

1.167.1 maybe

Types and predicates for type-checking and handling optional terms. Inspired by Haskell.

Availability:
logtalk_load(optionals(loader))

Author: Paulo Moura
Version: 0:9:0
Date: 2025-06-19

Compilation flags:
static, context_switching_calls

Provides:

type::type/1
type::check/2
arbitrary::arbitrary/1
arbitrary::arbitrary/2
arbitrary::shrinker/1
arbitrary::shrink/3
arbitrary::edge_case/2

Uses:

optional
optional(Optional)
random
type

Remarks:

- Type-checking support: Defines type maybe(Type) for checking optional terms where the value hold by the optional term must be of the given type.
- QuickCheck support: Defines clauses for the arbitrary::arbitrary/1-2, arbitrary::shrinker/1, arbitrary::shrink/3, and arbitrary::edge_case/2 predicates to allow generating random values for the maybe(Type) type.

Inherited public predicates:

(none)

- Public predicates
  - cat/2
  - sequence/2
  - traverse/3
- Protected predicates
- Private predicates
- Operators

Public predicates

cat/2

Returns the values stored in the non-empty optional terms.

Compilation flags:

static

Template:
cat(Optionals,Values)
Mode and number of proofs:
cat(+list(optional),-list) - one

---

sequence/2

Returns an optional term with a list of all values when all optional terms are not empty. Otherwise returns an empty optional term.

Compilation flags:
static

Template:
sequence(Optionals,Optional)
Mode and number of proofs:
sequence(+list(optional),--nonvar) - one

---

traverse/3

Applies a closure to each list element to generate optional terms and then sequences them into a single optional term holding all values or an empty optional term.

Compilation flags:
static

Template:
traverse(Closure,Terms,Optional)
Meta-predicate template:
traverse(2,*,*)
Mode and number of proofs:
traverse(+callable,+list,--nonvar) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

optional, optional(Optional), type, arbitrary

object

1.167.2 optional

Constructors for optional terms. An optional term is either empty or holds a value. Optional terms should be regarded as opaque terms and always used with the optional/1 object by passing the optional term as a parameter.

Availability:
logtalk_load(optionals(loader))

Author: Paulo Moura
Version: 2:2:0
Date: 2026-02-21

Compilation flags:
static, context_switching_calls

Provides:
type::type/1
type::check/2

Remarks:

- Type-checking support: This object also defines a type optional for use with the type library object.

Inherited public predicates:

(none)

- Public predicates
  - empty/1
  - of/2
  - from_goal/3
  - from_goal/2
  - from_generator/3
  - from_generator/2
  - from_goal_or_throw/3
  - from_goal_or_throw/2
- Protected predicates
- Private predicates
- Operators

Public predicates

empty/1

Constructs an empty optional term.

Compilation flags:
static

Template:
empty(Optional)
Mode and number of proofs:
empty(--nonvar) - one

of/2

Constructs an optional term holding the given value.

Compilation flags:
    static

Template:
    of(Value,Optional)
Mode and number of proofs:
    of(@term,--nonvar) - one

---

from_goal/3

Constructs an optional term holding a value bound by calling the given goal. Returns an empty optional term if the goal fails. Also returns an empty optional term if the goal throws an error, silently discarding the error information. Use from_goal_or_throw/3 if exceptions should be propagated or the expected library from_goal/3 predicate if error information should be preserved.

Compilation flags:
    static

Template:
    from_goal(Goal,Value,Optional)
Meta-predicate template:
    from_goal(0,*,*)
Mode and number of proofs:
    from_goal(+callable,--term,--nonvar) - one

---

from_goal/2

Constructs an optional term holding a value bound by calling the given closure. Returns an empty optional term if the closure fails. Also returns an empty optional term if the closure throws an error, silently discarding the error information. Use from_goal_or_throw/2 if exceptions should be propagated or the expected library from_goal/2 predicate if error information should be preserved.

Compilation flags:
    static

Template:
    from_goal(Closure,Optional)
Meta-predicate template:
    from_goal(1,*)
Mode and number of proofs:
    from_goal(+callable,--nonvar) - one

---

from_generator/3

Constructs optional terms with the values generated by calling the given goal. On goal failure, returns an empty optional. On goal error, also returns an empty optional, silently discarding the error information.

Compilation flags:
    static

Template:
    from_generator(Goal,Value,Optional)
Meta-predicate template:
    from_generator(0,*,*)
Mode and number of proofs:
    from_generator(+callable,--term,--nonvar) - one_or_more

---

from_generator/2

Constructs optional terms with the values generated by calling the given closure. On closure failure, returns an empty optional. On closure error, also returns an empty optional, silently discarding the error information.

Compilation flags:
    static

Template:
    from_generator(Closure,Optional)
Meta-predicate template:
    from_generator(1,*)
Mode and number of proofs:
    from_generator(+from_generator,--nonvar) - one_or_more

---

from_goal_or_throw/3

Constructs an optional term holding a value bound by calling the given goal. Returns an empty optional term if the goal fails. Propagates any exception thrown by the goal.

Compilation flags:
    static

Template:
    from_goal_or_throw(Goal,Value,Optional)
Meta-predicate template:
    from_goal_or_throw(0,*,*)
Mode and number of proofs:
    from_goal_or_throw(+callable,--term,--nonvar) - one_or_error

---

from_goal_or_throw/2

Constructs an optional term holding a value bound by calling the given closure. Returns an empty optional term if the closure fails. Propagates any exception thrown by the closure.

Compilation flags:
    static

Template:
    from_goal_or_throw(Closure,Optional)
Meta-predicate template:
    from_goal_or_throw(1,*)
Mode and number of proofs:
    from_goal_or_throw(+callable,--nonvar) - one_or_error

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

optional(Optional), type

object

1.167.3 optional(Optional)

Optional term handling predicates. Requires passing an optional term (constructed using the optional object predicates) as a parameter.

Availability:
logtalk_load(optionals(loader))

Author: Paulo Moura
Version: 1:9:0
Date: 2025-06-19

Compilation flags:
static, context_switching_calls

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - is_empty/0

  - is_present/0
  - if_empty/1
  - if_present/1
  - if_present_or_else/2
  - filter/2
  - map/2
  - flat_map/2
  - or/2
  - get/1
  - or_else/2
  - or_else_get/2
  - or_else_call/2
  - or_else_fail/1
  - or_else_throw/2
  - map_or_else/3
  - zip/3
  - flatten/1
  - to_expected/2
- Protected predicates
- Private predicates
- Operators

Public predicates

is_empty/0

True if the optional term is empty. See also the if_empty/1 predicate.

Compilation flags:
    static

Mode and number of proofs:
    is_empty - zero_or_one

is_present/0

True if the optional term holds a value. See also the if_present/1 predicate.

Compilation flags:
    static

Mode and number of proofs:
    is_present - zero_or_one

---

if_empty/1

Calls a goal if the optional term is empty. Succeeds otherwise.

Compilation flags:
    static

Template:
    if_empty(Goal)
Meta-predicate template:
    if_empty(0)
Mode and number of proofs:
    if_empty(+callable) - zero_or_more

---

if_present/1

Applies a closure to the value hold by the optional term if not empty. Succeeds otherwise.

Compilation flags:
    static

Template:
    if_present(Closure)
Meta-predicate template:
    if_present(1)
Mode and number of proofs:
    if_present(+callable) - zero_or_more

---

if_present_or_else/2

Applies a closure to the value hold by the optional term if not empty. Otherwise calls the given goal.

Compilation flags:
    static

Template:
    if_present_or_else(Closure,Goal)
Meta-predicate template:
    if_present_or_else(1,0)
Mode and number of proofs:
    if_present_or_else(+callable,+callable) - zero_or_more

---

filter/2

Returns the optional term when it is not empty and the value it holds satisfies a closure. Otherwise returns an empty optional term.

Compilation flags:
    static

Template:
    filter(Closure,NewOptional)
Meta-predicate template:
    filter(1,*)
Mode and number of proofs:
    filter(+callable,--nonvar) - one

---

map/2

When the optional term is not empty and mapping a closure with the value it holds and the new value as additional arguments is successful, returns an optional term with the new value. Otherwise returns an empty optional term.

Compilation flags:
    static

Template:
    map(Closure,NewOptional)
Meta-predicate template:
    map(2,*)
Mode and number of proofs:
    map(+callable,--nonvar) - one

---

flat_map/2

When the optional term is not empty and mapping a closure with the value it holds and the new optional term as additional arguments is successful, returns the new optional term. Otherwise returns an empty optional term.

Compilation flags:
    static

Template:
    flat_map(Closure,NewOptional)
Meta-predicate template:
    flat_map(2,*)
Mode and number of proofs:
    flat_map(+callable,--nonvar) - one

---

or/2

Returns the same optional term if not empty. Otherwise calls closure to generate a new optional term. Fails if optional term is empty and calling the closure fails or throws an error.

Compilation flags:
    static

Template:
    or(NewOptional,Closure)
Meta-predicate template:
    or(*,1)
Mode and number of proofs:
    or(--term,@callable) - zero_or_one

---

get/1

Returns the value hold by the optional term if not empty. Throws an error otherwise.

Compilation flags:
    static

Template:
    get(Value)
Mode and number of proofs:
    get(--term) - one_or_error

Exceptions:
    Optional is empty:
        existence_error(optional_term,Optional)

---

or_else/2

Returns the value hold by the optional term if not empty or the given default value if the optional term is empty.

Compilation flags:
    static

Template:
    or_else(Value,Default)
Mode and number of proofs:
    or_else(--term,@term) - one

---

or_else_get/2

Returns the value hold by the optional term if not empty. Applies a closure to compute the value otherwise. Throws an error when the optional term is empty and the value cannot be computed.

Compilation flags:
    static

Template:
    or_else_get(Value,Closure)
Meta-predicate template:
    or_else_get(*,1)
Mode and number of proofs:
    or_else_get(--term,+callable) - one_or_error

Exceptions:
    Optional is empty and the term cannot be computed:
        existence_error(optional_term,Optional)

---

or_else_call/2

Returns the value hold by the optional term if not empty or calls a goal deterministically if the optional term is empty.

Compilation flags:
    static

Template:
    or_else_call(Value,Goal)
Meta-predicate template:
    or_else_call(*,0)
Mode and number of proofs:
    or_else_call(--term,+callable) - zero_or_one

---

or_else_fail/1

Returns the value hold by the optional term if not empty. Fails otherwise. Usually called to skip over empty optional terms.

Compilation flags:
    static

Template:
    or_else_fail(Value)
Mode and number of proofs:
    or_else_fail(--term) - zero_or_one

---

or_else_throw/2

Returns the value hold by the optional term if not empty. Throws the given error otherwise.

Compilation flags:
    static

Template:
    or_else_throw(Value,Error)

Mode and number of proofs:
    or_else_throw(--term,@nonvar) - one_or_error

---

map_or_else/3

When the optional term is not empty and mapping a closure with the value it holds and the new value as additional arguments is successful, returns the new value. Otherwise returns the given default value.

Compilation flags:
    static

Template:
    map_or_else(Closure,Default,Value)
Meta-predicate template:
    map_or_else(2,*,*)
Mode and number of proofs:
    map_or_else(+callable,@term,--term) - one

---

zip/3

When both this optional and the other optional hold values and applying a closure with both values and the new value as additional arguments is successful, returns an optional term with the new value. Otherwise returns an empty optional term.

Compilation flags:
    static

Template:
    zip(Closure,OtherOptional,NewOptional)
Meta-predicate template:
    zip(3,*,*)
Mode and number of proofs:
    zip(+callable,+nonvar,--nonvar) - one

---

flatten/1

Flattens a nested optional term. When the optional term holds a value that is itself an optional term, returns the inner optional term. When the optional term holds a non-optional value, returns the same optional term. When the optional term is empty, returns an empty optional term.

Compilation flags:
    static

Template:
    flatten(NewOptional)
Mode and number of proofs:
    flatten(--nonvar) - one

---

to_expected/2

Converts the optional term to an expected term. Returns an expected term holding the value if the optional term is not empty. Returns an expected term with the given error otherwise.

Compilation flags:
    static

Template:
    to_expected(Error,Expected)
Mode and number of proofs:
    to_expected(@term,--nonvar) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

optional

1.168 options

category

1.168.1 options

Options processing predicates. Options are represented by compound terms where the functor is the option name.

Availability:
logtalk_load(options(loader))

Author: Paulo Moura
Version: 1:2:0
Date: 2022-01-03

Compilation flags:
static

Implements:
public options_protocol
Uses:
list

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 default_option/1 default_options/1 option/2 option/3 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.168.2 options_protocol

Options protocol.

Availability:
    logtalk_load(options(loader))

Author: Paulo Moura
Version: 1:2:0
Date: 2022-01-03

Compilation flags:
    static

Dependencies:
    (none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - check_option/1
  - check_options/1
  - valid_option/1
  - valid_options/1
  - default_option/1
  - default_options/1
  - option/2
  - option/3
- Protected predicates
  - merge_options/2
  - fix_options/2
  - fix_option/2
- Private predicates
- Operators

Public predicates

check_option/1

Succeeds if the option is valid. Throws an error otherwise.

Compilation flags:
static

Template:
check_option(Option)
Mode and number of proofs:
check_option(@term) - one_or_error

Exceptions:

Option is a variable:
instantiation_error
Option is neither a variable nor a compound term:
type_error(compound,Option)
Option is a compound term but not a valid option:
domain_error(option,Option)

---

check_options/1

Succeeds if all the options in a list are valid. Throws an error otherwise.

Compilation flags:
static

Template:
check_options(Options)
Mode and number of proofs:
check_options(@term) - one_or_error

Exceptions:
Options is a variable:
instantiation_error
Options is neither a variable nor a list:
type_error(list,Options)
An element Option of the list Options is a variable:
instantiation_error
An element Option of the list Options is neither a variable nor a compound term:
type_error(compound,Option)
An element Option of the list Options is a compound term but not a valid option:
domain_error(option,Option)

---

valid_option/1

Succeeds if the option is valid.

Compilation flags:
    static

Template:
    valid_option(Option)
Mode and number of proofs:
    valid_option(@term) - zero_or_one

---

valid_options/1

Succeeds if all the options in a list are valid.

Compilation flags:
    static

Template:
    valid_options(Options)
Mode and number of proofs:
    valid_options(@term) - one

---

default_option/1

Enumerates, by backtracking, the default options.

Compilation flags:
    static

Template:
    default_option(Option)
Mode and number of proofs:
    default_option(?compound) - zero_or_more

---

default_options/1

Returns a list of the default options.

Compilation flags:
    static

Template:
    default_options(Options)
Mode and number of proofs:
    default_options(-list(compound)) - one

---

option/2

True iff Option unifies with the first occurrence of the same option in the Options list.

Compilation flags:
    static

Template:
    option(Option,Options)
Mode and number of proofs:
    option(+compound,+list(compound)) - zero_or_one

---

option/3

True iff Option unifies with the first occurrence of the same option in the Options list or, when that is not the case, if Option unifies with Default.

Compilation flags:
    static

Template:
    option(Option,Options,Default)
Mode and number of proofs:
    option(+compound,+list(compound),+compound) - zero_or_one

Protected predicates

merge_options/2

Merges the user options with the default options, returning the final list of options. Calls the fix_options/2 predicate to preprocess the options after merging. Callers must ensure, if required, that the user options are valid.

Compilation flags:
    static

Template:
    merge_options(UserOptions,Options)
Mode and number of proofs:
    merge_options(+list(compound),-list(compound)) - one

---

fix_options/2

Fixes a list of options, returning the list of options.

Compilation flags:
    static

Template:
    fix_options(Options,FixedOptions)
Mode and number of proofs:
    fix_options(+list(compound),-list(compound)) - one

---

fix_option/2

Fixes an option.

Compilation flags:
    static

Template:
    fix_option(Option,FixedOption)
Mode and number of proofs:
    fix_option(+compound,-compound) - zero_or_one

Private predicates

(none)

Operators

(none)

See also

options

1.169 os

object

1.169.1 os

Portable operating-system access predicates.

Availability:
    logtalk_load(os(loader))

Author: Paulo Moura
Version: 1:103:1
Date: 2026-04-06

Compilation flags:
static, context_switching_calls

Implements:
public osp
Uses:
list
Aliases:
osp absolute_file_name/2 as expand_path/2

Remarks:

- File path expansion: To ensure portability, all file paths are expanded before being handed to the backend Prolog system.
- Exception terms: Currently, there is no standardization of the exception terms thrown by the different backend Prolog systems.
- B-Prolog portability: The wall_time/1 predicate is not supported.
- CxProlog portability: The date_time/7 predicate returns zeros for all arguments.
- JIProlog portability: The file_permission/2 and command_line_arguments/1 predicates are not supported.
- Quintus Prolog: The pid/1 and shell/2 predicates are not supported.
- XSB portability: The command_line_arguments/1 predicate is not supported.

Inherited public predicates:
absolute_file_name/2 change_directory/1 command_line_arguments/1 copy_file/2 cpu_time/1 date_time/7 decompose_file_name/3 decompose_file_name/4 delete_directory/1 delete_directory_and_contents/1 delete_directory_contents/1 delete_file/1 directory_exists/1 directory_files/2 directory_files/3 ensure_directory/1 ensure_file/1 environment_variable/2 file_exists/1 file_modification_time/2 file_permission/2 file_size/2 full_device_path/1 internal_os_path/2 is_absolute_file_name/1 make_directory/1 make_directory_path/1 null_device_path/1 operating_system_machine/1 operating_system_name/1 operating_system_release/1 operating_system_type/1 path_concat/3 pid/1 read_only_device_path/1 rename_file/2 shell/1 shell/2 sleep/1 temporary_directory/1 time_stamp/1 wall_time/1 working_directory/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

os_types

category

1.169.2 os_types

A set of operating-system related types.

Availability:
logtalk_load(os(loader))

Author: Paulo Moura
Version: 1:4:0
Date: 2021-02-12

Compilation flags:
static

Provides:
type::type/1
type::check/2
Uses:
list
os

Remarks:

- Provided types: This category adds file, file(Extensions), file(Extensions,Permissions), directory, directory(Permissions), and environment_variable types for type-checking when using the type library object.
- Type file: For checking if a term is an atom and an existing file.
- Type file(Extensions): For checking if a term is an atom and an existing file with one of the listed extensions (specified as `'.ext'`).
- Type file(Extensions,Permissions): For checking if a term is an atom and an existing file with one of the listed extensions (specified as `'.ext'`) and listed permissions ({read, write, execute}).
- Type directory: For checking if a term is an atom and an existing directory.
- Type directory(Permissions): For checking if a term is an atom and an existing directory with the listed permissions ({read, write, execute}).
- Type environment_variable: For checking if a term is an atom and an existing environment variable.

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

osp, os, type

protocol

1.169.3 osp

Portable operating-system access protocol.

Availability:
logtalk_load(os(loader))

Author: Paulo Moura
Version: 1:43:0
Date: 2026-04-03

Compilation flags:
static

Dependencies:
(none)

Remarks:

- Error handling: Predicates that require a file or directory to exist throw an error when that is not the case. But the exact exception term is currently backend Prolog compiler dependent.
- CPU and wall time accuracy: Depends on the backend and can be different between CPU and wall time (e.g. CPU time can have nanosecond accuracy with wall time only having millisecond accuracy).

Inherited public predicates:
(none)

- Public predicates
  - pid/1
  - shell/2

- shell/1
- is_absolute_file_name/1
- absolute_file_name/2
- decompose_file_name/3
- decompose_file_name/4
- path_concat/3
- internal_os_path/2
- make_directory/1
- make_directory_path/1
- delete_directory/1
- delete_directory_contents/1
- delete_directory_and_contents/1
- change_directory/1
- working_directory/1
- temporary_directory/1
- null_device_path/1
- full_device_path/1
- read_only_device_path/1
- directory_files/2
- directory_files/3
- directory_exists/1
- ensure_directory/1
- file_exists/1
- file_modification_time/2
- file_size/2
- file_permission/2
- copy_file/2
- rename_file/2
- delete_file/1
- ensure_file/1
- environment_variable/2
- time_stamp/1
- date_time/7
- cpu_time/1
- wall_time/1

    - operating_system_type/1
    - operating_system_name/1
    - operating_system_machine/1
    - operating_system_release/1
    - command_line_arguments/1
    - sleep/1
- Protected predicates
- Private predicates
- Operators

Public predicates

pid/1

Returns the process identifier of the running process.

Compilation flags:
    static

Template:
    pid(PID)
Mode and number of proofs:
    pid(-integer) - one

---

shell/2

Runs an operating-system shell command and returns its exit status.

Compilation flags:
    static

Template:
    shell(Command,Status)
Mode and number of proofs:
    shell(+atom,-integer) - one

---

shell/1

Runs an operating-system shell command.

Compilation flags:
    static

Template:
    shell(Command)
Mode and number of proofs:
    shell(+atom) - zero_or_one

---

is_absolute_file_name/1

True iff the argument is an absolute file path. On POSIX systems, this predicate is true if File starts with a /. On Windows systems, this predicate is true if File starts with a drive letter. No attempt is made to expand File as a path.

Compilation flags:
    static

Template:
    is_absolute_file_name(File)
Mode and number of proofs:
    is_absolute_file_name(+atom) - zero_or_one

---

absolute_file_name/2

Expands a file name to an absolute file path. An environment variable at the beginning of the file name is also expanded. A slash at the end of File is preserved in Path.

Compilation flags:
    static

Template:
    absolute_file_name(File,Path)
Mode and number of proofs:

absolute_file_name(+atom,-atom) - one

---

decompose_file_name/3

Decomposes a file name into its directory (which always ends with a slash; ./ is returned if absent) and its basename (which can be the empty atom).

Compilation flags:
static

Template:
decompose_file_name(File,Directory,Basename)
Mode and number of proofs:
decompose_file_name(+atom,?atom,?atom) - one

---

decompose_file_name/4

Decomposes a file name into its directory (which always ends with a slash; ./ is returned if absent), name (that can be the empty atom), and extension (which starts with a . when defined; the empty atom otherwise).

Compilation flags:
static

Template:
decompose_file_name(File,Directory,Name,Extension)
Mode and number of proofs:
decompose_file_name(+atom,?atom,?atom,?atom) - one

---

path_concat/3

Concatenates a path prefix and a path suffix, adding a / separator if required. Returns Suffix when it is an absolute path. Returns Prefix with a trailing / appended if missing when Suffix is the empty atom.

Compilation flags:
    static

Template:
    path_concat(Prefix,Suffix,Path)
Mode and number of proofs:
    path_concat(+atom,+atom,--atom) - one

---

internal_os_path/2

Converts between the internal path representation (which is backend dependent) and the operating-system native path representation.

Compilation flags:
    static

Template:
    internal_os_path(InternalPath,OSPath)
Mode and number of proofs:
    internal_os_path(+atom,-atom) - one
    internal_os_path(-atom,+atom) - one

---

make_directory/1

Makes a new directory. Succeeds if the directory already exists.

Compilation flags:
    static

Template:
    make_directory(Directory)
Mode and number of proofs:

make_directory(+atom) - one

---

make_directory_path/1

Makes a new directory creating all the intermediate directories if necessary. Succeeds if the directory already exists.

Compilation flags:
static

Template:
make_directory_path(Directory)
Mode and number of proofs:
make_directory_path(+atom) - one

---

delete_directory/1

Deletes an empty directory. Throws an error if the directory does not exist.

Compilation flags:
static

Template:
delete_directory(Directory)
Mode and number of proofs:
delete_directory(+atom) - one_or_error

---

delete_directory_contents/1

Deletes directory contents. Throws an error if the directory does not exist.

Compilation flags:
static

Template:
    delete_directory_contents(Directory)
Mode and number of proofs:
    delete_directory_contents(+atom) - one_or_error

---

delete_directory_and_contents/1

Deletes directory and its contents. Throws an error if the directory does not exist.

Compilation flags:
    static

Template:
    delete_directory_and_contents(Directory)
Mode and number of proofs:
    delete_directory_and_contents(+atom) - one_or_error

---

change_directory/1

Changes current working directory. Throws an error if the directory does not exist.

Compilation flags:
    static

Template:
    change_directory(Directory)
Mode and number of proofs:
    change_directory(+atom) - one_or_error

---

working_directory/1

Current working directory.

Compilation flags:
    static

Template:
    working_directory(Directory)
Mode and number of proofs:
    working_directory(?atom) - zero_or_one

---

temporary_directory/1

Temporary directory. Tries first environment variables: TEMP and TMP on Windows systems; TMPDIR on POSIX systems. When not defined, tries default locations. Returns the working directory as last resort.

Compilation flags:
    static

Template:
    temporary_directory(Directory)
Mode and number of proofs:
    temporary_directory(?atom) - one

---

null_device_path/1

Null device path: nul on Windows systems and /dev/null on POSIX systems.

Compilation flags:
    static

Template:
    null_device_path(Path)
Mode and number of proofs:
    null_device_path(?atom) - one

---

full_device_path/1

Full device path: /dev/full on Linux and BSD systems. Fails on other systems. Experimental.

Compilation flags:
    static

Template:
    full_device_path(Path)
Mode and number of proofs:
    full_device_path(?atom) - zero_or_one

---

read_only_device_path/1

Read-only device path: /dev/urandom on macOS. Fails on other systems. Experimental.

Compilation flags:
    static

Template:
    read_only_device_path(Path)
Mode and number of proofs:
    read_only_device_path(?atom) - zero_or_one

---

directory_files/2

Returns a list of all files (including directories, regular files, and hidden directories and files) in a directory. File paths are relative to the directory. Throws an error if the directory does not exist.

Compilation flags:
    static

Template:
    directory_files(Directory,Files)

Mode and number of proofs:
directory_files(+atom,-list(atom)) - one_or_error

---

directory_files/3

Returns a list of files filtered using the given list of options. Invalid options are ignored. Default option values are equivalent to directory_files/2. Throws an error if the directory does not exist.

Compilation flags:
static

Template:
directory_files(Directory,Files,Options)
Mode and number of proofs:
directory_files(+atom,-list(atom),+list(compound)) - one_or_error

Remarks:

- Option paths/1: Possible values are relative and absolute. Default is relative.
- Option type/1: Possible values are all, regular, directory. Default is all.
- Option extensions/1: Argument is a list of required extensions (using the format '.ext'). Default is the empty list.
- Option prefixes/1: Argument is a list of required file prefixes (atoms). Default is the empty list.
- Option suffixes/1: Argument is a list of required file suffixes (atoms). Default is the empty list.
- Option dot_files/1: Possible values are true and false. Default is true.

---

directory_exists/1

True if the specified directory exists (irrespective of directory permissions).

Compilation flags:
static

Template:
directory_exists(Directory)
Mode and number of proofs:
directory_exists(+atom) - zero_or_one

---

ensure_directory/1

Ensures that a directory exists, creating it if necessary.

Compilation flags:
    static

Template:
    ensure_directory(Directory)
Mode and number of proofs:
    ensure_directory(+atom) - one

---

file_exists/1

True if the specified file exists and is a regular file (irrespective of file permissions).

Compilation flags:
    static

Template:
    file_exists(File)
Mode and number of proofs:
    file_exists(+atom) - zero_or_one

---

file_modification_time/2

File modification time (which can be used for comparison). Throws an error if the file does not exist.

Compilation flags:
    static

Template:
    file_modification_time(File,Time)
Mode and number of proofs:

file_modification_time(+atom,-integer) - one_or_error

---

file_size/2

File size (in bytes). Throws an error if the file does not exist.

Compilation flags:
static

Template:
file_size(File,Size)
Mode and number of proofs:
file_size(+atom,-integer) - one_or_error

---

file_permission/2

True iff the specified file has the specified permission (read, write, or execute). Throws an error if the file does not exist.

Compilation flags:
static

Template:
file_permission(File,Permission)
Mode and number of proofs:
file_permission(+atom,+atom) - zero_or_one_or_error

---

copy_file/2

Copies a file. Throws an error if the original file does not exist or if the copy cannot be created.

Compilation flags:
static

Template:
    copy_file(File,Copy)
Mode and number of proofs:
    copy_file(+atom,+atom) - one_or_error

---

rename_file/2

Renames a file or a directory. Throws an error if the file or directory does not exist.

Compilation flags:
    static

Template:
    rename_file(Old,New)
Mode and number of proofs:
    rename_file(+atom,+atom) - one_or_error

---

delete_file/1

Deletes a file. Throws an error if the file does not exist.

Compilation flags:
    static

Template:
    delete_file(File)
Mode and number of proofs:
    delete_file(+atom) - one_or_error

---

ensure_file/1

Ensures that a file exists, creating it if necessary.

Compilation flags:
    static

Template:
    ensure_file(File)
Mode and number of proofs:
    ensure_file(+atom) - one

---

environment_variable/2

Returns an environment variable value. Fails if the variable does not exists.

Compilation flags:
    static

Template:
    environment_variable(Variable,Value)
Mode and number of proofs:
    environment_variable(+atom,?atom) - zero_or_one

---

time_stamp/1

Returns a system-dependent time stamp, which can be used for sorting, but should be regarded otherwise as an opaque term.

Compilation flags:
    static

Template:
    time_stamp(Time)
Mode and number of proofs:
    time_stamp(-ground) - one

---

date_time/7

Returns the current date and time. Note that most backends do not provide sub-second accuracy and in those cases the value of the Milliseconds argument is always zero.

Compilation flags:
    static

Template:
    date_time(Year,Month,Day,Hours,Minutes,Seconds,Milliseconds)
Mode and number of proofs:
    date_time(-integer,-integer,-integer,-integer,-integer,-integer,-integer) - one

---

cpu_time/1

System cpu time in seconds. Accuracy depends on the backend.

Compilation flags:
    static

Template:
    cpu_time(Seconds)
Mode and number of proofs:
    cpu_time(-number) - one

---

wall_time/1

Wall time in seconds. Accuracy depends on the backend.

Compilation flags:
    static

Template:
    wall_time(Seconds)

Mode and number of proofs:
wall_time(-number) - one

---

operating_system_type/1

Operating system type. Possible values are unix, windows, and unknown.

Compilation flags:
static

Template:
operating_system_type(Type)
Mode and number of proofs:
operating_system_type(?atom) - zero_or_one

---

operating_system_name/1

Operating system name. On POSIX systems, it returns the value of uname -s. On macOS systems, it returns 'Darwin'. On Windows systems, it returns 'Windows'.

Compilation flags:
static

Template:
operating_system_name(Name)
Mode and number of proofs:
operating_system_name(?atom) - zero_or_one

---

operating_system_machine/1

Operating system hardware platform. On POSIX systems, it returns the value of uname -m. On Windows systems, it returns the value of the PROCESSOR_ARCHITECTURE environment variable.

Compilation flags:
    static

Template:
    operating_system_machine(Machine)
Mode and number of proofs:
    operating_system_machine(?atom) - zero_or_one

---

operating_system_release/1

Operating system release. On POSIX systems, it returns the value of uname -r. On Windows systems, it uses WMI code.

Compilation flags:
    static

Template:
    operating_system_release(Release)
Mode and number of proofs:
    operating_system_release(?atom) - zero_or_one

---

command_line_arguments/1

Returns a list with the command line arguments that occur after --.

Compilation flags:
    static

Template:
    command_line_arguments(Arguments)
Mode and number of proofs:
    command_line_arguments(-list(atom)) - one

---

sleep/1

Suspends execution the given number of seconds.

Compilation flags:
    static

Template:
    sleep(Seconds)
Mode and number of proofs:
    sleep(+number) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

os, os_types

1.170 packs

protocol

1.170.1 pack_protocol

Pack specification protocol. Objects implementing this protocol should be named after the pack with a _pack suffix and saved in a file with the same name as the object.

Availability:
logtalk_load(packs(loader))

Author: Paulo Moura
Version: 0:18:0
Date: 2025-05-21

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - name/1
  - description/1
  - license/1
  - home/1
  - version/6
  - note/3
- Protected predicates
- Private predicates
- Operators

Public predicates

name/1

Pack name.

Compilation flags:
    static

Template:
    name(Name)
Mode and number of proofs:
    name(?atom) - zero_or_one

---

description/1

Pack one line description.

Compilation flags:
    static

Template:
    description(Description)
Mode and number of proofs:
    description(?atom) - zero_or_one

---

license/1

Pack license. Specified using the identifier from the SPDX License List (https://spdx.org/licenses/) when possible.

Compilation flags:
    static

Template:
    license(License)
Mode and number of proofs:

license(?atom) - zero_or_one

---

home/1

Pack home HTTPS or file URL.

Compilation flags:
static

Template:
home(Home)
Mode and number of proofs:
home(?atom) - zero_or_one

---

version/6

Table of available versions.

Compilation flags:
static

Template:
version(Version,Status,URL,Checksum,Dependencies,Portability)
Mode and number of proofs:
version(?compound,?atom,-atom,-pair(atom,atom),-list(pair(atom,callable)),?atom) - zero_or_more
version(?compound,?atom,-atom,-pair(atom,atom),-list(pair(atom,callable)),-list(atom)) - zero_or_more

Remarks:

- Version: This argument uses the same format as entity versions: Major:Minor:Patch. Semantic versioning should be used.
- Status: Version development status: stable, rc, beta, alpha, experimental, or deprecated.
- URL: File URL for a local directory, file URL for a local archive, download HTTPS URL for the pack archive, or download git archive URL for the pack archive.
- Checksum: A pair where the key is the hash algorithm and the value is the checksum. Currently, the hash algorithm must be sha256. For file:// URLs of local directories, use none instead of a pair.

- Dependencies: Pack dependencies list. Each dependency is a Dependency Operator Version term. Operator is a term comparison operator. Valid Dependency values are Registry::Pack, os(Name,Machine), logtalk, and a backend identifier atom.
- Portability: Either the atom all or a list of the supported backend Prolog compilers (using the identifier atoms used by the prolog_dialect flag).
- Clause order: Versions must be listed ordered from newest to oldest.

---

note/3

Table of notes per action and version.

Compilation flags:
    static

Template:
    note(Action,Version,Note)
Mode and number of proofs:
    note(?atom,?term,-atom) - zero_or_more

Remarks:

- Action: Possible values are install, update, and uninstall. When unbound, the note apply to all actions.
- Version: Version being installed, updated, or uninstalled. When unbound, the note apply to all versions.
- Note: Note to print when performing an action on a pack version.

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

object

1.170.2 packs

Pack handling predicates.

Availability:
logtalk_load(packs(loader))

Author: Paulo Moura
Version: 0:89:0
Date: 2026-03-24

Compilation flags:
static, context_switching_calls

Imports:
public packs_common
public options
Uses:
git
list
logtalk
os
registries
type
user

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 default_option/1 default_options/1 directory/1 directory/2 help/0 logtalk_packs/0 logtalk_packs/1 option/2 option/3 pin/0 pin/1 pinned/1 prefix/0 prefix/1 readme/1 readme/2 reset/0 setup/0 unpin/0 unpin/1 valid_option/1 valid_options/1 verify_commands_availability/0

- Public predicates
  - available/2
  - available/1
  - available/0
  - installed/4
  - installed/3
  - installed/1
  - installed/0
  - outdated/5
  - outdated/4
  - outdated/2
  - outdated/1
  - outdated/0
  - orphaned/2
  - orphaned/0
  - versions/3
  - describe/2
  - describe/1
  - pack_metadata/4
  - pack_property/4
  - pack_object/3
  - loaded_pack/3
  - loaded_pack_file/4
  - search/1
  - pack_dependency/6
  - loaded_pack_dependency/6
  - install/4
  - install/3
  - install/2
  - install/1
  - update/3
  - update/2
  - update/1
  - update/0
  - uninstall/2

  - uninstall/1
  - uninstall/0
  - clean/2
  - clean/1
  - clean/0
  - save/2
  - save/1
  - restore/2
  - restore/1
  - dependents/3
  - dependents/2
  - dependents/1
  - lint/2
  - lint/1
  - lint/0
- Protected predicates
- Private predicates
- Operators

Public predicates

available/2

Enumerates, by backtracking, all available packs.

Compilation flags:
    static

Template:
    available(Registry,Pack)
Mode and number of proofs:
    available(?atom,?atom) - zero_or_more

Exceptions:
    Registry is neither a variable nor an atom:
        type_error(atom,Registry)
    Pack is neither a variable nor an atom:
        type_error(atom,Pack)

---

available/1

Lists all the packs that are available for installation from the given registry.

Compilation flags:
    static

Template:
    available(Registry)
Mode and number of proofs:
    available(+atom) - one

Exceptions:
    Registry is a variable:
        instantiation_error
    Registry is neither a variable nor an atom:
        type_error(atom,Registry)

---

available/0

Lists all the packs that are available for installation from all defined registries.

Compilation flags:
    static

Mode and number of proofs:
    available - one

---

installed/4

Enumerates by backtracking all installed packs.

Compilation flags:
    static

Template:
    installed(Registry,Pack,Version,Pinned)
Mode and number of proofs:
    installed(?atom,?atom,?compound,?boolean) - zero_or_more

Exceptions:
    Registry is neither a variable nor an atom:
        type_error(atom,Registry)
    Pack is neither a variable nor an atom:
        type_error(atom,Pack)
    Version is neither a variable nor a compound term:
        type_error(compound,Version)
    Pinned is neither a variable nor a boolean:
        type_error(boolean,Pinned)

---

installed/3

Enumerates by backtracking all installed packs.

Compilation flags:
    static

Template:
    installed(Registry,Pack,Version)
Mode and number of proofs:
    installed(?atom,?atom,?compound) - zero_or_more

Exceptions:
    Registry is neither a variable nor an atom:
        type_error(atom,Registry)
    Pack is neither a variable nor an atom:
        type_error(atom,Pack)
    Version is neither a variable nor a compound term:

type_error(compound,Version)

---

installed/1

Lists all the packs that are installed from the given registry. Fails if the registry is unknown.

Compilation flags:
static

Template:
installed(Registry)
Mode and number of proofs:
installed(+atom) - zero_or_one

Exceptions:
Registry is a variable:
instantiation_error
Registry is neither a variable nor an atom:
type_error(atom,Registry)

---

installed/0

Lists all the packs that are installed.

Compilation flags:
static

Mode and number of proofs:
installed - one

---

outdated/5

Enumerates by backtracking all installed but outdated packs (together with the current version installed and the latest version available) using the given options.

Compilation flags:
    static

Template:
    outdated(Registry,Pack,Version,LatestVersion,Options)
Mode and number of proofs:
    outdated(?atom,?atom,?compound,?compound,++list(compound)) - zero_or_more

Exceptions:
    Registry is neither a variable nor an atom:
        type_error(atom,Registry)
    Pack is neither a variable nor an atom:
        type_error(atom,Pack)
    Version is neither a variable nor a compound term:
        type_error(compound,Version)
    LatestVersion is neither a variable nor a compound term:
        type_error(compound,LatestVersion)
    Options is a variable:
        instantiation_error
    Options is neither a variable nor a list:
        type_error(list,Options)
    An element Option of the list Options is a variable:
        instantiation_error
    An element Option of the list Options is neither a variable nor a compound term:
        type_error(compound,Option)
    An element Option of the list Options is a compound term but not a valid option:
        domain_error(option,Option)

Remarks:

- compatible(Boolean) option: Restrict listing to compatible packs. Default is true.
- status(Status) option: Restrict listing to updates with the given status. Default is [stable,rc,beta, alpha]. Set to all to also list experimental and deprecated updates.

outdated/4

Enumerates by backtracking all installed but outdated packs (together with the current version installed and the latest version available) using default options.

Compilation flags:
    static

Template:
    outdated(Registry,Pack,Version,LatestVersion)
Mode and number of proofs:
    outdated(?atom,?atom,?compound,?compound) - zero_or_more

Exceptions:
    Registry is neither a variable nor an atom:
        type_error(atom,Registry)
    Pack is neither a variable nor an atom:
        type_error(atom,Pack)
    Version is neither a variable nor a compound term:
        type_error(compound,Version)
    LatestVersion is neither a variable nor a compound term:
        type_error(compound,LatestVersion)

See also:
    outdated/5

---

outdated/2

Lists all the packs from the given registry that are installed but outdated using the given options.

Compilation flags:
    static

Template:
    outdated(Registry,Options)
Mode and number of proofs:
    outdated(+atom,++list(compound)) - one

Exceptions:
    Registry is neither a variable nor an atom:

type_error(atom,Registry)
Options is a variable:
instantiation_error
Options is neither a variable nor a list:
type_error(list,Options)
An element Option of the list Options is a variable:
instantiation_error
An element Option of the list Options is neither a variable nor a compound term:
type_error(compound,Option)
An element Option of the list Options is a compound term but not a valid option:
domain_error(option,Option)

Remarks:

- compatible(Boolean) option: Restrict installation to compatible packs. Default is true.
- status(Status) option: Restrict listing to updates with the given status. Default is [stable,rc,beta, alpha]. Set to all to also list experimental and deprecated updates.

---

outdated/1

Lists all the packs from the given registry that are installed but outdated using default options.

Compilation flags:
static

Template:
outdated(Registry)
Mode and number of proofs:
outdated(+atom) - one

Exceptions:
Registry is neither a variable nor an atom:
type_error(atom,Registry)

See also:
outdated/2

---

outdated/0

Lists all the packs that are installed but outdated using default options.

Compilation flags:
    static

Mode and number of proofs:
    outdated - one

See also:
    outdated/1

---

orphaned/2

Lists all the packs that are installed but whose registry is no longer defined.

Compilation flags:
    static

Template:
    orphaned(Registry,Pack)
Mode and number of proofs:
    orphaned(?atom,?atom) - zero_or_more

Exceptions:
    Registry is neither a variable nor an atom:
        type_error(atom,Registry)
    Pack is neither a variable nor an atom:
        type_error(atom,Pack)

---

orphaned/0

Lists all the packs that are installed but whose registry is no longer defined.

Compilation flags:
    static

Mode and number of proofs:
    orphaned - one

---

versions/3

Returns a list of all available pack versions. Fails if the pack is unknown.

Compilation flags:
    static

Template:
    versions(Registry,Pack,Versions)
Mode and number of proofs:
    versions(+atom,+atom,-list) - zero_or_one

Exceptions:
    Registry is a variable:
        instantiation_error
    Registry is neither a variable nor an atom:
        type_error(atom,Registry)
    Pack is a variable:
        instantiation_error
    Pack is neither a variable nor an atom:
        type_error(atom,Pack)

---

describe/2

Describes a registered pack, including installed version if applicable. Fails if the pack is unknown.

Compilation flags:
    static

Template:
    describe(Registry,Pack)
Mode and number of proofs:
    describe(+atom,+atom) - zero_or_one

Exceptions:
    Registry is a variable:
        instantiation_error
    Registry is neither a variable nor an atom:
        type_error(atom,Registry)
    Pack is a variable:
        instantiation_error
    Pack is neither a variable nor an atom:
        type_error(atom,Pack)

---

describe/1

Describes a registered pack, including installed version if applicable. Fails if the pack is unknown.

Compilation flags:
    static

Template:
    describe(Pack)
Mode and number of proofs:
    describe(+atom) - zero_or_one

Exceptions:
    Pack is a variable:
        instantiation_error
    Pack is neither a variable nor an atom:
        type_error(atom,Pack)

pack_metadata/4

Enumerates by backtracking resolved metadata for installed packs.

Compilation flags:
    static

Template:
    pack_metadata(Registry,Pack,Version,Metadata)
Mode and number of proofs:
    pack_metadata(?atom,?atom,?compound,?compound) - zero_or_more

Exceptions:
    Registry is neither a variable nor an atom:
        type_error(atom,Registry)
    Pack is neither a variable nor an atom:
        type_error(atom,Pack)
    Version is neither a variable nor a compound term:
        type_error(compound,Version)
    Metadata is neither a variable nor a compound term:
        type_error(compound,Metadata)

Remarks:

- Metadata: Resolved metadata term for the selected installed pack version.
- Resolved fields: The metadata term includes both version-independent pack metadata and version-specific metadata selected from the installed version.
- Absent optional metadata: Missing optional metadata is normalized to none instead of causing failure.
- Metadata term: Metadata is returned using the term metadata(Name, Description, License, Home, SourceURL, Checksum, Dependencies, Portability, Directory, Pinned, Installed, Loaded).

See also:
    pack_property/4
    pack_object/3
    loaded_pack/3

pack_property/4

Enumerates by backtracking resolved properties for installed pack versions.

Compilation flags:
static

Template:
pack_property(Registry,Pack,Version,Property)
Mode and number of proofs:
pack_property(?atom,?atom,?compound,?compound) - zero_or_more

Exceptions:
Registry is neither a variable nor an atom:
type_error(atom,Registry)
Pack is neither a variable nor an atom:
type_error(atom,Pack)
Version is neither a variable nor a compound term:
type_error(compound,Version)
Property is neither a variable nor a compound term:
type_error(compound,Property)

Remarks:

- Property: A resolved property term for the selected installed pack version.
- Supported properties: Supported properties are name(Name), description(Description), license(License), home(Home), source_url(URL), checksum(Checksum), dependencies(Dependencies), portability(Portability), directory(Directory), pinned(Boolean), installed(Boolean), and loaded(Boolean).
- Absent optional metadata: Missing optional metadata is normalized to none.

See also:
pack_metadata/4
pack_object/3

pack_object/3

Enumerates by backtracking pack specification objects for defined registry and pack pairs.

Compilation flags:
static

Template:
pack_object(Registry,Pack,PackObject)
Mode and number of proofs:
pack_object(?atom,?atom,?atom) - zero_or_more

Exceptions:
Registry is neither a variable nor an atom:
type_error(atom,Registry)
Pack is neither a variable nor an atom:
type_error(atom,Pack)
PackObject is neither a variable nor an atom:
type_error(atom,PackObject)

Remarks:

- PackObject: Identifier of the object implementing the pack_protocol for the given registry and pack.
- Intended use: This predicate is a low-level bridge for advanced tools and should not be required by most consumers when pack_metadata/4 or pack_property/4 is sufficient.

See also:
pack_metadata/4
pack_property/4

---

loaded_pack/3

Enumerates by backtracking all installed packs that contributed loaded files to the current session.

Compilation flags:
static

Template:
loaded_pack(Registry,Pack,Version)

Mode and number of proofs:
loaded_pack(?atom,?atom,?compound) - zero_or_more

Exceptions:
Registry is neither a variable nor an atom:
type_error(atom,Registry)
Pack is neither a variable nor an atom:
type_error(atom,Pack)
Version is neither a variable nor a compound term:
type_error(compound,Version)

Remarks:

- Loaded pack: A pack is considered loaded iff at least one currently loaded file belongs to its installation directory.
- Version: Installed version of the loaded pack.

See also:
loaded_pack_file/4
pack_metadata/4
installed/3

---

loaded_pack_file/4

Enumerates by backtracking loaded files that belong to installed packs in the current session.

Compilation flags:
static

Template:
loaded_pack_file(Registry,Pack,Version,File)
Mode and number of proofs:
loaded_pack_file(?atom,?atom,?compound,?atom) - zero_or_more

Exceptions:
Registry is neither a variable nor an atom:
type_error(atom,Registry)
Pack is neither a variable nor an atom:
type_error(atom,Pack)
Version is neither a variable nor a compound term:

type_error(compound,Version)
File is neither a variable nor an atom:
type_error(atom,File)

Remarks:

- File: Absolute path of a currently loaded file belonging to the selected installed pack.
- Intended use: This predicate supports diagnostics, tests, and the definition of loaded_pack/3.

See also:
loaded_pack/3

---

search/1

Searches packs whose name or description includes the search term (case sensitive).

Compilation flags:
static

Template:
search(Term)
Mode and number of proofs:
search(+atom) - one

Exceptions:
Term is a variable:
instantiation_error
Term is neither a variable nor an atom:
type_error(atom,Term)

---

pack_dependency/6

Enumerates by backtracking resolved direct pack-to-pack dependencies for installed pack versions.

Compilation flags:
    static

Template:
    pack_dependency(Registry,Pack,Version,DependencyRegistry,DependencyPack,DependencyVersion)
Mode and number of proofs:
    pack_dependency(?atom,?atom,?compound,?atom,?atom,?compound) - zero_or_more

Exceptions:
    Registry is neither a variable nor an atom:
        type_error(atom,Registry)
    Pack is neither a variable nor an atom:
        type_error(atom,Pack)
    Version is neither a variable nor a compound term:
        type_error(compound,Version)
    DependencyRegistry is neither a variable nor an atom:
        type_error(atom,DependencyRegistry)
    DependencyPack is neither a variable nor an atom:
        type_error(atom,DependencyPack)
    DependencyVersion is neither a variable nor a compound term:
        type_error(compound,DependencyVersion)

Remarks:

- Resolved dependency: Dependencies are resolved against the current installed packs using the same version comparison semantics used by the tool installation logic.
- Supported dependency forms: Version ranges, alternatives, and conjunctions are resolved using the canonical packs dependency semantics.
- Non-pack dependencies: Dependencies on Logtalk, Prolog backends, and operating systems are ignored by this predicate and therefore do not produce results.

See also:
    loaded_pack_dependency/6
    pack_metadata/4
    loaded_pack/3

loaded_pack_dependency/6

Enumerates by backtracking resolved direct pack-to-pack dependencies where both the source pack and the dependency pack contributed loaded files to the current session.

Compilation flags:
static

Template:
loaded_pack_dependency(Registry,Pack,Version,DependencyRegistry,DependencyPack,DependencyVersion)
Mode and number of proofs:
loaded_pack_dependency(?atom,?atom,?compound,?atom,?atom,?compound) - zero_or_more

Exceptions:
Registry is neither a variable nor an atom:
type_error(atom,Registry)
Pack is neither a variable nor an atom:
type_error(atom,Pack)
Version is neither a variable nor a compound term:
type_error(compound,Version)
DependencyRegistry is neither a variable nor an atom:
type_error(atom,DependencyRegistry)
DependencyPack is neither a variable nor an atom:
type_error(atom,DependencyPack)
DependencyVersion is neither a variable nor a compound term:
type_error(compound,DependencyVersion)

Remarks:

- Loaded dependency edge: This predicate is the loaded-pack counterpart of pack_dependency/6 and only succeeds when both endpoints are currently loaded.
- Intended use: This predicate is especially useful for tools that need to describe relationships between the installed packs that contributed code to the current session.

See also:
pack_dependency/6
loaded_pack/3

install/4

Installs a new pack using the specified options. Fails if the pack is unknown or already installed but not using update(true) or force(true) options. Fails also if the pack version is unknown.

Compilation flags:
static

Template:
install(Registry,Pack,Version,Options)
Mode and number of proofs:
install(+atom,+atom,++compound,++list(compound)) - zero_or_one

Exceptions:
Registry is a variable:
instantiation_error
Registry is neither a variable nor an atom:
type_error(atom,Registry)
Pack is a variable:
instantiation_error
Pack is neither a variable nor an atom:
type_error(atom,Pack)
Version is a variable:
instantiation_error
Version is neither a variable nor a valid version:
type_error(pack_version,Version)
Options is a variable:
instantiation_error
Options is neither a variable nor a list:
type_error(list,Options)
An element Option of the list Options is a variable:
instantiation_error
An element Option of the list Options is neither a variable nor a compound term:
type_error(compound,Option)
An element Option of the list Options is a compound term but not a valid option:
domain_error(option,Option)

Remarks:

- update(Boolean) option: Update pack if already installed. Default is false. Overrides the force/1 option.
- force(Boolean) option: Force pack re-installation if already installed. Default is false.
- compatible(Boolean) option: Restrict installation to compatible packs. Default is true.
- clean(Boolean) option: Clean pack archive after installation. Default is false.

- verbose(Boolean) option: Verbose installing steps. Default is false.
- checksum(Boolean) option: Verify pack archive checksum. Default is true.
- checksig(Boolean) option: Verify pack archive signature. Default is false.
- git(Atom) option: Extra command-line options. Default is ''.
- downloader(Atom) option: Downloader utility. Either curl or wget. Default is curl.
- curl(Atom) option: Extra command-line options. Default is ''.
- wget(Atom) option: Extra command-line options. Default is ''.
- gpg(Atom) option: Extra command-line options. Default is ''.
- tar(Atom) option: Extra command-line options. Default is ''.

---

install/3

Installs the specified version of a pack from the given registry using default options. Fails if the pack is already installed or unknown. Fails also if the pack version is unknown.

Compilation flags:
    static

Template:
    install(Registry,Pack,Version)
Mode and number of proofs:
    install(+atom,+atom,?compound) - zero_or_one

Exceptions:
    Registry is a variable:
        instantiation_error
    Registry is neither a variable nor an atom:
        type_error(atom,Registry)
    Pack is a variable:
        instantiation_error
    Pack is neither a variable nor an atom:
        type_error(atom,Pack)
    Version is a variable:
        instantiation_error
    Version is neither a variable nor a valid version:
        type_error(pack_version,Version)

See also:
    install/4

install/2

Installs the latest version of a pack from the given registry using default options. Fails if the pack is already installed or unknown.

Compilation flags:
static

Template:
install(Registry,Pack)
Mode and number of proofs:
install(+atom,+atom) - zero_or_one

Exceptions:
Registry is a variable:
instantiation_error
Registry is neither a variable nor an atom:
type_error(atom,Registry)
Pack is a variable:
instantiation_error
Pack is neither a variable nor an atom:
type_error(atom,Pack)

See also:
install/3

---

install/1

Installs a pack (if its name is unique among all registries) using default options. Fails if the pack is already installed or unknown. Fails also if the pack is available from multiple registries.

Compilation flags:
static

Template:
install(Pack)
Mode and number of proofs:

install(+atom) - zero_or_one

Exceptions:
Pack is a variable:
instantiation_error
Pack is not an atom:
type_error(atom,Pack)

See also:
install/2

---

update/3

Updates an outdated pack to the specified version using the specified options. Fails if the pack or the pack version is unknown or if the pack is not installed. Fails also if the pack is orphaned or pinned and not using a force(true) option.

Compilation flags:
static

Template:
update(Pack,Version,Options)
Mode and number of proofs:
update(+atom,++callable,++list(callable)) - zero_or_one

Exceptions:
Pack is a variable:
instantiation_error
Pack is neither a variable nor an atom:
type_error(atom,Pack)
Version is a variable:
instantiation_error
Version is neither a variable nor a valid version:
type_error(pack_version,Version)
Options is a variable:
instantiation_error
Options is neither a variable nor a list:
type_error(list,Options)
An element Option of the list Options is a variable:
instantiation_error
An element Option of the list Options is neither a variable nor a compound term:

type_error(compound,Option)
An element Option of the list Options is a compound term but not a valid option:
domain_error(option,Option)

Remarks:

- install(Boolean) option: Install pack latest version if not already installed. Default is false.
- force(Boolean) option: Force update if the pack is pinned or breaks installed packs. Default is false.
- compatible(Boolean) option: Restrict updating to compatible packs. Default is true.
- status(Status) option: Specify allowed pack status. Default is [stable,rc,beta,alpha]. Set to all to also allow experimental and deprecated.
- clean(Boolean) option: Clean pack archive after updating. Default is false.
- verbose(Boolean) option: Verbose updating steps. Default is false.
- checksum(Boolean) option: Verify pack archive checksum. Default is true.
- checksig(Boolean) option: Verify pack archive signature. Default is false.
- git(Atom) option: Extra command-line options. Default is ''.
- downloader(Atom) option: Downloader utility. Either curl or wget. Default is curl.
- curl(Atom) option: Extra command-line options. Default is ''.
- wget(Atom) option: Extra command-line options. Default is ''.
- gpg(Atom) option: Extra command-line options. Default is ''.
- tar(Atom) option: Extra command-line options. Default is ''.

---

update/2

Updates an outdated pack to its latest version using the specified options. Fails if the pack is orphaned, unknown, or not installed. Fails also if the pack is pinned and not using a force(true) option.

Compilation flags:
static

Template:
update(Pack,Options)
Mode and number of proofs:
update(+atom,++list(callable)) - zero_or_one

Exceptions:
Pack is a variable:
instantiation_error
Pack is neither a variable nor an atom:

type_error(atom,Pack)
Options is a variable:
instantiation_error
Options is neither a variable nor a list:
type_error(list,Options)
An element Option of the list Options is a variable:
instantiation_error
An element Option of the list Options is neither a variable nor a compound term:
type_error(compound,Option)
An element Option of the list Options is a compound term but not a valid option:
domain_error(option,Option)

Remarks:

- install(Boolean) option: Install pack latest version if not already installed. Default is false.
- force(Boolean) option: Force update if the pack is pinned or breaks installed packs. Default is false.
- compatible(Boolean) option: Restrict updating to compatible packs. Default is true.
- status(Status) option: Specify allowed pack update status. Default is [stable,rc,beta,alpha]. Set to all to also allow experimental and deprecated.
- clean(Boolean) option: Clean pack archive after updating. Default is false.
- verbose(Boolean) option: Verbose updating steps. Default is false.
- checksum(Boolean) option: Verify pack archive checksum. Default is true.
- checksig(Boolean) option: Verify pack archive signature. Default is false.
- git(Atom) option: Extra command-line options. Default is ''.
- downloader(Atom) option: Downloader utility. Either curl or wget. Default is curl.
- curl(Atom) option: Extra command-line options. Default is ''.
- wget(Atom) option: Extra command-line options. Default is ''.
- gpg(Atom) option: Extra command-line options. Default is ''.
- tar(Atom) option: Extra command-line options. Default is ''.

---

update/1

Updates an outdated pack to its latest version using default options. Fails if the pack is pinned, orphaned, not installed, unknown, or breaks installed packs.

Compilation flags:
static

Template:

update(Pack)
Mode and number of proofs:
update(+atom) - zero_or_one

Exceptions:
Pack is a variable:
instantiation_error
Pack is neither a variable nor an atom:
type_error(atom,Pack)

See also:
update/2
update/3

---

update/0

Updates all outdated packs (that are not pinned) using default options.

Compilation flags:
static

Mode and number of proofs:
update - zero_or_one

---

uninstall/2

Uninstalls a pack using the specified options. Fails if the pack is unknown or not installed. Fails also if the pack is pinned or have dependents and not using a force(true) option.

Compilation flags:
static

Template:
uninstall(Pack,Options)
Mode and number of proofs:
uninstall(+atom,++list(compound)) - zero_or_one

Exceptions:
Pack is a variable:
instantiation_error
Pack is neither a variable nor an atom:
type_error(atom,Pack)
Options is a variable:
instantiation_error
Options is neither a variable nor a list:
type_error(list,Options)
An element Option of the list Options is a variable:
instantiation_error
An element Option of the list Options is neither a variable nor a compound term:
type_error(compound,Option)
An element Option of the list Options is a compound term but not a valid option:
domain_error(option,Option)

Remarks:

- force(Boolean) option: Force deletion if the pack is pinned. Default is false.
- clean(Boolean) option: Clean pack archive after deleting. Default is false.
- verbose(Boolean) option: Verbose uninstalling steps. Default is false.

---

uninstall/1

Uninstalls a pack using default options. Fails if the pack is pinned, have dependents, not installed, or unknown.

Compilation flags:
static

Template:
uninstall(Pack)
Mode and number of proofs:
uninstall(+atom) - zero_or_one

Exceptions:
Pack is a variable:
instantiation_error
Pack is neither a variable nor an atom:
type_error(atom,Pack)

See also:

uninstall/2

---

uninstall/0

Uninstalls all packs using the force(true) option.

Compilation flags:
static

Mode and number of proofs:
uninstall - zero_or_one

---

clean/2

Cleans all pack archives. Fails if the the pack is unknown.

Compilation flags:
static

Template:
clean(Registry,Pack)
Mode and number of proofs:
clean(+atom,+atom) - zero_or_one

Exceptions:
Registry is a variable:
instantiation_error
Registry is neither a variable nor an atom:
type_error(atom,Registry)
Pack is a variable:
instantiation_error
Pack is neither a variable nor an atom:
type_error(atom,Pack)

---

clean/1

Cleans all pack archives. Fails if the pack is unknown.

Compilation flags:
    static

Template:
    clean(Pack)
Mode and number of proofs:
    clean(+atom) - zero_or_one

Exceptions:
    Pack is a variable:
        instantiation_error
    Pack is neither a variable nor an atom:
        type_error(atom,Pack)

See also:
    clean/2

---

clean/0

Cleans all archives for all packs.

Compilation flags:
    static

Mode and number of proofs:
    clean - one

---

save/2

Saves a list of all installed packs and registries plus pinning status to a file using the given options.

Compilation flags:
    static

Template:
    save(File,Options)
Mode and number of proofs:
    save(+atom,++list(compound)) - one_or_error

Exceptions:
    File is a variable:
        instantiation_error
    File is neither a variable nor an atom:
        type_error(atom,File)
    File is an existing file but cannot be written:
        permission_error(open,source_sink,File)
    Options is a variable:
        instantiation_error
    Options is neither a variable nor a list:
        type_error(list,Options)
    An element Option of the list Options is a variable:
        instantiation_error
    An element Option of the list Options is neither a variable nor a compound term:
        type_error(compound,Option)
    An element Option of the list Options is a compound term but not a valid option:
        domain_error(option,Option)

Remarks:

- lock(Boolean) option: Save lock extension facts (lockfile version, git registry commits, and pack integrity hashes) in addition to the standard requirements facts. Default is false.
- save(What) option: Save registries without installed packs when What is all and skip them when What is installed. Default is installed.

save/1

Saves a list of all installed packs and their registries plus pinning status to a file using default options.

Compilation flags:
    static

Template:
    save(File)
Mode and number of proofs:
    save(+atom) - one_or_error

Exceptions:
    File is a variable:
        instantiation_error
    File is neither a variable nor an atom:
        type_error(atom,File)
    File is an existing file but cannot be written:
        permission_error(open,source_sink,File)

See also:
    save/2

---

restore/2

Restores a list of registries and packs plus their pinning status from a file using the given options. Fails if restoring is not possible.

Compilation flags:
    static

Template:
    restore(File,Options)
Mode and number of proofs:
    restore(+atom,++list(compound)) - zero_or_one_or_error

Exceptions:
    File is a variable:
        instantiation_error
    File is neither a variable nor an atom:

type_error(atom,File)
File is an atom but not an existing file:
existence_error(file,File)
File is an existing file but cannot be read:
permission_error(open,source_sink,File)
Options is a variable:
instantiation_error
Options is neither a variable nor a list:
type_error(list,Options)
An element Option of the list Options is a variable:
instantiation_error
An element Option of the list Options is neither a variable nor a compound term:
type_error(compound,Option)
An element Option of the list Options is a compound term but not a valid option:
domain_error(option,Option)

Remarks:

- lock(Boolean) option: Require lock extension facts and enforce strict restoring (exact versions, git registry commits, and pack integrity hashes). Default is false.
- force(Boolean) option: Force restoring if a registry is already defined or a pack is already installed. Default is true.
- compatible(Boolean) option: Restrict installation to compatible packs. Default is true.
- clean(Boolean) option: Clean registry and pack archives after restoring. Default is false.
- verbose(Boolean) option: Verbose restoring steps. Default is false.
- checksum(Boolean) option: Verify pack archive checksums. Default is true.
- checksig(Boolean) option: Verify pack archive signatures. Default is false.
- git(Atom) option: Extra command-line options. Default is ''.
- downloader(Atom) option: Downloader utility. Either curl or wget. Default is curl.
- curl(Atom) option: Extra command-line options. Default is ''.
- wget(Atom) option: Extra command-line options. Default is ''.
- gpg(Atom) option: Extra command-line options. Default is ''.
- tar(Atom) option: Extra command-line options. Default is ''.

restore/1

Restores a list of registries and packs plus their pinning status from a file using default options. Fails if restoring is not possible.

Compilation flags:
    static

Template:
    restore(File)
Mode and number of proofs:
    restore(+atom) - zero_or_one_or_error

Exceptions:
    File is a variable:
        instantiation_error
    File is neither a variable nor an atom:
        type_error(atom,File)
    File is an atom but not an existing file:
        existence_error(file,File)
    File is an existing file but cannot be read:
        permission_error(open,source_sink,File)

See also:
    restore/2

---

dependents/3

Returns a list of all installed packs that depend on the given pack from the given registry. Fails if the pack is unknown.

Compilation flags:
    static

Template:
    dependents(Registry,Pack,Dependents)
Mode and number of proofs:
    dependents(+atom,+atom,-list(atom)) - zero_or_one

Exceptions:

Registry is a variable:
instantiation_error
Registry is neither a variable nor an atom:
type_error(atom,Registry)
Pack is a variable:
instantiation_error
Pack is neither a variable nor an atom:
type_error(atom,Pack)

---

dependents/2

Prints a list of all installed packs that depend on the given pack from the given registry. Fails if the pack is unknown.

Compilation flags:
static

Template:
dependents(Registry,Pack)
Mode and number of proofs:
dependents(+atom,+atom) - zero_or_one

Exceptions:
Registry is a variable:
instantiation_error
Registry is neither a variable nor an atom:
type_error(atom,Registry)
Pack is a variable:
instantiation_error
Pack is neither a variable nor an atom:
type_error(atom,Pack)

---

dependents/1

Prints a list of all installed packs that depend on the given pack if unique from all defined registries. Fails if the pack is unknown or available from multiple registries.

Compilation flags:
    static

Template:
    dependents(Pack)
Mode and number of proofs:
    dependents(+atom) - zero_or_one

Exceptions:
    Pack is a variable:
        instantiation_error
    Pack is neither a variable nor an atom:
        type_error(atom,Pack)

---

lint/2

Checks the pack specification. Fails if the pack is unknown or if linting detects errors.

Compilation flags:
    static

Template:
    lint(Registry,Pack)
Mode and number of proofs:
    lint(+atom,+atom) - zero_or_one

Exceptions:
    Registry is a variable:
        instantiation_error
    Registry is neither a variable nor an atom:
        type_error(atom,Registry)
    Pack is a variable:
        instantiation_error
    Pack is neither a variable nor an atom:
        type_error(atom,Pack)

---

lint/1

Checks the pack specification. Fails if the pack is unknown, or available from multiple registries, or if linting detects errors.

Compilation flags:
    static

Template:
    lint(Pack)
Mode and number of proofs:
    lint(+atom) - zero_or_one

Exceptions:
    Pack is a variable:
        instantiation_error
    Pack is neither a variable nor an atom:
        type_error(atom,Pack)

---

lint/0

Checks all pack specifications.

Compilation flags:
    static

Mode and number of proofs:
    lint - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.170.3 packs_common

Common predicates for the packs tool objects.

Availability:
logtalk_load(packs(loader))

Author: Paulo Moura
Version: 0:33:0
Date: 2025-01-23

Compilation flags:
static

Provides:
type::type/1
type::check/2
Uses:
list
logtalk
os
type
user

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - setup/0
  - reset/0
  - verify_commands_availability/0
  - help/0
  - pin/1
  - pin/0
  - unpin/1
  - unpin/0
  - pinned/1
  - directory/2
  - directory/1
  - readme/2
  - readme/1
  - logtalk_packs/1
  - logtalk_packs/0
  - prefix/1
  - prefix/0
- Protected predicates
  - readme_file_path/2
  - print_readme_file_path/1
  - command/2
  - load_registry/1
  - sha256sum_command/1
  - tar_command/1
  - supported_archive/1
  - supported_url_archive/1
  - decode_url_spaces/2
- Private predicates
- Operators

Public predicates

setup/0

Ensures that registries and packs directory structure exists. Preserves any defined registries and installed packs.

Compilation flags:
    static

Mode and number of proofs:
    setup - one

---

reset/0

Resets registries and packs directory structure. Deletes any defined registries and installed packs.

Compilation flags:
    static

Mode and number of proofs:
    reset - one

---

verify_commands_availability/0

Verifies required shell commands availability. Fails printing an error message if a command is missing.

Compilation flags:
    static

Mode and number of proofs:
    verify_commands_availability - zero_or_one

---

help/0

Provides help about the main predicates.

Compilation flags:
    static

Mode and number of proofs:
    help - one

---

pin/1

Pins a resource (pack or registry) preventing it from being updated, uninstalled, or deleted. Fails if the resource is not found.

Compilation flags:
    static

Template:
    pin(Resource)
Mode and number of proofs:
    pin(+atom) - zero_or_one

Exceptions:
    Resource is a variable:
        instantiation_error
    Resource is neither a variable nor an atom:
        type_error(atom,Resource)

---

pin/0

Pins all resource (packs or registries) preventing them from being updated, uninstalled, or deleted. Note that resources added after calling this predicate will not be pinned.

Compilation flags:
    static

Mode and number of proofs:
pin - one

---

unpin/1

Unpins a resource (pack or registry), allowing it to be updated, uninstalled, or deleted. Fails if the resource is not found.

Compilation flags:
static

Template:
unpin(Resource)
Mode and number of proofs:
unpin(+atom) - zero_or_one

Exceptions:
Resource is a variable:
instantiation_error
Resource is neither a variable nor an atom:
type_error(atom,Resource)

---

unpin/0

Unpins all resources (packs or registries), allowing them to be updated, uninstalled, or deleted.

Compilation flags:
static

Mode and number of proofs:
unpin - one

---

pinned/1

True iff the resource (pack or registry) is defined or installed and if it is pinned.

Compilation flags:
    static

Template:
    pinned(Resource)
Mode and number of proofs:
    pinned(+atom) - zero_or_one

Exceptions:
    Resource is a variable:
        instantiation_error
    Resource is neither a variable nor an atom:
        type_error(atom,Resource)

---

directory/2

Enumerates by backtracking all packs or registries and respective installation or definition directories (using the internal backend format).

Compilation flags:
    static

Template:
    directory(Resource,Directory)
Mode and number of proofs:
    directory(?atom,?atom) - zero_or_more

Exceptions:
    Resource is neither a variable nor an atom:
        type_error(atom,Resource)
    Directory is neither a variable nor an atom:
        type_error(atom,Directory)

---

directory/1

Prints the directory where the registry or the pack is installed (using the native operating-system format).

Compilation flags:
    static

Template:
    directory(Resource)
Mode and number of proofs:
    directory(+atom) - zero_or_one

Exceptions:
    Resource is a variable:
        instantiation_error
    Resource is neither a variable nor an atom:
        type_error(atom,Resource)

---

readme/2

Returns the path to the resource (pack or registry) readme file (using the internal backend format). Fails if the resource is not defined or installed or if no readme file is found for it.

Compilation flags:
    static

Template:
    readme(Resource,ReadMeFile)
Mode and number of proofs:
    readme(+atom,-atom) - zero_or_one

Exceptions:
    Resource is a variable:
        instantiation_error
    Resource is neither a variable nor an atom:
        type_error(atom,Resource)
    ReadMeFile is neither a variable nor an atom:
        type_error(atom,ReadMeFile)

---

readme/1

Prints the path to the resource (pack or registry) readme file (using the native operating-system format). Fails if the resource is not defined or installed or if no readme file is found for it.

Compilation flags:
    static

Template:
    readme(Resource)
Mode and number of proofs:
    readme(+atom) - zero_or_one

Exceptions:
    Resource is a variable:
        instantiation_error
    Resource is neither a variable nor an atom:
        type_error(atom,Resource)

---

logtalk_packs/1

Returns the directory prefix (using the internal backend format) where the registries, packs, and archives are installed.

Compilation flags:
    static

Template:
    logtalk_packs(LogtalkPacks)
Mode and number of proofs:
    logtalk_packs(-atom) - one

Exceptions:
    LogtalkPacks is neither a variable nor an atom:
        type_error(atom,LogtalkPacks)

---

logtalk_packs/0

Prints the directory prefix (using the native operating-system format) where the registries, packs, and archives are installed.

Compilation flags:
    static

Mode and number of proofs:
    logtalk_packs - one

---

prefix/1

Returns the directory prefix (using the internal backend format) where the registries or packs are installed.

Compilation flags:
    static

Template:
    prefix(Prefix)
Mode and number of proofs:
    prefix(-atom) - one

Exceptions:
    Prefix is neither a variable nor an atom:
        type_error(atom,Prefix)

---

prefix/0

Prints the directory prefix (using the native operating-system format) where the registries or packs are installed.

Compilation flags:
    static

Mode and number of proofs:

prefix - one

Protected predicates

readme_file_path/2

Returns the absolute path for the given directory readme file if it exists.

Compilation flags:
static

Template:
readme_file_path(Directory,ReadMeFile)
Mode and number of proofs:
readme_file_path(+atom,-atom) - zero_or_one

Remarks:

- Valid file names: Case variations of README and NOTES with or without a .md or .txt extension. The recommended file name is README.md.

---

print_readme_file_path/1

Prints the absolute path for the given directory readme file if it exists. Succeeds otherwise.

Compilation flags:
static

Template:
print_readme_file_path(Directory)
Mode and number of proofs:
print_readme_file_path(+atom) - one

---

command/2

Executes a shell command. Prints an error message and fails if the command fails.

Compilation flags:
    static

Template:
    command(Command,FailureMessage)
Mode and number of proofs:
    command(+atom,@nonvar) - zero_or_one

---

load_registry/1

Loads all registry files from the given directory.

Compilation flags:
    static

Template:
    load_registry(Directory)
Mode and number of proofs:
    load_registry(+atom) - zero_or_one

---

sha256sum_command/1

Returns the name of the sha256sum command to be used on POSIX systems. Fails if neither gsha256sum or sha256sum commands are found.

Compilation flags:
    static

Template:
    sha256sum_command(Command)
Mode and number of proofs:
    sha256sum_command(-atom) - zero_or_one

---

tar_command/1

Returns the name of the tar command to be used depending on the operating-system.

Compilation flags:
    static

Template:
    tar_command(Command)
Mode and number of proofs:
    tar_command(-atom) - one

---

supported_archive/1

True iff the archive format is supported.

Compilation flags:
    static

Template:
    supported_archive(Extension)
Mode and number of proofs:
    supported_archive(+atom) - zero_or_one

---

supported_url_archive/1

True iff the URL archive is supported.

Compilation flags:
    static

Template:
    supported_url_archive(URL)
Mode and number of proofs:

supported_url_archive(+atom) - zero_or_one

---

decode_url_spaces/2

Decodes encoded spaces (%20) in URLs to spaces.

Compilation flags:
static

Template:
decode_url_spaces(URL,Decoded)
Mode and number of proofs:
decode_url_spaces(+atom,-atom) - one

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.170.4 packs_messages

Packs default message translations.

Availability:
logtalk_load(packs(loader))

Author: Paulo Moura
Version: 0:42:0
Date: 2026-03-02

Compilation flags:
static

Provides:
logtalk::message_prefix_stream/4
logtalk::message_tokens//2

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.170.5 packs_specs_hook

Hook object for filtering registry and pack specification file contents.

Availability:
logtalk_load(packs(loader))

Author: Paulo Moura

Version: 0:13:0
Date: 2022-06-28

Compilation flags:
static, context_switching_calls

Implements:
public expanding
Uses:
character
logtalk

Remarks:
(none)

Inherited public predicates:
goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.170.6 registries

Registry handling predicates.

Availability:
logtalk_load(packs(loader))

Author: Paulo Moura
Version: 0:64:1
Date: 2026-03-21

Compilation flags:
static, context_switching_calls

Imports:
public packs_common
public options
Uses:
list
logtalk
os
type
user

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 default_option/1 default_options/1 directory/1 directory/2 help/0 logtalk_packs/0 logtalk_packs/1 option/2 option/3 pin/0 pin/1 pinned/1 prefix/0 prefix/1 readme/1 readme/2 reset/0 setup/0 unpin/0 unpin/1 valid_option/1 valid_options/1 verify_commands_availability/0

- Public predicates
  - list/0

  - describe/1
  - defined/4
  - add/3
  - add/2
  - add/1
  - update/2
  - update/1
  - update/0
  - delete/2
  - delete/1
  - delete/0
  - clean/1
  - clean/0
  - provides/2
  - lint/1
  - lint/0
- Protected predicates
- Private predicates
- Operators

Public predicates

list/0

Prints a list of all defined registries, including how defined (git, archive, or directory) and if they are pinned.

Compilation flags:
    static

Mode and number of proofs:
    list - one

---

describe/1

Prints all registry entries.

Compilation flags:
    static

Template:
    describe(Registry)
Mode and number of proofs:
    describe(+atom) - one

Exceptions:
    Registry is a variable:
        instantiation_error
    Registry is neither a variable nor an atom:
        type_error(atom,Registry)

---

defined/4

Enumerates by backtracking all defined registries, their definition URL, how they are defined (git, archive, or directory), and if they are pinned.

Compilation flags:
    static

Template:
    defined(Registry,URL,HowDefined,Pinned)
Mode and number of proofs:
    defined(?atom,?atom,?atom,?boolean) - zero_or_more

Exceptions:
    Registry is neither a variable nor an atom:
        type_error(atom,Registry)
    URL is neither a variable nor an atom:
        type_error(atom,URL)
    HowDefined is neither a variable nor an atom:
        type_error(atom,HowDefined)
    Pinned is neither a variable nor a boolean:
        type_error(boolean,Pinned)

add/3

Adds a new registry using the given options. Fails if the registry cannot be added or if it is already defined but not using update(true) or force(true) options. A file:// URL can be used for a local directory or archive.

Compilation flags:
    static

Template:
    add(Registry,URL,Options)
Mode and number of proofs:
    add(+atom,+atom,++list(compound)) - zero_or_one

Exceptions:
    Registry is a variable:
        instantiation_error
    Registry is neither a variable nor an atom:
        type_error(atom,Registry)
    URL is a variable:
        instantiation_error
    URL is neither a variable nor an atom:
        type_error(atom,URL)
    Options is a variable:
        instantiation_error
    Options is neither a variable nor a list:
        type_error(list,Options)
    An element Option of the list Options is a variable:
        instantiation_error
    An element Option of the list Options is neither a variable nor a compound term:
        type_error(compound,Option)
    An element Option of the list Options is a compound term but not a valid option:
        domain_error(option,Option)

Remarks:

- Registry name: Must be the URL basename when using a git URL or a local directory URL. Must also be the declared registry name in the registry specification object.
- HTTPS URLs: Must end with either a .git extension or an archive extension.
- commit(Hash) option: When present, checkout the registry git repository at the specified commit hash after cloning.

- update(Boolean) option: Update registry if already defined. Default is false. Overrides the force/1 option.
- force(Boolean) option: Force registry re-installation if already defined by first deleting the previous installation. Default is false.
- clean(Boolean) option: Clean registry archive after updating. Default is false.
- verbose(Boolean) option: Verbose adding steps. Default is false.
- downloader(Atom) option: Downloader utility. Either curl or wget. Default is curl.
- curl(Atom) option: Extra command-line options. Default is ''.
- wget(Atom) option: Extra command-line options. Default is ''.
- gpg(Atom) option: Extra command-line options. Default is ''.
- tar(Atom) option: Extra command-line options. Default is ''.

---

add/2

Adds a new registry using default options. Fails if the registry cannot be added or if it is already defined. HTTPS URLs must end with either a .git extension or an archive extension. A file:// URL can be used for a local directory or archive.

Compilation flags:
    static

Template:
    add(Registry,URL)
Mode and number of proofs:
    add(+atom,+atom) - zero_or_one

Exceptions:
    Registry is a variable:
        instantiation_error
    Registry is neither a variable nor an atom:
        type_error(atom,Registry)
    URL is a variable:
        instantiation_error
    URL is neither a variable nor an atom:
        type_error(atom,URL)

Remarks:

- Registry name: Must be the URL basename when using a git URL or a local directory URL. Must also be the declared registry name in the registry specification object.

See also:
add/3

---

add/1

Adds a new registry using default options. Fails if the registry cannot be added or if it is already defined. HTTPS URLs must end with a .git extension or an archive extension. A file:// URL can be used for a local directory or archive.

Compilation flags:
static

Template:
add(URL)
Mode and number of proofs:
add(+atom) - zero_or_one

Exceptions:
URL is a variable:
instantiation_error
URL is neither a variable nor an atom:
type_error(atom,URL)

Remarks:

- Registry name: Taken from the URL basename.

See also:
add/2

---

update/2

Updates a defined registry using the specified options. Fails if the registry is not defined.

Compilation flags:
    static

Template:
    update(Registry,Options)
Mode and number of proofs:
    update(+atom,++list(compound)) - zero_or_one

Exceptions:
    Registry is a variable:
        instantiation_error
    Registry is neither a variable nor an atom:
        type_error(atom,Registry)
    Options is a variable:
        instantiation_error
    Options is neither a variable nor a list:
        type_error(list,Options)
    An element Option of the list Options is a variable:
        instantiation_error
    An element Option of the list Options is neither a variable nor a compound term:
        type_error(compound,Option)
    An element Option of the list Options is a compound term but not a valid option:
        domain_error(option,Option)

Remarks:

- commit(Hash) option: When present, checkout the registry git repository at the specified commit hash after updating.
- force(Boolean) option: Force update if the registry is pinned. Default is false.
- clean(Boolean) option: Clean registry archive after updating. Default is false.
- verbose(Boolean) option: Verbose updating steps. Default is false.
- downloader(Atom) option: Downloader utility. Either curl or wget. Default is curl.
- curl(Atom) option: Extra command-line options. Default is ''.
- wget(Atom) option: Extra command-line options. Default is ''.
- gpg(Atom) option: Extra command-line options. Default is ''.
- tar(Atom) option: Extra command-line options. Default is ''.

update/1

Updates a defined registry using default options. Fails if the registry is not defined.

Compilation flags:
    static

Template:
    update(Registry)
Mode and number of proofs:
    update(+atom) - zero_or_one

Exceptions:
    Registry is a variable:
        instantiation_error
    Registry is neither a variable nor an atom:
        type_error(atom,Registry)

See also:
    update/2

---

update/0

Updates all defined registries using default options.

Compilation flags:
    static

Mode and number of proofs:
    update - zero_or_one

---

delete/2

Deletes a registry using the specified options (if not pinned).

Compilation flags:
    static

Template:
    delete(Registry,Options)
Mode and number of proofs:
    delete(+atom,++list(compound)) - zero_or_one

Exceptions:
    Registry is a variable:
        instantiation_error
    Registry is neither a variable nor an atom:
        type_error(atom,Registry)
    Options is a variable:
        instantiation_error
    Options is neither a variable nor a list:
        type_error(list,Options)
    An element Option of the list Options is a variable:
        instantiation_error
    An element Option of the list Options is neither a variable nor a compound term:
        type_error(compound,Option)
    An element Option of the list Options is a compound term but not a valid option:
        domain_error(option,Option)

Remarks:

- force(Boolean) option: Force deletion if the registry is pinned or there are installed registry packs. Default is false.
- clean(Boolean) option: Clean registry archive after deleting. Default is false.
- verbose(Boolean) option: Verbose deleting steps. Default is false.
- downloader(Atom) option: Downloader utility. Either curl or wget. Default is curl.
- curl(Atom) option: Extra command-line options. Default is ''.
- wget(Atom) option: Extra command-line options. Default is ''.
- gpg(Atom) option: Extra command-line options. Default is ''.
- tar(Atom) option: Extra command-line options. Default is ''.

delete/1

Deletes a registry using default options.

Compilation flags:
    static

Template:
    delete(Registry)
Mode and number of proofs:
    delete(+atom) - zero_or_one

Exceptions:
    Registry is a variable:
        instantiation_error
    Registry is neither a variable nor an atom:
        type_error(atom,Registry)

See also:
    delete/2

---

delete/0

Deletes all registries using the force(true) option.

Compilation flags:
    static

Mode and number of proofs:
    delete - zero_or_one

---

clean/1

Cleans all registry archives. Fails if the registry is not defined.

Compilation flags:
    static

Template:
    clean(Registry)
Mode and number of proofs:
    clean(+atom) - zero_or_one

Exceptions:
    Registry is a variable:
        instantiation_error
    Registry is neither a variable nor an atom:
        type_error(atom,Registry)

---

clean/0

Cleans all archives for all registries.

Compilation flags:
    static

Mode and number of proofs:
    clean - one

---

provides/2

Enumerates by backtracking all packs provided by a registry.

Compilation flags:
    static

Template:

provides(Registry,Pack)
Mode and number of proofs:
provides(?atom,?atom) - zero_or_more

Exceptions:
Registry is neither a variable nor an atom:
type_error(atom,Registry)
Pack is neither a variable nor an atom:
type_error(atom,Pack)

---

lint/1

Checks the registry specification. Fails if the registry is not defined or if linting detects errors.

Compilation flags:
static

Template:
lint(Registry)
Mode and number of proofs:
lint(+atom) - zero_or_one

Exceptions:
Registry is a variable:
instantiation_error
Registry is neither a variable nor an atom:
type_error(atom,Registry)

---

lint/0

Checks all registry specifications.

Compilation flags:
static

Mode and number of proofs:
lint - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.170.7 registry_loader_hook

Hook object for filtering registry loader file contents.

Availability:
logtalk_load(packs(loader))

Author: Paulo Moura
Version: 0:13:0
Date: 2022-11-20

Compilation flags:
static, context_switching_calls

Implements:
public expanding
Uses:
character
logtalk

Remarks:
(none)

Inherited public predicates:
goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.170.8 registry_protocol

Registry specification protocol. Objects implementing this protocol should be named after the pack with a _registry suffix and saved in a file with the same name as the object.

Availability:
    logtalk_load(packs(loader))

Author: Paulo Moura
Version: 0:12:0
Date: 2022-06-28

Compilation flags:
    static

Dependencies:
    (none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - name/1
  - description/1
  - home/1
  - clone/1
  - archive/1
  - note/2
- Protected predicates
- Private predicates
- Operators

Public predicates

name/1

Registry name. Preferably a valid unquoted atom.

Compilation flags:
static

Template:
name(Name)
Mode and number of proofs:
name(?atom) - zero_or_one

description/1

Registry one line description.

Compilation flags:
    static

Template:
    description(Description)
Mode and number of proofs:
    description(?atom) - zero_or_one

---

home/1

Registry home HTTPS or file URL.

Compilation flags:
    static

Template:
    home(Home)
Mode and number of proofs:
    home(?atom) - zero_or_one

---

clone/1

Registry git clone HTTPS URL (must end with the .git extension). Git repos should have the same name as the registry.

Compilation flags:
    static

Template:
    clone(URL)
Mode and number of proofs:
    clone(?atom) - zero_or_one

---

archive/1

Registry archive download HTTPS URL.

Compilation flags:
    static

Template:
    archive(URL)
Mode and number of proofs:
    archive(?atom) - zero_or_one

---

note/2

Table of notes per action.

Compilation flags:
    static

Template:
    note(Action,Note)
Mode and number of proofs:
    note(?atom,-atom) - zero_or_more

Remarks:

- Action: Possible values are add, update, and delete. When unbound, the note apply to all actions.
- Note: Note to print when performing an action on a registry.

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.171 partitions

object

1.171.1 partitions

Implementation of set partition operations over lists.

Availability:
    logtalk_load(partitions(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-12

Compilation flags:
    static, context_switching_calls

Implements:
    public partitions_protocol
Uses:
    fast_random(Algorithm)
    integer
    list
    natural

Remarks:
    (none)

Inherited public predicates:

count_distinct_partitions/2 count_distinct_partitions/3 count_partitions/2 count_partitions/3
distinct_partition/2 distinct_partition/3 distinct_partition/4 distinct_partition_index/3
distinct_partition_index/4 distinct_partition_index/5 distinct_partitions/2 distinct_partitions/3
distinct_partitions/4 next_partition/3 next_partition/4 nth_distinct_partition/3
nth_distinct_partition/4 nth_distinct_partition/5 nth_partition/3 nth_partition/4
nth_partition/5 partition/2 partition/3 partition/4 partition_index/3 partition_index/4
partition_index/5 partitions/2 partitions/3 partitions/4 previous_partition/3
previous_partition/4 random_distinct_partition/2 random_distinct_partition/3
random_partition/2 random_partition/3 sample_distinct_partitions/3
sample_distinct_partitions/4 sample_partitions/3 sample_partitions/4

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.171.2 partitions_protocol

Protocol for set partition operations over lists.

Availability:
logtalk_load(partitions(loader))

Author: Paulo Moura
Version: 1:0:0

Date: 2026-05-12

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - partitions/2
  - partition/2
  - partitions/3
  - partition/3
  - partitions/4
  - partition/4
  - distinct_partitions/2
  - distinct_partition/2
  - distinct_partitions/3
  - distinct_partition/3
  - distinct_partitions/4
  - distinct_partition/4
  - count_partitions/2
  - nth_partition/3
  - nth_partition/4
  - nth_partition/5
  - partition_index/3
  - partition_index/4
  - partition_index/5
  - count_partitions/3

  - count_distinct_partitions/2
  - nth_distinct_partition/3
  - nth_distinct_partition/4
  - nth_distinct_partition/5
  - distinct_partition_index/3
  - distinct_partition_index/4
  - distinct_partition_index/5
  - count_distinct_partitions/3
  - random_partition/2
  - random_partition/3
  - sample_partitions/3
  - sample_partitions/4
  - random_distinct_partition/2
  - random_distinct_partition/3
  - sample_distinct_partitions/3
  - sample_distinct_partitions/4
  - next_partition/3
  - next_partition/4
  - previous_partition/3
  - previous_partition/4
- Protected predicates
- Private predicates
- Operators

Public predicates

partitions/2

Generates all set partitions of a list using default order. Partitions are represented as lists of non-empty blocks. The empty list has one partition, represented by the empty list of blocks.

Compilation flags:
    static

Template:
    partitions(List,Partitions)
Mode and number of proofs:

partitions(+list,-list) - one

---

partition/2

True iff the second argument is a set partition of a list using default order.

Compilation flags:
static

Template:
partition(List,Partition)
Mode and number of proofs:
partition(+list,-list) - one_or_more

---

partitions/3

Generates either all set partitions with the given order (when the second argument is default or lexicographic) or all set partitions with an exact number of blocks using default order (when the first argument is an integer).

Compilation flags:
static

Template:
partitions(ListOrK,OrderOrList,Partitions)
Mode and number of proofs:
partitions(+list,+atom,-list) - one
partitions(+integer,+list,-list) - one

---

partition/3

True iff the third argument is a set partition generated either in the given order or with the given exact number of blocks using default order.

Compilation flags:
    static

Template:
    partition(ListOrK,OrderOrList,Partition)
Mode and number of proofs:
    partition(+list,+atom,-list) - one_or_more
    partition(+integer,+list,-list) - one_or_more

---

partitions/4

Generates all set partitions with an exact number of blocks and the given order: default or lexicographic.

Compilation flags:
    static

Template:
    partitions(K,List,Order,Partitions)
Mode and number of proofs:
    partitions(+integer,+list,+atom,-list) - one

---

partition/4

True iff the fourth argument is a set partition with an exact number of blocks and the given order: default or lexicographic.

Compilation flags:
    static

Template:
    partition(K,List,Order,Partition)
Mode and number of proofs:

partition(+integer,+list,+atom,-list) - one_or_more

---

distinct_partitions/2

Generates all distinct set partitions of a list (deduplicating equal-valued partitions after canonicalizing block order) using default order.

Compilation flags:
static

Template:
distinct_partitions(List,Partitions)
Mode and number of proofs:
distinct_partitions(+list,-list) - one

---

distinct_partition/2

True iff the second argument is a distinct set partition of a list using default order.

Compilation flags:
static

Template:
distinct_partition(List,Partition)
Mode and number of proofs:
distinct_partition(+list,-list) - one_or_more

---

distinct_partitions/3

Generates either all distinct set partitions with the given order (when the second argument is default or lexicographic) or all distinct set partitions with an exact number of blocks using default order (when the first argument is an integer).

Compilation flags:

static

Template:
distinct_partitions(ListOrK,OrderOrList,Partitions)
Mode and number of proofs:
distinct_partitions(+list,+atom,-list) - one
distinct_partitions(+integer,+list,-list) - one

---

distinct_partition/3

True iff the third argument is a distinct set partition generated either in the given order or with the given exact number of blocks using default order.

Compilation flags:
static

Template:
distinct_partition(ListOrK,OrderOrList,Partition)
Mode and number of proofs:
distinct_partition(+list,+atom,-list) - one_or_more
distinct_partition(+integer,+list,-list) - one_or_more

---

distinct_partitions/4

Generates all distinct set partitions with an exact number of blocks and the given order: default or lexicographic.

Compilation flags:
static

Template:
distinct_partitions(K,List,Order,Partitions)
Mode and number of proofs:
distinct_partitions(+integer,+list,+atom,-list) - one

---

distinct_partition/4

True iff the fourth argument is a distinct set partition with an exact number of blocks and the given order: default or lexicographic.

Compilation flags:
    static

Template:
    distinct_partition(K,List,Order,Partition)
Mode and number of proofs:
    distinct_partition(+integer,+list,+atom,-list) - one_or_more

---

count_partitions/2

Counts the number of set partitions of a list.

Compilation flags:
    static

Template:
    count_partitions(List,Count)
Mode and number of proofs:
    count_partitions(+list,-integer) - one

---

nth_partition/3

Returns the set partition at a given zero-based index using default order.

Compilation flags:
    static

Template:
    nth_partition(List,Index,Partition)
Mode and number of proofs:
    nth_partition(+list,+integer,-list) - zero_or_one

---

nth_partition/4

Returns either the set partition at a given zero-based index in the given order (when the second argument is default or lexicographic) or the set partition with an exact number of blocks at a given zero-based index using default order (when the first argument is an integer).

Compilation flags:
    static

Template:
    nth_partition(ListOrK,OrderOrList,Index,Partition)
Mode and number of proofs:
    nth_partition(+list,+atom,+integer,-list) - zero_or_one
    nth_partition(+integer,+list,+integer,-list) - zero_or_one

---

nth_partition/5

Returns the set partition with an exact number of blocks at a given zero-based index in the given order: default or lexicographic.

Compilation flags:
    static

Template:
    nth_partition(K,List,Order,Index,Partition)
Mode and number of proofs:
    nth_partition(+integer,+list,+atom,+integer,-list) - zero_or_one

---

partition_index/3

Returns the zero-based index of a set partition using default order.

Compilation flags:
    static

Template:
    partition_index(List,Partition,Index)
Mode and number of proofs:
    partition_index(+list,+list,-integer) - zero_or_one

---

partition_index/4

Returns either the zero-based index of a set partition in the given order (when the second argument is default or lexicographic) or the zero-based index of a set partition with an exact number of blocks using default order (when the first argument is an integer).

Compilation flags:
    static

Template:
    partition_index(ListOrK,OrderOrList,Partition,Index)
Mode and number of proofs:
    partition_index(+list,+atom,+list,-integer) - zero_or_one
    partition_index(+integer,+list,+list,-integer) - zero_or_one

---

partition_index/5

Returns the zero-based index of a set partition with an exact number of blocks in the given order: default or lexicographic.

Compilation flags:
    static

Template:
    partition_index(K,List,Order,Partition,Index)

Mode and number of proofs:
    partition_index(+integer,+list,+atom,+list,-integer) - zero_or_one

---

count_partitions/3

Counts the number of set partitions of a list with an exact number of blocks.

Compilation flags:
    static

Template:
    count_partitions(K,List,Count)
Mode and number of proofs:
    count_partitions(+integer,+list,-integer) - zero_or_one

---

count_distinct_partitions/2

Counts the number of distinct set partitions of a list (deduplicating equal-valued partitions).

Compilation flags:
    static

Template:
    count_distinct_partitions(List,Count)
Mode and number of proofs:
    count_distinct_partitions(+list,-integer) - one

---

nth_distinct_partition/3

Returns the distinct set partition at a given zero-based index using default order.

Compilation flags:
    static

Template:
nth_distinct_partition(List,Index,Partition)
Mode and number of proofs:
nth_distinct_partition(+list,+integer,-list) - zero_or_one

---

nth_distinct_partition/4

Returns either the distinct set partition at a given zero-based index in the given order (when the second argument is default or lexicographic) or the distinct set partition with an exact number of blocks at a given zero-based index using default order (when the first argument is an integer).

Compilation flags:
static

Template:
nth_distinct_partition(ListOrK,OrderOrList,Index,Partition)
Mode and number of proofs:
nth_distinct_partition(+list,+atom,+integer,-list) - zero_or_one
nth_distinct_partition(+integer,+list,+integer,-list) - zero_or_one

---

nth_distinct_partition/5

Returns the distinct set partition with an exact number of blocks at a given zero-based index in the given order: default or lexicographic.

Compilation flags:
static

Template:
nth_distinct_partition(K,List,Order,Index,Partition)
Mode and number of proofs:
nth_distinct_partition(+integer,+list,+atom,+integer,-list) - zero_or_one

---

distinct_partition_index/3

Returns the zero-based index of a distinct set partition using default order.

Compilation flags:
    static

Template:
    distinct_partition_index(List,Partition,Index)
Mode and number of proofs:
    distinct_partition_index(+list,+list,-integer) - zero_or_one

---

distinct_partition_index/4

Returns either the zero-based index of a distinct set partition in the given order (when the second argument is default or lexicographic) or the zero-based index of a distinct set partition with an exact number of blocks using default order (when the first argument is an integer).

Compilation flags:
    static

Template:
    distinct_partition_index(ListOrK,OrderOrList,Partition,Index)
Mode and number of proofs:
    distinct_partition_index(+list,+atom,+list,-integer) - zero_or_one
    distinct_partition_index(+integer,+list,+list,-integer) - zero_or_one

---

distinct_partition_index/5

Returns the zero-based index of a distinct set partition with an exact number of blocks in the given order: default or lexicographic.

Compilation flags:
    static

Template:
    distinct_partition_index(K,List,Order,Partition,Index)

Mode and number of proofs:
distinct_partition_index(+integer,+list,+atom,+list,-integer) - zero_or_one

---

count_distinct_partitions/3

Counts the number of distinct set partitions of a list with an exact number of blocks.

Compilation flags:
static

Template:
count_distinct_partitions(K,List,Count)
Mode and number of proofs:
count_distinct_partitions(+integer,+list,-integer) - zero_or_one

---

random_partition/2

Returns a random set partition of a list.

Compilation flags:
static

Template:
random_partition(List,Partition)
Mode and number of proofs:
random_partition(+list,-list) - one

---

random_partition/3

Returns a random set partition of a list with an exact number of blocks.

Compilation flags:
static

Template:
random_partition(K,List,Partition)
Mode and number of proofs:
random_partition(+integer,+list,-list) - zero_or_one

---

sample_partitions/3

Returns SampleCount random set partitions of a list, sampled with replacement.

Compilation flags:
static

Template:
sample_partitions(List,SampleCount,Samples)
Mode and number of proofs:
sample_partitions(+list,+integer,-list) - zero_or_one

---

sample_partitions/4

Returns SampleCount random set partitions of a list with an exact number of blocks, sampled with replacement.

Compilation flags:
static

Template:
sample_partitions(K,List,SampleCount,Samples)
Mode and number of proofs:
sample_partitions(+integer,+list,+integer,-list) - zero_or_one

---

random_distinct_partition/2

Returns a random distinct set partition of a list (deduplicating equal-valued partitions).

Compilation flags:
    static

Template:
    random_distinct_partition(List,Partition)
Mode and number of proofs:
    random_distinct_partition(+list,-list) - one

---

random_distinct_partition/3

Returns a random distinct set partition of a list with an exact number of blocks (deduplicating equal-valued partitions).

Compilation flags:
    static

Template:
    random_distinct_partition(K,List,Partition)
Mode and number of proofs:
    random_distinct_partition(+integer,+list,-list) - zero_or_one

---

sample_distinct_partitions/3

Returns SampleCount random distinct set partitions of a list, sampled with replacement after deduplicating equal-valued partitions.

Compilation flags:
    static

Template:
    sample_distinct_partitions(List,SampleCount,Samples)
Mode and number of proofs:
    sample_distinct_partitions(+list,+integer,-list) - zero_or_one

---

sample_distinct_partitions/4

Returns SampleCount random distinct set partitions of a list with an exact number of blocks, sampled with replacement after deduplicating equal-valued partitions.

Compilation flags:
    static

Template:
    sample_distinct_partitions(K,List,SampleCount,Samples)
Mode and number of proofs:
    sample_distinct_partitions(+integer,+list,+integer,-list) - zero_or_one

---

next_partition/3

Returns the next distinct set partition value in lexicographic order induced by the first argument.

Compilation flags:
    static

Template:
    next_partition(List,Partition,Next)
Mode and number of proofs:
    next_partition(+list,+list,-list) - zero_or_one

---

next_partition/4

Returns the next distinct set partition value with an exact number of blocks in lexicographic order induced by the second argument.

Compilation flags:
    static

Template:

next_partition(K,List,Partition,Next)
Mode and number of proofs:
next_partition(+integer,+list,+list,-list) - zero_or_one

---

previous_partition/3

Returns the previous distinct set partition value in lexicographic order induced by the first argument.

Compilation flags:
static

Template:
previous_partition(List,Partition,Previous)
Mode and number of proofs:
previous_partition(+list,+list,-list) - zero_or_one

---

previous_partition/4

Returns the previous distinct set partition value with an exact number of blocks in lexicographic order induced by the second argument.

Compilation flags:
static

Template:
previous_partition(K,List,Partition,Previous)
Mode and number of proofs:
previous_partition(+integer,+list,+list,-list) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.172 pattern_mining_protocols

category

1.172.1 pattern_miner_common

Shared predicates for pattern miner diagnostics, defaults, option handling, and export helpers.

Availability:
logtalk_load(pattern_mining_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-29

Compilation flags:
static

Implements:
public pattern_miner_protocol
Extends:
public options
Uses:
format
list
type

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_pattern_miner/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 mine/2 mine/3 option/2 option/3 pattern_miner_options/2 print_pattern_miner/1 valid_option/1 valid_options/1 valid_pattern_miner/1

- Public predicates
- Protected predicates
  - pattern_miner_diagnostics_data/2
  - pattern_miner_diagnostics/6
  - pattern_miner_export_template/4
  - valid_pattern_miner_metadata/5
  - check_item_domain/1
  - effective_support_count/3
  - effective_maximum_pattern_length/3
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

pattern_miner_diagnostics_data/2

Hook predicate that importing pattern miner implementations must define in order to expose diagnostics metadata.

Compilation flags:
static

Template:
pattern_miner_diagnostics_data(PatternMiner,Diagnostics)
Mode and number of proofs:
pattern_miner_diagnostics_data(+compound,-list(compound)) - one

pattern_miner_diagnostics/6

Builds a standard diagnostics list from generic pattern miner metadata and algorithm-specific diagnostics terms.

Compilation flags:
static

Template:
pattern_miner_diagnostics(Model,ItemDomain,Patterns,Options,SpecificDiagnostics,Diagnostics)
Mode and number of proofs:
pattern_miner_diagnostics(+atom,+list(atom),+list(compound),+list(compound),+list(compound),-list(compound)) - one

---

pattern_miner_export_template/4

Hook predicate that importing pattern miner implementations must define in order to expose the exported pattern miner template for a given functor.

Compilation flags:
static

Template:
pattern_miner_export_template(Dataset,PatternMiner,Functor,Template)
Mode and number of proofs:
pattern_miner_export_template(+object_identifier,+compound,+atom,-callable) - one

---

valid_pattern_miner_metadata/5

True when the diagnostics list matches the generic pattern miner metadata implied by the model, item domain, mined patterns, and effective options.

Compilation flags:
static

Template:
valid_pattern_miner_metadata(Model,ItemDomain,Patterns,Options,Diagnostics)

Mode and number of proofs:
valid_pattern_miner_metadata(+atom,+list(atom),+list(compound),+list(compound),
+list(compound)) - zero_or_one

---

check_item_domain/1

Validates that the declared item domain is canonical and contains only atoms.

Compilation flags:
static

Template:
check_item_domain(ItemDomain)
Mode and number of proofs:
check_item_domain(+list(atom)) - one

---

effective_support_count/3

Computes the effective minimum support count from the effective options and the dataset size.

Compilation flags:
static

Template:
effective_support_count(DatasetSize,Options,SupportCount)
Mode and number of proofs:
effective_support_count(+integer,+list(compound),-integer) - one

---

effective_maximum_pattern_length/3

Computes the effective maximum pattern length from the effective options and the dataset maximum pattern length.

Compilation flags:
static

Template:
effective_maximum_pattern_length(DatasetMaximumPatternLength,Options,
MaximumPatternLength)
Mode and number of proofs:
effective_maximum_pattern_length(+integer,+list(compound),-integer) - one

---

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.172.2 pattern_miner_protocol

Protocol for machine learning pattern-mining algorithms.

Availability:
logtalk_load(pattern_mining_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-30

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - mine/3
  - mine/2
  - diagnostics/2
  - diagnostic/2
  - pattern_miner_options/2
  - check_pattern_miner/1
  - valid_pattern_miner/1
  - export_to_clauses/4
  - export_to_file/4
  - print_pattern_miner/1
- Protected predicates
- Private predicates
- Operators

Public predicates

mine/3

Mines a pattern miner result from the given dataset object using the specified options.

Compilation flags:
static

Template:
mine(Dataset,PatternMiner,Options)
Mode and number of proofs:
mine(+object_identifier,-compound,+list(compound)) - one

mine/2

Mines a pattern miner result from the given dataset object using default options.

Compilation flags:
    static

Template:
    mine(Dataset,PatternMiner)
Mode and number of proofs:
    mine(+object_identifier,-compound) - one

---

diagnostics/2

Returns diagnostics and metadata associated with a mined pattern miner in a representation-independent way.

Compilation flags:
    static

Template:
    diagnostics(PatternMiner,Diagnostics)
Mode and number of proofs:
    diagnostics(+compound,-list(compound)) - one

---

diagnostic/2

Tests or enumerates individual diagnostics metadata terms for a mined pattern miner.

Compilation flags:
    static

Template:
    diagnostic(PatternMiner,Diagnostic)
Mode and number of proofs:
    diagnostic(+compound,?compound) - zero_or_more

---

pattern_miner_options/2

Returns the effective mining options recorded in a mined pattern miner diagnostics list.

Compilation flags:
    static

Template:
    pattern_miner_options(PatternMiner,Options)
Mode and number of proofs:
    pattern_miner_options(+compound,-list(compound)) - zero_or_one

---

check_pattern_miner/1

Checks that the argument is a structurally valid mined pattern miner term for the receiving pattern miner implementation, throwing an exception on invalid input when applicable.

Compilation flags:
    static

Template:
    check_pattern_miner(PatternMiner)
Mode and number of proofs:
    check_pattern_miner(@compound) - one_or_error

Exceptions:
    PatternMiner is a variable:
        instantiation_error
    PatternMiner is neither a variable nor a valid pattern miner:
        domain_error(pattern_miner,PatternMiner)

---

valid_pattern_miner/1

True when check_pattern_miner/1 succeeds for the argument without throwing an exception.

Compilation flags:
static

Template:
valid_pattern_miner(PatternMiner)
Mode and number of proofs:
valid_pattern_miner(@compound) - zero_or_one

---

export_to_clauses/4

Converts a pattern miner result into a list of predicate clauses. Functor is the functor for the generated predicate clauses.

Compilation flags:
static

Template:
export_to_clauses(Dataset,PatternMiner,Functor,Clauses)
Mode and number of proofs:
export_to_clauses(+object_identifier,+compound,+callable,-list(clause)) - one

---

export_to_file/4

Exports a pattern miner result to a file. Functor is the functor for the generated predicate clauses.

Compilation flags:
static

Template:
export_to_file(Dataset,PatternMiner,Functor,File)
Mode and number of proofs:
export_to_file(+object_identifier,+compound,+callable,+atom) - one

print_pattern_miner/1

Prints a pattern miner result to the current output stream in a human-readable format.

Compilation flags:
    static

Template:
    print_pattern_miner(PatternMiner)
Mode and number of proofs:
    print_pattern_miner(+compound) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

transaction_dataset_protocol, sequence_dataset_protocol

1.173 pca_projection

object

1.173.1 pca_projection

Principal Component Analysis reducer for continuous datasets using a portable power-iteration eigensolver.

Availability:
logtalk_load(pca_projection(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-07

Compilation flags:
static, context_switching_calls

Imports:
public dimension_reducer_common
Uses:
format
linear_algebra
list
pairs
type

Remarks:
(none)

Inherited public predicates:
check_dimension_reducer/1 check_option/1 check_options/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 dimension_reducer_options/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 print_dimension_reducer/1 transform/3 valid_dimension_reducer/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

lda_projection, random_projection

1.174 pddl_parser

object

1.174.1 pddl

Simple parser of PDDL 3.0 files.

Availability:
logtalk_load(pddl_parser(loader))

Author: Robert Sasak, Charles University in Prague. Adapted to Logtalk by Paulo Moura.
Version: 1:2:2
Date: 2024-03-14

Compilation flags:
static, context_switching_calls

Imports:
public read_file
Uses:
user

Remarks:
(none)

Inherited public predicates:
read_file/2

- Public predicates
  - parse_domain/3
  - parse_domain/2
  - parse_problem/2
  - parse_problem/3
- Protected predicates
- Private predicates
- Operators

Public predicates

parse_domain/3

Parses a PDDL 3.0 domain file, returning a compound term representing its contents and rest of the file. Useful when domain and problem are in one file.

Compilation flags:
static

Template:
parse_domain(File,Output,RestOfFile)
Mode and number of proofs:
parse_domain(+atom,-compound,-list(atom)) - one

parse_domain/2

Parses a PDDL 3.0 domain file, returning a compound term representing its contents.

Compilation flags:
    static

Template:
    parse_domain(File,Output)
Mode and number of proofs:
    parse_domain(+atom,-compound) - one

---

parse_problem/2

Parses a PDDL 3.0 problem file, returning a compound term representing its contents.

Compilation flags:
    static

Template:
    parse_problem(File,Output)
Mode and number of proofs:
    parse_problem(+atom,-compound) - one

---

parse_problem/3

Parses a PDDL 3.0 problem file, returning a compound term representing its contents and rest of the file. Useful when domain and problem are in one file.

Compilation flags:
    static

Template:
    parse_problem(File,Output,RestOfFile)
Mode and number of proofs:
    parse_problem(+atom,-compound,-list(atom)) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.174.2 read_file

Utility predicates for parsing a file as a list of atoms.

Availability:
logtalk_load(pddl_parser(loader))

Author: Robert Sasak, Charles University in Prague. Adapted to Logtalk by Paulo Moura.
Version: 1:0:0
Date: 2011-08-04

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - read_file/2
- Protected predicates
- Private predicates
- Operators

Public predicates

read_file/2

Reads a file character by character, parsing it into a list of atoms.

Compilation flags:
    static

Template:
    read_file(File,List)
Mode and number of proofs:
    read_file(+atom,-list(atom)) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.175 permutations

object

1.175.1 permutations

Implementation of permutations operations over lists.

Availability:
logtalk_load(permutations(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-12

Compilation flags:
static, context_switching_calls

Implements:
public permutations_protocol
Uses:
fast_random(Algorithm)
list
natural

Remarks:
(none)

Inherited public predicates:
count_distinct_permutations/2 count_permutations/2 distinct_permutation/2 distinct_permutation/3 distinct_permutation_index/3 distinct_permutations/2 distinct_permutations/3 k_permutation/3 k_permutation/4 k_permutations/3 k_permutations/4 next_permutation/2 nth_distinct_permutation/3 nth_permutation/3 nth_permutation/4 permutation/2 permutation/3 permutation_index/3 permutation_index/4 permutations/2 permutations/3 previous_permutation/2 random_distinct_permutation/2 random_permutation/2 sample_distinct_permutations/3 sample_permutations/3

- Public predicates
- Protected predicates
- Private predicates

- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.175.2 permutations_protocol

Protocol for permutations operations over lists.

Availability:
    logtalk_load(permutations(loader))

Author: Paulo Moura
Version: 2:0:0
Date: 2026-05-12

Compilation flags:
    static

Dependencies:
    (none)

Remarks:
    (none)

Inherited public predicates:

(none)

- Public predicates
  - permutations/2
  - permutation/2
  - permutations/3
  - permutation/3
  - distinct_permutations/2
  - distinct_permutation/2
  - distinct_permutations/3
  - distinct_permutation/3
  - k_permutations/3
  - k_permutation/3
  - k_permutations/4
  - k_permutation/4
  - next_permutation/2
  - previous_permutation/2
  - nth_permutation/3
  - nth_permutation/4
  - permutation_index/3
  - permutation_index/4
  - count_permutations/2
  - count_distinct_permutations/2
  - nth_distinct_permutation/3
  - distinct_permutation_index/3
  - random_permutation/2
  - sample_permutations/3
  - random_distinct_permutation/2
  - sample_distinct_permutations/3
- Protected predicates
- Private predicates
- Operators

Public predicates

permutations/2

Generates all permutations of a list using default order.

Compilation flags:
    static

Template:
    permutations(List,Permutations)
Mode and number of proofs:
    permutations(+list,-list) - one

---

permutation/2

True iff the second argument is a permutation of the first argument using default order.

Compilation flags:
    static

Template:
    permutation(List,Permutation)
Mode and number of proofs:
    permutation(+list,-list) - one_or_more

---

permutations/3

Generates all permutations with the given order: default, lexicographic, or shortlex.

Compilation flags:
    static

Template:
    permutations(List,Order,Permutations)
Mode and number of proofs:
    permutations(+list,+atom,-list) - one

---

permutation/3

True iff the third argument is a permutation with the given order: default, lexicographic, or shortlex.

Compilation flags:
    static

Template:
    permutation(List,Order,Permutation)
Mode and number of proofs:
    permutation(+list,+atom,-list) - one_or_more

---

distinct_permutations/2

Generates all distinct permutations of a list (deduplicating repeated values in the input list) using default order.

Compilation flags:
    static

Template:
    distinct_permutations(List,Permutations)
Mode and number of proofs:
    distinct_permutations(+list,-list) - one

---

distinct_permutation/2

True iff the second argument is a distinct permutation of the first argument using default order.

Compilation flags:
    static

Template:
    distinct_permutation(List,Permutation)

Mode and number of proofs:
distinct_permutation(+list,-list) - one_or_more

---

distinct_permutations/3

Generates all distinct permutations with the given order: default, lexicographic, or shortlex.

Compilation flags:
static

Template:
distinct_permutations(List,Order,Permutations)
Mode and number of proofs:
distinct_permutations(+list,+atom,-list) - one

---

distinct_permutation/3

True iff the third argument is a distinct permutation with the given order: default, lexicographic, or shortlex.

Compilation flags:
static

Template:
distinct_permutation(List,Order,Permutation)
Mode and number of proofs:
distinct_permutation(+list,+atom,-list) - one_or_more

---

k_permutations/3

Generates all K-permutations (ordered selections) of a list using default order.

Compilation flags:
static

Template:
    k_permutations(K,List,Permutations)
Mode and number of proofs:
    k_permutations(+integer,+list,-list) - one

---

k_permutation/3

True iff the third argument is a K-permutation (ordered selection) of a list using default order.

Compilation flags:
    static

Template:
    k_permutation(K,List,Permutation)
Mode and number of proofs:
    k_permutation(+integer,+list,-list) - one_or_more

---

k_permutations/4

Generates all K-permutations with the given order: default, lexicographic, or shortlex.

Compilation flags:
    static

Template:
    k_permutations(K,List,Order,Permutations)
Mode and number of proofs:
    k_permutations(+integer,+list,+atom,-list) - one

---

k_permutation/4

True iff the fourth argument is a K-permutation with the given order: default, lexicographic, or shortlex.

Compilation flags:
    static

Template:
    k_permutation(K,List,Order,Permutation)
Mode and number of proofs:
    k_permutation(+integer,+list,+atom,-list) - one_or_more

---

next_permutation/2

Returns the next permutation value in lexicographic order.

Compilation flags:
    static

Template:
    next_permutation(Permutation,Next)
Mode and number of proofs:
    next_permutation(+list,-list) - zero_or_one

---

previous_permutation/2

Returns the previous permutation value in lexicographic order.

Compilation flags:
    static

Template:
    previous_permutation(Permutation,Previous)
Mode and number of proofs:
    previous_permutation(+list,-list) - zero_or_one

---

nth_permutation/3

Returns the permutation at a given zero-based index using default order.

Compilation flags:
    static

Template:
    nth_permutation(List,Index,Permutation)
Mode and number of proofs:
    nth_permutation(+list,+integer,-list) - zero_or_one

---

nth_permutation/4

Returns the permutation at a given zero-based index in the given order: default, lexicographic, or shortlex.

Compilation flags:
    static

Template:
    nth_permutation(List,Order,Index,Permutation)
Mode and number of proofs:
    nth_permutation(+list,+atom,+integer,-list) - zero_or_one

---

permutation_index/3

Returns the zero-based index of a permutation using default order.

Compilation flags:
    static

Template:
    permutation_index(List,Permutation,Index)
Mode and number of proofs:
    permutation_index(+list,+list,-integer) - zero_or_one

---

permutation_index/4

Returns the zero-based index of a permutation in the given order: default, lexicographic, or shortlex.

Compilation flags:
    static

Template:
    permutation_index(List,Order,Permutation,Index)
Mode and number of proofs:
    permutation_index(+list,+atom,+list,-integer) - zero_or_one

---

count_permutations/2

Counts the number of permutations of a list.

Compilation flags:
    static

Template:
    count_permutations(List,Count)
Mode and number of proofs:
    count_permutations(+list,-integer) - one

---

count_distinct_permutations/2

Counts the number of distinct permutations of a list (deduplicating equal-valued permutations).

Compilation flags:
    static

Template:
    count_distinct_permutations(List,Count)
Mode and number of proofs:
    count_distinct_permutations(+list,-integer) - one

---

nth_distinct_permutation/3

Returns the distinct permutation at a given zero-based index in default generation order.

Compilation flags:
    static

Template:
    nth_distinct_permutation(List,Index,Permutation)
Mode and number of proofs:
    nth_distinct_permutation(+list,+integer,-list) - zero_or_one

---

distinct_permutation_index/3

Returns the zero-based index of a distinct permutation in default generation order.

Compilation flags:
    static

Template:
    distinct_permutation_index(List,Permutation,Index)
Mode and number of proofs:
    distinct_permutation_index(+list,+list,-integer) - zero_or_one

---

random_permutation/2

Returns a random permutation of a list.

Compilation flags:
    static

Template:
    random_permutation(List,Permutation)
Mode and number of proofs:
    random_permutation(+list,-list) - one

---

sample_permutations/3

Returns SampleCount random permutations of a list, sampled with replacement.

Compilation flags:
    static

Template:
    sample_permutations(List,SampleCount,Samples)
Mode and number of proofs:
    sample_permutations(+list,+integer,-list) - zero_or_one

---

random_distinct_permutation/2

Returns a random distinct permutation of a list (deduplicating equal-valued permutations).

Compilation flags:
    static

Template:
    random_distinct_permutation(List,Permutation)
Mode and number of proofs:
    random_distinct_permutation(+list,-list) - one

---

sample_distinct_permutations/3

Returns SampleCount random distinct permutations of a list, sampled with replacement after deduplicating equal-valued permutations.

Compilation flags:
    static

Template:
    sample_distinct_permutations(List,SampleCount,Samples)
Mode and number of proofs:
    sample_distinct_permutations(+list,+integer,-list) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.176 plackett_luce_last_ranker

object

1.176.1 plackett_luce_last_ranker

Tie-aware Plackett-Luce-last grouped-ranking ranker. Learns one positive preference strength parameter per item from a dataset object implementing the ranking_dataset_protocol protocol by fitting a last-choice Plackett-Luce model to grouped tie blocks, and returns a self-describing ranker term with diagnostics that can be used for ranking and export.

Availability:
    logtalk_load(plackett_luce_last_ranker(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
    static, context_switching_calls

Imports:
    public ranking_dataset_common
    public grouped_strength_ranker_common
    public score_ranker_common
    public plackett_luce_common
Uses:
    avltree
    format

list
type

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_ranker/1 default_option/1 default_options/1
diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 grouped_dataset_groups/2
grouped_dataset_items/2 grouped_dataset_items/3 grouped_dataset_summary/2 learn/2
learn/3 option/2 option/3 pairwise_dataset_connected_components/2
pairwise_dataset_declared_items/2 pairwise_dataset_items/2 pairwise_dataset_matchups/2
pairwise_dataset_preferences/2 pairwise_dataset_summary/2 pairwise_dataset_win_totals/2
pairwise_measurement_dataset_connected_components/2
pairwise_measurement_dataset_declared_items/2 pairwise_measurement_dataset_items/2
pairwise_measurement_dataset_measurements/2 pairwise_measurement_dataset_summary/2
print_ranker/1 rank/3 ranker_options/2 scores/2
temporal_pairwise_dataset_connected_components/2
temporal_pairwise_dataset_declared_items/2 temporal_pairwise_dataset_games/2
temporal_pairwise_dataset_games/3 temporal_pairwise_dataset_items/2
temporal_pairwise_dataset_periods/2 temporal_pairwise_dataset_summary/2 valid_option/1
valid_options/1 valid_ranker/1 validate_grouped_dataset/1 validate_grouped_dataset/2
validate_pairwise_dataset/1 validate_pairwise_dataset/2
validate_pairwise_measurement_dataset/1 validate_pairwise_measurement_dataset/2
validate_temporal_pairwise_dataset/1 validate_temporal_pairwise_dataset/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

ranking_dataset_protocol, ranker_protocol, borda_ranker

1.177 plackett_luce_ranker

object

1.177.1 plackett_luce_ranker

Tie-aware Plackett-Luce grouped-ranking ranker. Learns one positive preference strength parameter per item from a dataset object implementing the ranking_dataset_protocol protocol by fitting a top-choice Plackett-Luce model to grouped tie blocks, and returns a self-describing ranker term with diagnostics that can be used for ranking and export.

Availability:
logtalk_load(plackett_luce_ranker(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public ranking_dataset_common
public grouped_strength_ranker_common
public score_ranker_common
public plackett_luce_common
Uses:
avltree
format
list
type

Remarks:

(none)

Inherited public predicates:

check_option/1 check_options/1 check_ranker/1 default_option/1 default_options/1
diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 grouped_dataset_groups/2
grouped_dataset_items/2 grouped_dataset_items/3 grouped_dataset_summary/2 learn/2
learn/3 option/2 option/3 pairwise_dataset_connected_components/2
pairwise_dataset_declared_items/2 pairwise_dataset_items/2 pairwise_dataset_matchups/2
pairwise_dataset_preferences/2 pairwise_dataset_summary/2 pairwise_dataset_win_totals/2
pairwise_measurement_dataset_connected_components/2
pairwise_measurement_dataset_declared_items/2 pairwise_measurement_dataset_items/2
pairwise_measurement_dataset_measurements/2 pairwise_measurement_dataset_summary/2
print_ranker/1 rank/3 ranker_options/2 scores/2
temporal_pairwise_dataset_connected_components/2
temporal_pairwise_dataset_declared_items/2 temporal_pairwise_dataset_games/2
temporal_pairwise_dataset_games/3 temporal_pairwise_dataset_items/2
temporal_pairwise_dataset_periods/2 temporal_pairwise_dataset_summary/2 valid_option/1
valid_options/1 valid_ranker/1 validate_grouped_dataset/1 validate_grouped_dataset/2
validate_pairwise_dataset/1 validate_pairwise_dataset/2
validate_pairwise_measurement_dataset/1 validate_pairwise_measurement_dataset/2
validate_temporal_pairwise_dataset/1 validate_temporal_pairwise_dataset/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

ranking_dataset_protocol, ranker_protocol, plackett_luce_last_ranker, borda_ranker

1.178 pls_projection

object

1.178.1 pls_projection

Partial Least Squares projection for target-valued continuous datasets using deterministic PLS1 deflation.

Availability:
logtalk_load(pls_projection(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public dimension_reducer_common
Uses:
format
linear_algebra
list
numberlist
pairs
population

type

Remarks:
(none)

Inherited public predicates:
check_dimension_reducer/1 check_option/1 check_options/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 dimension_reducer_options/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 print_dimension_reducer/1 transform/3 valid_dimension_reducer/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

lda_projection, pca_projection

1.179 ports_profiler

object

1.179.1 ports_profiler

Predicate execution box model port profiler.

Availability:
logtalk_load(ports_profiler(loader))

Author: Paulo Moura
Version: 2:0:0
Date: 2024-05-18

Compilation flags:
static, context_switching_calls

Provides:
logtalk::debug_handler/1
logtalk::debug_handler/3
Uses:
logtalk
user

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - start/0
  - stop/0
  - data/0
  - data/1
  - data/2
  - reset/0

    - reset/1
    - port/5
    - clause_location/6
    - clause/5
- Protected predicates
- Private predicates
    - clause_location_/6
    - port_/5
    - clause_/5
    - entity_defines_/2
- Operators

Public predicates

start/0

Activates the ports profiler for followup goals.

Compilation flags:
    static

Mode and number of proofs:
    start - one

---

stop/0

Deactivates the ports profiler.

Compilation flags:
    static

Mode and number of proofs:
    stop - one

---

data/0

Prints a table with all port profiling data.

Compilation flags:
    static

Mode and number of proofs:
    data - one

---

data/1

Prints a table with all port profiling data for the specified entity.

Compilation flags:
    static

Template:
    data(Entity)
Mode and number of proofs:
    data(+entity_identifier) - one

---

data/2

Prints a table with all port profiling data for the specified entity predicate (or non-terminal).

Compilation flags:
    static

Template:
    data(Entity,Predicate)
Mode and number of proofs:
    data(+entity_identifier,+predicate_indicator) - one
    data(+entity_identifier,+non_terminal_indicator) - one

---

reset/0

Resets all port profiling data.

Compilation flags:
    static

Mode and number of proofs:
    reset - one

---

reset/1

Resets all port profiling data for the specified entity.

Compilation flags:
    static

Template:
    reset(Entity)
Mode and number of proofs:
    reset(+entity_identifier) - one

---

port/5

Enumerates, by backtracking, all collected port profiling data.

Compilation flags:
    static

Template:
    port(Port,Entity,Functor,Arity,Count)
Mode and number of proofs:
    port(?atom,?entity_identifier,?atom,?integer,?integer) - zero_or_more

---

clause_location/6

Enumerates, by backtracking, all collected profiled clause location data.

Compilation flags:
    static

Template:
    clause_location(Entity,Functor,Arity,ClauseNumber,File,BeginLine)
Mode and number of proofs:
    clause_location(?entity_identifier,?atom,?integer,?integer,?atom,?integer) - zero_or_more

---

clause/5

Enumerates, by backtracking, all collected clause profiling data.

Compilation flags:
    dynamic

Template:
    clause(Entity,Functor,Arity,ClauseNumber,Count)
Mode and number of proofs:
    clause(?entity_identifier,?atom,?integer,?integer,?integer) - zero_or_more

---

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

clause_location_/6

Internal table of collected profiled clause location data.

Compilation flags:
    dynamic

Template:
clause_location_(Entity,Functor,Arity,ClauseNumber,File,BeginLine)
Mode and number of proofs:
clause_location_(?entity_identifier,?atom,?integer,?integer,?atom,?integer) - zero_or_more

---

port_/5

Internal table of collected port profiling data.

Compilation flags:
dynamic

Template:
port_(Port,Entity,Functor,Arity,Count)
Mode and number of proofs:
port_(?atom,?entity_identifier,?atom,?integer,?integer) - zero_or_more

---

clause_/5

Internal table of collected clause profiling data.

Compilation flags:
dynamic

Template:
clause_(Entity,Functor,Arity,ClauseNumber,Count)
Mode and number of proofs:
clause_(?entity_identifier,?atom,?integer,?integer,?integer) - zero_or_more

---

entity_defines_/2

Internal cache for profiled predicates.

Compilation flags:
dynamic

Template:
entity_defines_(Entity,Predicate)
Mode and number of proofs:
entity_defines_(?entity_identifier,?predicate_indicator) - zero_or_more

---

Operators

(none)

1.180 prefix_span_pattern_miner

object

1.180.1 prefix_span_pattern_miner

PrefixSpan sequential pattern miner for sequence datasets using recursive projected databases.

Availability:
logtalk_load(prefix_span_pattern_miner(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public sequential_pattern_mining_common
Uses:
avltree
format

list

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_pattern_miner/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 mine/2 mine/3 option/2 option/3 pattern_miner_options/2 print_pattern_miner/1 valid_option/1 valid_options/1 valid_pattern_miner/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

pattern_miner_protocol, sequence_dataset_protocol

1.181 probabilistic_pca_projection

object

1.181.1 probabilistic_pca_projection

Probabilistic Principal Component Analysis reducer for continuous datasets using a portable covariance eigensolver.

Availability:
logtalk_load(probabilistic_pca_projection(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public dimension_reducer_common
Uses:
format
linear_algebra
list
numberlist
pairs
type

Remarks:
(none)

Inherited public predicates:
check_dimension_reducer/1 check_option/1 check_options/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 dimension_reducer_options/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 print_dimension_reducer/1 transform/3 valid_dimension_reducer/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates

- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

kernel_pca_projection, pca_projection, truncated_svd_projection

1.182 process

object

1.182.1 process

Portable process handling predicates.

Availability:
logtalk_load(process(loader))

Author: Paulo Moura
Version: 1:0:1
Date: 2026-02-04

Compilation flags:
static, context_switching_calls

Dependencies:
(none)

Remarks:

- Supported backend Prolog systems: ECLiPSe, GNU Prolog, SICStus Prolog, SWI-Prolog, Trealla Prolog, and XVM.

Inherited public predicates:
(none)

- Public predicates
  - create/3
  - wait/2
  - kill/2
  - kill/1
- Protected predicates
- Private predicates
- Operators

Public predicates

create/3

Creates a new process from the given executable and list of arguments. Supported options are process(Pid), stdin(Stream), stdout(Stream), and stderr(Stream).

Compilation flags:
static

Template:
create(Executable,Arguments,Options)
Mode and number of proofs:
create(+atom,+list(atom),+list(compound)) - zero_or_one

wait/2

Waits for a process to terminate and retrieves its exit status.

Compilation flags:
    static

Template:
    wait(Process,Status)
Mode and number of proofs:
    wait(+process_or_pid,-integer) - zero_or_one

---

kill/2

Kills the given process with the specified signal (an integer or one of the following atoms: sighup, sigint, sigkill, or sigterm).

Compilation flags:
    static

Template:
    kill(Process,Signal)
Mode and number of proofs:
    kill(+process_or_pid,+atom_or_integer) - zero_or_one

---

kill/1

Kills the given process using the default signal (sigkill).

Compilation flags:
    static

Template:
    kill(Process)
Mode and number of proofs:
    kill(+process_or_pid) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

os

1.183 protobuf

object

1.183.1 protobuf

Google Protocol Buffers binary format parser and generator.

Availability:
logtalk_load(protobuf(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-05-09

Compilation flags:
static, context_switching_calls

Uses:
ieee_754(Precision,ByteOrder,NaNRepresentation)
json(ObjectRepresentation,PairRepresentation,StringRepresentation)
list
reader

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - parse/2
  - parse/3
  - generate/3
  - generate/4
- Protected predicates
- Private predicates
- Operators

Public predicates

parse/2

Parses Protocol Buffers binary data from the given source (bytes(List), stream(Stream), or file(Path)) returning a Schema-Data pair. When the schema is not present in the file, Schema is unified with false.

Compilation flags:
static

Template:
parse(Source,Schema-Data)
Mode and number of proofs:
parse(++compound,--pair) - one_or_error

parse/3

Parses Protocol Buffers binary data from the given source using the provided schema, returning the decoded data.

Compilation flags:
    static

Template:
    parse(Source,Schema,Data)
Mode and number of proofs:
    parse(++compound,++term,--term) - one_or_error

---

generate/3

Generates Protocol Buffers binary data to the given sink (bytes(List), stream(Stream), or file(Path)) from the given schema and data. The schema is not included in the output.

Compilation flags:
    static

Template:
    generate(Sink,Schema,Data)
Mode and number of proofs:
    generate(++compound,++term,++term) - one_or_error

---

generate/4

Generates Protocol Buffers binary data to the given sink from the given schema and data. When IncludeSchema is true, the schema is embedded in a wrapper message.

Compilation flags:
    static

Template:
    generate(Sink,IncludeSchema,Schema,Data)
Mode and number of proofs:

generate(++compound,++boolean,++term,++term) - one_or_error

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.184 qda_classifier

object

1.184.1 qda_classifier

Quadratic Discriminant Analysis classifier for continuous datasets using class-specific regularized covariance models.

Availability:
logtalk_load(qda_classifier(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-11

Compilation flags:
static, context_switching_calls

Imports:
public classifier_common
Uses:
format
linear_algebra
list
type

Remarks:
(none)

Inherited public predicates:
check_classifier/1 check_option/1 check_options/1 classifier_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 print_classifier/1 valid_classifier/1 valid_option/1 valid_options/1

- Public predicates
  - predict_scores/3
- Protected predicates
- Private predicates
- Operators

Public predicates

predict_scores/3

Predicts class discriminant scores for a new instance.

Compilation flags:
static

Template:
predict_scores(Classifier,Instance,Scores)
Mode and number of proofs:
predict_scores(+compound,+list,-list) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

dataset_protocol, lda_classifier, nearest_centroid_classifier

1.185 queues

object

1.185.1 queue

Queue predicates implemented using difference lists.

Availability:
logtalk_load(queues(loader))

Author: Paulo Moura
Version: 1:3:0
Date: 2020-12-09

Compilation flags:
static, context_switching_calls

Implements:
public queuep
Extends:
public compound

Remarks:
(none)

Inherited public predicates:

(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 append/3 as_list/2 check/1 depth/2 empty/1 ground/1 head/2 join/3 join_all/3 jump/3 jump_all/3 jump_all_block/3 length/2 map/2 map/3 new/1 numbervars/1 numbervars/3 occurs/2 serve/3 singletons/2 subsumes/2 subterm/2 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.185.2 queuep

Queue protocol.

Availability:
logtalk_load(queues(loader))

Author: Paulo Moura
Version: 1:3:0
Date: 2020-12-09

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - empty/1
  - head/2
  - join/3
  - join_all/3
  - jump/3
  - jump_all/3
  - jump_all_block/3
  - append/3
  - length/2
  - serve/3
  - as_list/2
  - map/2
  - map/3
- Protected predicates
- Private predicates
- Operators

Public predicates

empty/1

True if the queue is empty.

Compilation flags:
static

Template:

empty(Queue)
Mode and number of proofs:
empty(@queue) - zero_or_one

---

head/2

Unifies Head with the first element of the queue.

Compilation flags:
static

Template:
head(Queue,Head)
Mode and number of proofs:
head(+queue,?term) - zero_or_one

---

join/3

Adds the new element at the end of the queue.

Compilation flags:
static

Template:
join(Element,Queue,NewQueue)
Mode and number of proofs:
join(@term,+queue,-queue) - zero_or_one

---

join_all/3

Adds the new elements at the end of the queue. The elements are added in the same order that they appear in the list.

Compilation flags:
    static

Template:
    join_all(List,Queue,NewQueue)
Mode and number of proofs:
    join_all(+list,+queue,-queue) - zero_or_one

---

jump/3

Adds the new element at the front of the queue.

Compilation flags:
    static

Template:
    jump(Element,Queue,NewQueue)
Mode and number of proofs:
    jump(@term,+queue,-queue) - zero_or_one

---

jump_all/3

Adds the new elements at the front of the queue. The last element in the list will be at the front of the queue.

Compilation flags:
    static

Template:
    jump_all(Elements,Queue,NewQueue)
Mode and number of proofs:
    jump_all(+list,+queue,-queue) - zero_or_one

---

jump_all_block/3

Adds the new elements as a block at the front of the queue. The first element in the list will be at the front of the queue.

Compilation flags:
    static

Template:
    jump_all_block(Elements,Queue,NewQueue)
Mode and number of proofs:
    jump_all_block(+list,+queue,-queue) - zero_or_one

---

append/3

Appends two queues. The new queue will have the elements of the first queue followed by the elements of the second queue.

Compilation flags:
    static

Template:
    append(Queue1,Queue2,NewQueue)
Mode and number of proofs:
    append(+queue,+queue,-queue) - one

---

length/2

Queue length.

Compilation flags:
    static

Template:

length(Queue,Length)
Mode and number of proofs:
length(+heap,?integer) - zero_or_one

---

serve/3

Removes the first element of the queue for service.

Compilation flags:
static

Template:
serve(Queue,Head,NewQueue)
Mode and number of proofs:
serve(+queue,?term,-queue) - zero_or_one

---

as_list/2

Converts a queue to a list.

Compilation flags:
static

Template:
as_list(Queue,List)
Mode and number of proofs:
as_list(+queue,-list) - one

---

map/2

Applies a closure to all elements of a queue.

Compilation flags:
    static

Template:
    map(Closure,Queue)
Meta-predicate template:
    map(1,*)
Mode and number of proofs:
    map(+callable,+queue) - zero_or_one

---

map/3

Applies a closure to all elements of a queue constructing a new queue.

Compilation flags:
    static

Template:
    map(Closure,Queue,NewQueue)
Meta-predicate template:
    map(2,*,*)
Mode and number of proofs:
    map(+callable,+queue,?queue) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

queue

1.186 random

object

1.186.1 backend_random

Random number generator predicates using the backend Prolog compiler built-in random generator.

Availability:
logtalk_load(random(loader))

Author: Paulo Moura
Version: 1:22:0
Date: 2026-02-11

Compilation flags:
static, context_switching_calls

Implements:
public pseudo_random_protocol
public sampling_protocol
Uses:
list

Remarks:

- Implementation: The backend Prolog compiler built-in random generator is only used for the basic random/1, get_seed/1, and set_seed/1 predicates.

- Portability: B-Prolog, CxProlog, ECLiPSe, JIProlog, Qu-Prolog, and Quintus Prolog do not provide implementations for the get_seed/1 and set_seed/1 predicates and calling these predicates simply succeed without performing any action.

Inherited public predicates:

bernoulli/2 beta/3 between/3 binomial/3 chi_squared/2 circular_uniform_cartesian/3 circular_uniform_polar/3 dirichlet/2 enumerate/2 exponential/2 fisher/3 gamma/3 geometric/2 get_seed/1 gumbel/3 hypergeometric/4 logistic/3 lognormal/3 logseries/2 maybe/0 maybe/1 maybe/2 maybe_call/1 maybe_call/2 member/2 normal/3 permutation/2 poisson/2 power/2 random/1 random/3 randseq/4 randset/4 select/3 select/4 sequence/4 set/4 set_seed/1 standard_cauchy/3 standard_exponential/1 standard_gamma/2 standard_normal/1 standard_t/2 swap/2 swap_consecutive/2 triangular/4 uniform/1 uniform/3 von_mises/3 wald/3 weibull/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

random, fast_random

object

1.186.2 fast_random

Portable random number generator predicates. Core predicates originally written by Richard O'Keefe. Based on algorithm AS 183 from Applied Statistics.

Availability:
logtalk_load(random(loader))

Author: Paulo Moura
Version: 3:0:0
Date: 2026-01-25

Compilation flags:
static, context_switching_calls

Extends:
public fast_random(as183)

Remarks:

- Single random number generator: This object provides a faster version of the random library object but does not support being extended to define multiple random number generators.
- Randomness: Loading this object always initializes the random generator seed to the same value, thus providing a pseudo random number generator. The randomize/1 predicate can be used to initialize the seed with a random value.

Inherited public predicates:
bernoulli/2 beta/3 between/3 binomial/3 chi_squared/2 circular_uniform_cartesian/3 circular_uniform_polar/3 dirichlet/2 enumerate/2 exponential/2 fisher/3 gamma/3 geometric/2 get_seed/1 gumbel/3 hypergeometric/4 logistic/3 lognormal/3 logseries/2 maybe/0 maybe/1 maybe/2 maybe_call/1 maybe_call/2 member/2 normal/3 permutation/2 poisson/2 power/2 random/1 random/3 randomize/1 randseq/4 randset/4 reset_seed/0 select/3 select/4 sequence/4 set/4 set_seed/1 standard_cauchy/3 standard_exponential/1 standard_gamma/2 standard_normal/1 standard_t/2 swap/2 swap_consecutive/2 triangular/4 uniform/1 uniform/3 von_mises/3 wald/3 weibull/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

fast_random, random, random(Algorithm), backend_random

object

1.186.3 fast_random(Algorithm)

- Algorithm - Random number generator algorithm. One of as183, splitmix64, xoshiro128pp, xoshiro128ss, xoshiro256pp, xoshiro256ss, well512a.

Fast portable random number generator predicates.

Availability:
logtalk_load(random(loader))

Author: Paulo Moura
Version: 3:0:0
Date: 2026-02-23

Compilation flags:
static, context_switching_calls

Implements:
public pseudo_random_protocol
public sampling_protocol
Uses:
list

Remarks:

- Single random number generator: This object provides a faster version of the random library object but does not support being extended to define multiple random number generators.
- Randomness: Loading this object always initializes the random generator seed to the same value, thus providing a pseudo random number generator. The randomize/1 predicate can be used to initialize the seed with a random value.
- as183: Algorithm AS 183 from Applied Statistics. 32-bit PRNG with period 2^60. Not cryptographically secure.
- xoshiro128pp: Xoshiro128++ random number generator. 32-bit state-of-the-art PRNG with period 2^128-1. Algorithm by David Blackman and Sebastiano Vigna.
- xoshiro128ss: Xoshiro128** random number generator. 32-bit PRNG with period 2^128-1. Algorithm by David Blackman and Sebastiano Vigna.
- xoshiro256pp: Xoshiro256++ random number generator. 64-bit state-of-the-art PRNG with period 2^256-1. Algorithm by David Blackman and Sebastiano Vigna.
- xoshiro256ss: Xoshiro256** random number generator. 64-bit PRNG with period 2^256-1. Algorithm by David Blackman and Sebastiano Vigna.
- splitmix64: SplitMix64 random number generator. 64-bit PRNG primarily used for seeding other generators. Algorithm by Guy L. Steele Jr. et al.
- well512a: WELL512a 16x32-bit state-of-the-art PRNG with period 2^512-1. Algorithm by by François Panneton, Pierre L'Ecuyer, and Makoto Matsumoto.
- Algorithms backend compatibility: The SplitMix64, Xoshiro256++, and Xoshiro256** algorithms require support for unbound integer arithmetic.

Inherited public predicates:

bernoulli/2 beta/3 between/3 binomial/3 chi_squared/2 circular_uniform_cartesian/3 circular_uniform_polar/3 dirichlet/2 enumerate/2 exponential/2 fisher/3 gamma/3 geometric/2 get_seed/1 gumbel/3 hypergeometric/4 logistic/3 lognormal/3 logseries/2 maybe/0 maybe/1 maybe/2 maybe_call/1 maybe_call/2 member/2 normal/3 permutation/2 poisson/2 power/2 random/1 random/3 randseq/4 randset/4 select/3 select/4 sequence/4 set/4 set_seed/1 standard_cauchy/3 standard_exponential/1 standard_gamma/2 standard_normal/1 standard_t/2 swap/2 swap_consecutive/2 triangular/4 uniform/1 uniform/3 von_mises/3 wald/3 weibull/3

- Public predicates
  - reset_seed/0
  - randomize/1
- Protected predicates
- Private predicates
  - seed_/2
- Operators

Public predicates

reset_seed/0

Resets the random generator seed to its default value. Use get_seed/1 and set_seed/1 instead if you need reproducibility.

Compilation flags:
    static, synchronized

Mode and number of proofs:
    reset_seed - one

---

randomize/1

Randomizes the random generator using a positive integer to compute a new seed. Use of a large integer is recommended. In alternative, when using a small integer argument, discard the first dozen random values.

Compilation flags:
    static, synchronized

Template:
    randomize(Seed)
Mode and number of proofs:
    randomize(+positive_integer) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

seed_/2

Stores the current random generator seed values (a ground but otherwise opaque term).

Compilation flags:
    dynamic

Template:
seed_(Algorithm,Seed)
Mode and number of proofs:
seed_(+atom,-ground) - one

Operators

(none)

See also

fast_random, random(Algorithm), random, backend_random

protocol

1.186.4 pseudo_random_protocol

Pseudo-random number generator protocol for seed handling predicates. These predicates are declared as synchronized when the library is compiled using a backend supporting threads.

Availability:
logtalk_load(random(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2021-02-21

Compilation flags:
static

Extends:
public random_protocol

Remarks:
(none)

Inherited public predicates:
between/3 enumerate/2 maybe/0 maybe/1 maybe/2 maybe_call/1 maybe_call/2 member/2 permutation/2 random/1 random/3 randseq/4 randset/4 select/3 select/4 sequence/4 set/4 swap/2 swap_consecutive/2

- Public predicates
  - get_seed/1
  - set_seed/1
- Protected predicates
- Private predicates
- Operators

Public predicates

get_seed/1

Gets the current random generator seed. Seed should be regarded as an opaque ground term.

Compilation flags:
    static, synchronized

Template:
    get_seed(Seed)
Mode and number of proofs:
    get_seed(-ground) - one

---

set_seed/1

Sets the random generator seed to a given value returned by calling the get_seed/1 predicate.

Compilation flags:
    static, synchronized

Template:
    set_seed(Seed)
Mode and number of proofs:
    set_seed(+ground) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

random, backend_random, fast_random

object

1.186.5 random

Portable random number generator predicates. Core predicates originally written by Richard O'Keefe. Based on algorithm AS 183 from Applied Statistics.

Availability:
logtalk_load(random(loader))

Author: Paulo Moura
Version: 3:0:0
Date: 2026-01-24

Compilation flags:
static, context_switching_calls

Extends:
public random(as183)

Remarks:

- Multiple random number generators: To define multiple random number generators, simply extend this object. The derived objects must send to self the reset_seed/0 message.
- Randomness: Loading this object always initializes the random generator seed to the same value, thus providing a pseudo random number generator. The randomize/1 predicate can be used to initialize the seed with a random value.

Inherited public predicates:

bernoulli/2 beta/3 between/3 binomial/3 chi_squared/2 circular_uniform_cartesian/3 circular_uniform_polar/3 dirichlet/2 enumerate/2 exponential/2 fisher/3 gamma/3 geometric/2 get_seed/1 gumbel/3 hypergeometric/4 logistic/3 lognormal/3 logseries/2 maybe/0 maybe/1 maybe/2 maybe_call/1 maybe_call/2 member/2 normal/3 permutation/2 poisson/2 power/2 random/1 random/3 randomize/1 randseq/4 randset/4 reset_seed/0 select/3 select/4 sequence/4 set/4 set_seed/1 standard_cauchy/3 standard_exponential/1 standard_gamma/2 standard_normal/1 standard_t/2 swap/2 swap_consecutive/2 triangular/4 uniform/1 uniform/3 von_mises/3 wald/3 weibull/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

fast_random, backend_random

object

1.186.6 random(Algorithm)

- Algorithm - Random number generator algorithm. One of as183, splitmix64, xoshiro128pp, xoshiro128ss, xoshiro256pp, xoshiro256ss, well512a.

Portable random number generator predicates.

Availability:
logtalk_load(random(loader))

Author: Paulo Moura
Version: 3:0:0
Date: 2026-02-23

Compilation flags:
static, context_switching_calls

Implements:
public pseudo_random_protocol
public sampling_protocol
Uses:
list

Remarks:

- Multiple random number generators: To define multiple random number generators, simply extend this object. The derived objects must send to self the reset_seed/0 message.
- Randomness: Loading this object always initializes the random generator seed to the same value, thus providing a pseudo random number generator. The randomize/1 predicate can be used to initialize the seed with a random value.
- as183: Algorithm AS 183 from Applied Statistics. 32-bit PRNG with period 2^60. Not cryptographically secure.
- xoshiro128pp: Xoshiro128++ random number generator. 32-bit state-of-the-art PRNG with period 2^128-1. Algorithm by David Blackman and Sebastiano Vigna.
- xoshiro128ss: Xoshiro128** random number generator. 32-bit PRNG with period 2^128-1. Algorithm by David Blackman and Sebastiano Vigna.
- xoshiro256pp: Xoshiro256++ random number generator. 64-bit state-of-the-art PRNG with period 2^256-1. Algorithm by David Blackman and Sebastiano Vigna.
- xoshiro256ss: Xoshiro256** random number generator. 64-bit PRNG with period 2^256-1. Algorithm by David Blackman and Sebastiano Vigna.
- splitmix64: SplitMix64 random number generator. 64-bit PRNG primarily used for seeding other generators. Algorithm by Guy L. Steele Jr. et al.
- well512a: WELL512a 16x32-bit state-of-the-art PRNG with period 2^512-1. Algorithm by by François Panneton, Pierre L'Ecuyer, and Makoto Matsumoto.

- Algorithms backend compatibility: The SplitMix64, Xoshiro256++, and Xoshiro256** algorithms require support for unbound integer arithmetic.

Inherited public predicates:
bernoulli/2 beta/3 between/3 binomial/3 chi_squared/2 circular_uniform_cartesian/3 circular_uniform_polar/3 dirichlet/2 enumerate/2 exponential/2 fisher/3 gamma/3 geometric/2 get_seed/1 gumbel/3 hypergeometric/4 logistic/3 lognormal/3 logseries/2 maybe/0 maybe/1 maybe/2 maybe_call/1 maybe_call/2 member/2 normal/3 permutation/2 poisson/2 power/2 random/1 random/3 randseq/4 randset/4 select/3 select/4 sequence/4 set/4 set_seed/1 standard_cauchy/3 standard_exponential/1 standard_gamma/2 standard_normal/1 standard_t/2 swap/2 swap_consecutive/2 triangular/4 uniform/1 uniform/3 von_mises/3 wald/3 weibull/3

- Public predicates
  - reset_seed/0
  - randomize/1
- Protected predicates
- Private predicates
  - seed_/2
- Operators

Public predicates

reset_seed/0

Resets the random generator seed to its default value. Use get_seed/1 and set_seed/1 instead if you need reproducibility.

Compilation flags:
static, synchronized

Mode and number of proofs:
reset_seed - one

randomize/1

Randomizes the random generator using a positive integer to compute a new seed. Use of a large integer is recommended. In alternative, when using a small integer argument, discard the first dozen random values.

Compilation flags:
    static, synchronized

Template:
    randomize(Seed)
Mode and number of proofs:
    randomize(+positive_integer) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

seed_/2

Stores the current random generator seed (a ground but otherwise opaque term).

Compilation flags:
    dynamic

Template:
    seed_(Algorithm,Seed)
Mode and number of proofs:
    seed_(+atom,-ground) - one

Operators

(none)

See also

random, fast_random(Algorithm), backend_random

protocol

1.186.7 random_protocol

Random number generator protocol. The predicates are declared as synchronized when the library is compiled using a backend supporting threads.

Availability:
logtalk_load(random(loader))

Author: Paulo Moura
Version: 3:3:0
Date: 2023-11-24

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - random/1
  - between/3
  - member/2
  - select/3

  - select/4
  - swap/2
  - swap_consecutive/2
  - enumerate/2
  - permutation/2
  - sequence/4
  - set/4
  - random/3
  - randseq/4
  - randset/4
  - maybe/0
  - maybe/1
  - maybe/2
  - maybe_call/1
  - maybe_call/2
- Protected predicates
- Private predicates
- Operators

Public predicates

random/1

Returns a new random float value in the interval [0.0, 1.0[.

Compilation flags:
    static, synchronized

Template:
    random(Random)
Mode and number of proofs:
    random(-float) - one

between/3

Returns a new random integer in the interval [Lower, Upper]. Fails if Lower or Upper are not integers or if Lower > Upper.

Compilation flags:
    static

Template:
    between(Lower,Upper,Random)
Mode and number of proofs:
    between(+integer,+integer,-integer) - zero_or_one

---

member/2

Returns a random member of a list. Fails if the list is empty.

Compilation flags:
    static

Template:
    member(Random,List)
Mode and number of proofs:
    member(-term,+list(term)) - zero_or_one

---

select/3

Returns a random member of a list and the rest of the list. Fails if the list is empty.

Compilation flags:
    static

Template:
    select(Random,List,Rest)
Mode and number of proofs:
    select(-term,+list(term),-list(term)) - zero_or_one

---

select/4

Returns a random member of a list, replacing it with a new element and returning the resulting list.

Compilation flags:
    static

Template:
    select(Random,OldList,New,NewList)
Mode and number of proofs:
    select(-term,+list(term),@term,-list(term)) - zero_or_one

---

swap/2

Swaps two randomly selected elements of a list. Fails if the list is empty or contains a single element.

Compilation flags:
    static

Template:
    swap(OldList,NewList)
Mode and number of proofs:
    swap(-term,+list(term)) - zero_or_one

---

swap_consecutive/2

Swaps two randomly selected consecutive elements of a list. Fails if the list is empty or contains a single element.

Compilation flags:
    static

Template:
    swap_consecutive(OldList,NewList)

Mode and number of proofs:
swap_consecutive(-term,+list(term)) - zero_or_one

---

enumerate/2

Enumerates the elements of a list in random order. Fails if the list is empty.

Compilation flags:
static

Template:
enumerate(List,Random)
Mode and number of proofs:
enumerate(+list(term),--term) - zero_or_more

---

permutation/2

Returns a random permutation of a list.

Compilation flags:
static, synchronized

Template:
permutation(List,Permutation)
Mode and number of proofs:
permutation(+list,-list) - one

---

sequence/4

Returns list of random integers of given length in random order in interval [Lower, Upper]. Fails if Length, Lower, or Upper are not integers or if Lower > Upper.

Compilation flags:
static, synchronized

Template:
sequence(Length,Lower,Upper,List)
Mode and number of proofs:
sequence(+integer,+integer,+integer,-list(integer)) - zero_or_one

---

set/4

Returns ordered set of random integers of given size in interval [Lower, Upper]. Fails if Length, Lower, or Upper are not integers, if Lower > Upper, or if Length > Upper - Lower + 1.

Compilation flags:
static, synchronized

Template:
set(Length,Lower,Upper,Set)
Mode and number of proofs:
set(+integer,+integer,+integer,-list(integer)) - zero_or_one

---

random/3

Returns a new random value in the interval [Lower, Upper[. Fails if Lower > Upper. Deprecated. Use between/3 for integers.

Compilation flags:
static, synchronized

Template:
random(Lower,Upper,Random)
Mode and number of proofs:
random(+integer,+integer,-integer) - zero_or_one
random(+float,+float,-float) - zero_or_one

---

randseq/4

Returns list of random values of given length in random order in interval [Lower, Upper[. Fails if Lower > Upper or if the arguments are neither integers or floats. Deprecated. Use sequence/4 for integers.

Compilation flags:
    static, synchronized

Template:
    randseq(Length,Lower,Upper,List)
Mode and number of proofs:
    randseq(+integer,+integer,+integer,-list(integer)) - zero_or_one
    randseq(+integer,+float,+float,-list(float)) - zero_or_one

---

randset/4

Returns ordered set of random values of given size in interval [Lower, Upper[. Fails if the arguments are neither integers or floats, Lower > Upper, or Length > Upper - Lower when arguments are integers. Deprecated. Use set/4 for integers.

Compilation flags:
    static, synchronized

Template:
    randset(Length,Lower,Upper,Set)
Mode and number of proofs:
    randset(+integer,+integer,+integer,-list(integer)) - zero_or_one
    randset(+integer,+float,+float,-list(float)) - zero_or_one

---

maybe/0

Succeeds or fails with equal probability.

Compilation flags:
    static

Mode and number of proofs:

maybe - zero_or_one

---

maybe/1

Succeeds with probability Probability or fails with probability 1 - Probability. Fails if Probability is not a float or is outside the interval [0.0, 1.0].

Compilation flags:
static

Template:
maybe(Probability)
Mode and number of proofs:
maybe(+probability) - zero_or_one

---

maybe/2

Succeeds with probability K/N where K and N are integers satisfying the equation 0 =< K =< N. Fails otherwise.

Compilation flags:
static

Template:
maybe(K,N)
Mode and number of proofs:
maybe(+non_negative_integer,+non_negative_integer) - zero_or_one

---

maybe_call/1

Calls a goal or fails without calling it with equal probability. When the goal is called, it determines if this predicate succeeds once or fails.

Compilation flags:
    static

Template:
    maybe_call(Goal)
Meta-predicate template:
    maybe_call(0)
Mode and number of proofs:
    maybe_call(+callable) - zero_or_one

---

maybe_call/2

Calls a goal or fails without calling it with probability Probability. When the goal is called, it determines if this predicate succeeds once or fails.

Compilation flags:
    static

Template:
    maybe_call(Probability,Goal)
Meta-predicate template:
    maybe_call(*,0)
Mode and number of proofs:
    maybe_call(+probability,+callable) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

random, backend_random, fast_random

protocol

1.186.8 sampling_protocol

Predicates for sampling probability distributions.

Availability:
logtalk_load(random(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2025-02-25

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - normal/3
  - lognormal/3

    - wald/3
    - chi_squared/2
    - fisher/3
    - logseries/2
    - geometric/2
    - hypergeometric/4
    - exponential/2
    - binomial/3
    - bernoulli/2
    - beta/3
    - gamma/3
    - logistic/3
    - poisson/2
    - power/2
    - weibull/3
    - uniform/3
    - uniform/1
    - triangular/4
    - von_mises/3
    - gumbel/3
    - dirichlet/2
    - circular_uniform_polar/3
    - circular_uniform_cartesian/3
    - standard_t/2
    - standard_cauchy/3
    - standard_exponential/1
    - standard_gamma/2
    - standard_normal/1
- Protected predicates
- Private predicates
- Operators

Public predicates

normal/3

Returns a scaled normally (Gaussian) distributed random value with the given mean and standard deviation.

Compilation flags:
    static

Template:
    normal(Mean,Deviation,Value)
Mode and number of proofs:
    normal(+float,+non_negative_float,-float) - one

---

lognormal/3

Returns a scaled log normally distributed random value with the given mean and standard deviation for the normal distribution.

Compilation flags:
    static

Template:
    lognormal(Mean,Deviation,Value)
Mode and number of proofs:
    lognormal(+float,+non_negative_float,-float) - one

---

wald/3

Returns a scaled Wald (inverse Gaussian) distributed random value with the given mean.

Compilation flags:
    static

Template:
    wald(Mean,Scale,Value)
Mode and number of proofs:

wald(+positive_float,+positive_float,-float) - one

---

chi_squared/2

Returns a chi-squared distributed random value given the degrees of freedom.

Compilation flags:
static

Template:
chi_squared(DegreesOfFreedom,Value)
Mode and number of proofs:
chi_squared(+positive_integer,-float) - one

---

fisher/3

Returns a Fisher distributed random value given the degrees of freedom in the numerator and in the denominator.

Compilation flags:
static

Template:
fisher(DegreesOfFreedomNumerator,DegreesOfFreedomDenominator,Value)
Mode and number of proofs:
fisher(+positive_integer,+positive_integer,-float) - one

---

logseries/2

Returns a logseries distributed random value. Requires $0.0 < \text{Shape} < 1$ and fails otherwise.

Compilation flags:
static

Template:
    logseries(Shape,Value)
Mode and number of proofs:
    logseries(+non_negative_integer,-positive_integer) - zero_or_one

---

geometric/2

Returns a geometric distributed random value (trials until the first success).

Compilation flags:
    static

Template:
    geometric(Probability,Value)
Mode and number of proofs:
    geometric(+probability,-positive_integer) - one

---

hypergeometric/4

Returns a hypergeometric distributed random value.

Compilation flags:
    static

Template:
    hypergeometric(Population,Successes,Draws,Value)
Mode and number of proofs:
    hypergeometric(+non_negative_integer,+non_negative_integer,+non_negative_integer,
    -non_negative_integer) - one

---

exponential/2

Returns a scaled exponentially distributed random value.

Compilation flags:
    static

Template:
    exponential(Scale,Value)
Mode and number of proofs:
    exponential(+positive_float,-float) - one

---

binomial/3

Returns a binomial distributed random value.

Compilation flags:
    static

Template:
    binomial(Trials,Probability,Value)
Mode and number of proofs:
    binomial(+positive_integer,+positive_float,-float) - one

---

bernoulli/2

Returns a Bernoulli distributed random value.

Compilation flags:
    static

Template:
    bernoulli(Probability,Value)
Mode and number of proofs:
    bernoulli(+positive_integer,-float) - one

---

beta/3

Returns a beta distributed random value.

Compilation flags:
    static

Template:
    beta(Alpha,Beta,Value)
Mode and number of proofs:
    beta(+positive_float,+positive_float,-float) - one

---

gamma/3

Returns a scaled gamma distributed random value.

Compilation flags:
    static

Template:
    gamma(Shape,Scale,Value)
Mode and number of proofs:
    gamma(+positive_float,+positive_float,-float) - one

---

logistic/3

Returns a scaled logistic distributed random value.

Compilation flags:
    static

Template:
    logistic(Location,Scale,Value)
Mode and number of proofs:
    logistic(+float,+positive_float,-float) - one

---

poisson/2

Returns a Poisson distributed random value given the expected number of events.

Compilation flags:
    static

Template:
    poisson(Mean,Value)
Mode and number of proofs:
    poisson(+non_negative_float,-non_negative_integer) - one

---

power/2

Returns a power distributed random value.

Compilation flags:
    static

Template:
    power(Exponent,Value)
Mode and number of proofs:
    power(+positive_float,-float) - one

---

weibull/3

Returns a scaled Weibull distributed random value.

Compilation flags:
    static

Template:
    weibull(Shape,Scale,Value)
Mode and number of proofs:
    weibull(+float,+positive_float,-float) - one

---

uniform/3

Returns a uniform distributed random value in the interval“[Lower, Upper[. Fails if ``Lower or Upper are not integers or if Lower > Upper. Same as random/3.

Compilation flags:
    static

Template:
    uniform(Lower,Upper,Value)
Mode and number of proofs:
    uniform(+float,+float,-float) - zero_or_one

---

uniform/1

Returns a uniform distributed random value in the interval“[0.0, 1.0[. Same as ``random/1.

Compilation flags:
    static

Template:
    uniform(Value)
Mode and number of proofs:
    uniform(-float) - one

---

triangular/4

Returns a triangular distributed random value. Fails if the Left =< Mode =< Right condition does not hold.

Compilation flags:
    static

Template:
    triangular(Left,Mode,Right,Value)
Mode and number of proofs:
    triangular(+float,+float,+float,-float) - zero_or_one

---

von_mises/3

Returns a von Mises distributed random value.

Compilation flags:
    static

Template:
    von_mises(Mode,Concentration,Value)
Mode and number of proofs:
    von_mises(+float,+non_negative_float,-float) - zero_or_one

---

gumbel/3

Returns a Gumbel distributed random value.

Compilation flags:
    static

Template:
    gumbel(Location,Scale,Value)
Mode and number of proofs:
    gumbel(+float,+non_negative_float,-float) - zero_or_one

---

dirichlet/2

Returns a Dirichlet distributed list of random values.

Compilation flags:
    static

Template:
    dirichlet(Alphas,Thetas)
Mode and number of proofs:

dirichlet(+list(positive_float),-list(positive_float)) - one

---

circular_uniform_polar/3

Returns a circular uniform distributed random point in polar coordinates given the circle radius.

Compilation flags:
static

Template:
circular_uniform_polar(Radius,Rho,Theta)
Mode and number of proofs:
circular_uniform_polar(+float,+float,-float) - one

---

circular_uniform_cartesian/3

Returns a circular uniform distributed random point in cartesian coordinates given the circle radius.

Compilation flags:
static

Template:
circular_uniform_cartesian(Radius,X,Y)
Mode and number of proofs:
circular_uniform_cartesian(+float,+float,-float) - one

---

standard_t/2

Returns a standard Student's t distributed random value given the degrees of freedom.

Compilation flags:
static

Template:

standard_t(DegreesOfFreedom,Value)
Mode and number of proofs:
standard_t(+positive_integer,-float) - one

---

standard_cauchy/3

Returns a standard Cauchy distributed random value.

Compilation flags:
static

Template:
standard_cauchy(Location,Scale,Value)
Mode and number of proofs:
standard_cauchy(+float,+float,-float) - one

---

standard_exponential/1

Returns a standard exponential distributed random value.

Compilation flags:
static

Template:
standard_exponential(Value)
Mode and number of proofs:
standard_exponential(-float) - one

---

standard_gamma/2

Returns a standard gamma distributed random value.

Compilation flags:
    static

Template:
    standard_gamma(Shape,Value)
Mode and number of proofs:
    standard_gamma(+positive_float,-float) - one

---

standard_normal/1

Returns a standard normally (Gaussian) distributed random value (using a default mean of 0.0 and a default deviation of 1.0).

Compilation flags:
    static

Template:
    standard_normal(Value)
Mode and number of proofs:
    standard_normal(-float) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

random_protocol, pseudo_random_protocol

1.187 random_forest_classifier

object

1.187.1 random_forest_classifier

Random Forest classifier using C4.5 decision trees as base learners. Builds an ensemble of decision trees trained on bootstrap samples with random feature subsets and combines their predictions through majority voting.

Availability:
logtalk_load(random_forest_classifier(loader))

Author: Paulo Moura
Version: 2:0:0
Date: 2026-05-11

Compilation flags:
static, context_switching_calls

Imports:
public probabilistic_classifier_common
Uses:
c45_classifier
fast_random(Algorithm)
format
list
pairs
type

Remarks:
(none)

Inherited public predicates:

check_classifier/1 check_option/1 check_options/1 classifier_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 predict_probabilities/3 print_classifier/1 valid_classifier/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

dataset_protocol, c45_classifier, isolation_forest_anomaly_detector, knn_classifier, naive_bayes_classifier, nearest_centroid_classifier, adaptive_boosting_classifier

1.188 random_forest_regression

object

1.188.1 random_forest_regression

Random Forest regression using regression trees as base learners trained on bootstrap samples and per-split random feature subsets.

Availability:
logtalk_load(random_forest_regression(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:
static, context_switching_calls

Imports:
public regressor_common
Uses:
fast_random(Algorithm)
format
list
population
regression_tree
type

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_regressor/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 print_regressor/1 regressor_options/2 valid_option/1 valid_options/1 valid_regressor/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

linear_regression, knn_regression, regression_tree, gradient_boosting_regression

1.189 random_projection

object

1.189.1 random_projection

Random projection reducer for continuous datasets using a portable seeded Rademacher projection matrix.

Availability:
logtalk_load(random_projection(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public dimension_reducer_common
Uses:
fast_random(Algorithm)
format

list
pairs
type

Remarks:
(none)

Inherited public predicates:
check_dimension_reducer/1 check_option/1 check_options/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 dimension_reducer_options/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 print_dimension_reducer/1 transform/3 valid_dimension_reducer/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

lda_projection, pca_projection

1.190 rank_centrality

object

1.190.1 rank_centrality

Rank Centrality pairwise preference ranker. Learns one stationary probability score per item from a dataset object implementing the pairwise_ranking_dataset_protocol protocol by applying power iteration to the Rank Centrality Markov chain built from aggregated head-to-head outcomes, and returns a self-describing ranker term with diagnostics that can be used for ranking and export.

Availability:
logtalk_load(rank_centrality(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public ranking_dataset_common
public score_ranker_common
Uses:
avltree
format
list

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_ranker/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 grouped_dataset_groups/2 grouped_dataset_items/2 grouped_dataset_items/3 grouped_dataset_summary/2 learn/2 learn/3 option/2 option/3 pairwise_dataset_connected_components/2 pairwise_dataset_declared_items/2 pairwise_dataset_items/2 pairwise_dataset_matchups/2 pairwise_dataset_preferences/2 pairwise_dataset_summary/2 pairwise_dataset_win_totals/2 pairwise_measurement_dataset_connected_components/2 pairwise_measurement_dataset_declared_items/2 pairwise_measurement_dataset_items/2 pairwise_measurement_dataset_measurements/2 pairwise_measurement_dataset_summary/2 print_ranker/1 rank/3 ranker_options/2 scores/2 temporal_pairwise_dataset_connected_components/2

temporal_pairwise_dataset_declared_items/2 temporal_pairwise_dataset_games/2
temporal_pairwise_dataset_games/3 temporal_pairwise_dataset_items/2
temporal_pairwise_dataset_periods/2 temporal_pairwise_dataset_summary/2 valid_option/1
valid_options/1 valid_ranker/1 validate_grouped_dataset/1 validate_grouped_dataset/2
validate_pairwise_dataset/1 validate_pairwise_dataset/2
validate_pairwise_measurement_dataset/1 validate_pairwise_measurement_dataset/2
validate_temporal_pairwise_dataset/1 validate_temporal_pairwise_dataset/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

pairwise_ranking_dataset_protocol, ranker_protocol, bradley_terry_ranker, copeland_ranker

1.191 ranked_pairs

object

1.191.1 ranked_pairs

Ranked Pairs pairwise preference ranker. Learns one deterministic score per item from a dataset object implementing the pairwise_ranking_dataset_protocol protocol by locking direct pairwise victories in descending strength order while avoiding cycles and returns a self-describing ranker term with diagnostics that can be used for ranking and export.

Availability:
logtalk_load(ranked_pairs(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public ranking_dataset_common
public score_ranker_model_common
public condorcet_victory_common
Uses:
avltree
list
numberlist

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_ranker/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 grouped_dataset_groups/2 grouped_dataset_items/2 grouped_dataset_items/3 grouped_dataset_summary/2 learn/2 learn/3 option/2 option/3 pairwise_dataset_connected_components/2 pairwise_dataset_declared_items/2 pairwise_dataset_items/2 pairwise_dataset_matchups/2 pairwise_dataset_preferences/2 pairwise_dataset_summary/2 pairwise_dataset_win_totals/2 pairwise_measurement_dataset_connected_components/2 pairwise_measurement_dataset_declared_items/2 pairwise_measurement_dataset_items/2 pairwise_measurement_dataset_measurements/2 pairwise_measurement_dataset_summary/2 print_ranker/1 rank/3 ranker_options/2 scores/2 temporal_pairwise_dataset_connected_components/2 temporal_pairwise_dataset_declared_items/2 temporal_pairwise_dataset_games/2 temporal_pairwise_dataset_games/3 temporal_pairwise_dataset_items/2 temporal_pairwise_dataset_periods/2 temporal_pairwise_dataset_summary/2 valid_option/1 valid_options/1 valid_ranker/1 validate_grouped_dataset/1 validate_grouped_dataset/2 validate_pairwise_dataset/1 validate_pairwise_dataset/2

validate_pairwise_measurement_dataset/1 validate_pairwise_measurement_dataset/2
validate_temporal_pairwise_dataset/1 validate_temporal_pairwise_dataset/2

- Public predicates
  - locked_pairs/2
- Protected predicates
- Private predicates
- Operators

Public predicates

locked_pairs/2

Returns the accepted lock graph in lock order as lock(Item1,Item2,Strength) terms for the learned items.

Compilation flags:
static

Template:
locked_pairs(Ranker,LockedPairs)
Mode and number of proofs:
locked_pairs(+compound,-list(compound)) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

pairwise_ranking_dataset_protocol, ranker_protocol, schulze_ranker, copeland_ranker

1.192 ranking_protocols

category

1.192.1 condorcet_victory_common

Shared direct-victory preprocessing helpers for Condorcet-family rankers.

Availability:
logtalk_load(ranking_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-26

Compilation flags:
static

Uses:
avltree
list

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
  - index_items/3

    - build_direct_strengths/5
    - zero_matrix/2
    - matrix_entry/4
    - set_matrix_entry/5
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

index_items/3

Builds Item-Index pairs for the ordered training items.

Compilation flags:
    static

Template:
    index_items(Items,Index,Indices)
Mode and number of proofs:
    index_items(+list,+integer,-list) - one

---

build_direct_strengths/5

Builds the dense direct-victory strength matrix from aggregated pairwise matchups using the selected victory-strength semantics.

Compilation flags:
    static

Template:
    build_direct_strengths(Matchups,IndexDictionary,Count,VictoryStrength,DirectStrengths)
Mode and number of proofs:
    build_direct_strengths(+list,+compound,+integer,+atom,-list) - one

---

zero_matrix/2

Builds a square zero matrix with the given row and column count.

Compilation flags:
    static

Template:
    zero_matrix(Count,Matrix)
Mode and number of proofs:
    zero_matrix(+integer,-list) - one

---

matrix_entry/4

Returns the value stored at the given matrix row and column.

Compilation flags:
    static

Template:
    matrix_entry(Matrix,RowIndex,ColumnIndex,Value)
Mode and number of proofs:
    matrix_entry(+list,+integer,+integer,-integer) - one

---

set_matrix_entry/5

Returns an updated matrix where the given row and column entry was replaced by the supplied value.

Compilation flags:
    static

Template:
    set_matrix_entry(Matrix,RowIndex,ColumnIndex,Value,UpdatedMatrix)
Mode and number of proofs:

set_matrix_entry(+list,+integer,+integer,+integer,-list) - one

---

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.192.2 glicko2_common

Shared internal Glicko-2 numeric helpers for rating updates, volatility search, and scale conversions.

Availability:
logtalk_load(ranking_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-02

Compilation flags:
static

Uses:
avltree

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
- Protected predicates

  - update_item_parameters/10
  - to_mu/2
  - to_phi/2
  - to_rating/2
  - to_deviation/2
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

update_item_parameters/10

Computes the simultaneous Glicko-2 rating, deviation, and volatility update for one item over a rating period using the current dictionaries and aggregated game results.

Compilation flags:
static

Template:
update_item_parameters(Item,ItemResults,Ratings,Deviations,Volatilities,Tau,VolatilityTolerance,UpdatedRating,UpdatedDeviation,UpdatedVolatility)

Mode and number of proofs:
update_item_parameters(+term,+list(compound),+compound,+compound,+compound,+number,+number,-number,-number,-number) - one

---

to_mu/2

Converts a rating on the external Glicko scale to the internal mu scale.

Compilation flags:
static

Template:
to_mu(Rating,Mu)

Mode and number of proofs:
to_mu(+number,-number) - one

---

to_phi/2

Converts a rating deviation on the external Glicko scale to the internal phi scale.

Compilation flags:
static

Template:
to_phi(Deviation,Phi)
Mode and number of proofs:
to_phi(+number,-number) - one

---

to_rating/2

Converts a value on the internal mu scale back to the external Glicko rating scale.

Compilation flags:
static

Template:
to_rating(Mu,Rating)
Mode and number of proofs:
to_rating(+number,-number) - one

---

to_deviation/2

Converts a value on the internal phi scale back to the external Glicko rating-deviation scale.

Compilation flags:
static

Template:
to_deviation(Phi,Deviation)
Mode and number of proofs:
to_deviation(+number,-number) - one

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.192.3 grouped_strength_ranker_common

Shared connectivity, positive-strength, and iterative-update helpers for grouped ranking models that learn one strength parameter per item.

Availability:
logtalk_load(ranking_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-25

Compilation flags:
static

Uses:
avltree
list

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
  - update_strengths/4
  - initial_strengths/2
  - strength_pairs/3
  - optimize_strengths/8
  - strength_dictionary/2
  - require_strong_connectivity/4
  - strongly_connected/3
  - strongly_connected_components/4
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

update_strengths/4

Hook predicate that importing rankers must define to perform one synchronous strength-update iteration.

Compilation flags:
    static

Template:
    update_strengths(Context,Strengths0,Strengths,MaximumDifference)
Mode and number of proofs:
    update_strengths(+nonvar,+list(number),-list(number),-number) - one

initial_strengths/2

Initializes a positive strength vector with a uniform value for each ordered item.

Compilation flags:
    static

Template:
    initial_strengths(Items,Strengths)
Mode and number of proofs:
    initial_strengths(+list,-list(float)) - one

---

strength_pairs/3

Pairs ordered items with their learned strength values.

Compilation flags:
    static

Template:
    strength_pairs(Items,StrengthValues,Strengths)
Mode and number of proofs:
    strength_pairs(+list,+list(number),-list(pair)) - one

---

optimize_strengths/8

Runs the iterative strength-update loop until convergence or the maximum iteration bound is reached.

Compilation flags:
    static

Template:
    optimize_strengths(MaximumIterations,Tolerance,UpdateContext,Strengths0,Strengths,Status,
    Iterations,FinalDifference)
Mode and number of proofs:
    optimize_strengths(+integer,+number,+nonvar,+list(number),-list(number),-atom,-integer,-number)
    - one

---

strength_dictionary/2

Builds an index-addressable dictionary from an ordered list of strength values.

Compilation flags:
    static

Template:
    strength_dictionary(Strengths,Dictionary)
Mode and number of proofs:
    strength_dictionary(+list(number),-compound) - one

---

require_strong_connectivity/4

Checks that the directed graph is strongly connected and throws a domain error with the strongly connected components otherwise.

Compilation flags:
    static

Template:
    require_strong_connectivity(Count,DirectedAdjacency,ReverseAdjacency,ErrorDomain)
Mode and number of proofs:
    require_strong_connectivity(+integer,+compound,+compound,+atom) - one

---

strongly_connected/3

True when all indexed items are mutually reachable in both the forward and reverse adjacency graphs.

Compilation flags:
    static

Template:
    strongly_connected(Count,DirectedAdjacency,ReverseAdjacency)

Mode and number of proofs:
strongly_connected(+integer,+compound,+compound) - zero_or_one

---

strongly_connected_components/4

Computes the strongly connected components of the directed graph induced by the forward and reverse adjacency dictionaries.

Compilation flags:
static

Template:
strongly_connected_components(Count,DirectedAdjacency,ReverseAdjacency,Components)
Mode and number of proofs:
strongly_connected_components(+integer,+compound,+compound,-list) - one

---

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.192.4 pairwise_measurement_dataset_protocol

Protocol for pairwise measurement datasets.

Availability:
logtalk_load(ranking_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:

static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - item/1
  - measurement/4
- Protected predicates
- Private predicates
- Operators

Public predicates

item/1

Enumerates by backtracking the items that can be ranked.

Compilation flags:
static

Template:
item(Item)
Mode and number of proofs:
item(-atom) - zero_or_more

measurement/4

Enumerates by backtracking weighted pairwise signed measurements where Value is the observed scalar measurement for the oriented edge Item1 -> Item2 and Weight is its positive confidence or repetition weight.

Compilation flags:
    static

Template:
    measurement(Item1,Item2,Value,Weight)
Mode and number of proofs:
    measurement(-atom,-atom,-number,-number) - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

protocol

1.192.5 pairwise_ranking_dataset_protocol

Protocol for pairwise ranking datasets.

Availability:
    logtalk_load(ranking_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:
    static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - item/1
  - preference/3
- Protected predicates
- Private predicates
- Operators

Public predicates

item/1

Enumerates by backtracking the items that can be ranked.

Compilation flags:
static

Template:
item(Item)
Mode and number of proofs:
item(-atom) - zero_or_more

preference/3

Enumerates by backtracking weighted pairwise preferences where Winner is preferred over Loser Weight times.

Compilation flags:
    static

Template:
    preference(Winner,Loser,Weight)
Mode and number of proofs:
    preference(-atom,-atom,-number) - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

category

1.192.6 pairwise_strength_ranker_common

Shared preprocessing, connectivity, and MM-iteration helpers for pairwise strength rankers.

Availability:
    logtalk_load(ranking_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-25

Compilation flags:
    static

Uses:

avltree
list

Remarks:

(none)

Inherited public predicates:

(none)

- Public predicates
- Protected predicates
  - update_strengths/4
  - initial_strengths/2
  - strength_pairs/3
  - index_items/3
  - preprocess_matchups/6
  - build_dataset_model/5
  - require_strong_connectivity/4
  - strongly_connected/3
  - strongly_connected_components/4
  - optimize_strengths/8
  - strength_dictionary/2
  - item_denominator/5
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

update_strengths/4

Hook predicate that importing rankers must define to perform one synchronous strength-update iteration.

Compilation flags:
    static

Template:
    update_strengths(Context,Strengths0,Strengths,MaximumDifference)
Mode and number of proofs:
    update_strengths(+nonvar,+list(number),-list(number),-number) - one

---

initial_strengths/2

Initializes a positive strength vector with a uniform value for each ordered item.

Compilation flags:
    static

Template:
    initial_strengths(Items,Strengths)
Mode and number of proofs:
    initial_strengths(+list,-list(float)) - one

---

strength_pairs/3

Pairs ordered items with their learned strength values.

Compilation flags:
    static

Template:
strength_pairs(Items,StrengthValues,Strengths)
Mode and number of proofs:
strength_pairs(+list,+list(number),-list(pair)) - one

---

index_items/3

Builds Item-Index pairs for the ordered training items.

Compilation flags:
static

Template:
index_items(Items,Index,Indices)
Mode and number of proofs:
index_items(+list,+integer,-list(pair)) - one

---

preprocess_matchups/6

Builds directed adjacency, reverse adjacency, per-item win totals, and undirected pair-adjacency structures from aggregated pairwise matchups.

Compilation flags:
static

Template:
preprocess_matchups(Matchups,IndexDictionary,DirectedAdjacency,ReverseAdjacency,WinWeights,PairAdjacency)
Mode and number of proofs:
preprocess_matchups(+list,+compound,-compound,-compound,-compound,-compound) - one

---

build_dataset_model/5

Materializes dense per-item win totals and weighted undirected adjacency lists from the sparse preprocessing dictionaries.

Compilation flags:
    static

Template:
    build_dataset_model(Count,WinWeights,PairAdjacency,PairWeights,Wins)
Mode and number of proofs:
    build_dataset_model(+integer,+compound,+compound,-list,-list(number)) - one

---

require_strong_connectivity/4

Checks that the directed graph is strongly connected and throws a domain error with the strongly connected components otherwise.

Compilation flags:
    static

Template:
    require_strong_connectivity(Count,DirectedAdjacency,ReverseAdjacency,ErrorDomain)
Mode and number of proofs:
    require_strong_connectivity(+integer,+compound,+compound,+atom) - one

---

strongly_connected/3

True when all indexed items are mutually reachable in both the forward and reverse adjacency graphs.

Compilation flags:
    static

Template:
    strongly_connected(Count,DirectedAdjacency,ReverseAdjacency)
Mode and number of proofs:
    strongly_connected(+integer,+compound,+compound) - zero_or_one

---

strongly_connected_components/4

Computes the strongly connected components of the directed graph induced by the forward and reverse adjacency dictionaries.

Compilation flags:
    static

Template:
    strongly_connected_components(Count,DirectedAdjacency,ReverseAdjacency,Components)
Mode and number of proofs:
    strongly_connected_components(+integer,+compound,+compound,-list) - one

---

optimize_strengths/8

Runs the iterative strength-update loop until convergence or the maximum iteration bound is reached.

Compilation flags:
    static

Template:
    optimize_strengths(MaximumIterations,Tolerance,UpdateContext,Strengths0,Strengths,Status,
    Iterations,FinalDifference)
Mode and number of proofs:
    optimize_strengths(+integer,+number,+nonvar,+list(number),-list(number),-atom,-integer,-number)
    - one

---

strength_dictionary/2

Builds an index-addressable dictionary from an ordered list of strength values.

Compilation flags:
    static

Template:

strength_dictionary(Strengths,Dictionary)
Mode and number of proofs:
strength_dictionary(+list(number),-compound) - one

---

item_denominator/5

Accumulates the pairwise MM denominator for one item using the current strength dictionary and weighted neighbors.

Compilation flags:
static

Template:
item_denominator(Neighbors,StrengthDictionary,CurrentStrength,Denominator0,Denominator)
Mode and number of proofs:
item_denominator(+list(pair),+compound,+number,+number,-number) - one

---

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.192.7 plackett_luce_common

Shared helpers for grouped Plackett-Luce rankers.

Availability:
logtalk_load(ranking_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-26

Compilation flags:
static

Uses:
avltree
list

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
  - index_items/3
  - preprocess_groups/10
  - selection_exposures/4
  - normalize_strengths/6
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

index_items/3

Builds Item-Index pairs for the ordered training items.

Compilation flags:
static

Template:
index_items(Items,Index,Indices)
Mode and number of proofs:
index_items(+list,+integer,-list) - one

---

preprocess_groups/10

Builds per-group choice steps, selection counts, and forward and reverse adjacency dictionaries from grouped rankings under the selected traversal order.

Compilation flags:
    static

Template:
    preprocess_groups(Order,Groups,Dataset,MissingRelevance,IndexDictionary,Count,Steps,
    SelectionCounts,DirectedAdjacency,ReverseAdjacency)
Mode and number of proofs:
    preprocess_groups(+atom,+list,+object_identifier,+atom,+compound,+integer,-list,-list,
    -compound,-compound) - one

---

selection_exposures/4

Computes the per-item exposure counts implied by the grouped choice steps and the current value dictionary.

Compilation flags:
    static

Template:
    selection_exposures(Steps,SelectionCounts,ValueDictionary,ExposureCounts)
Mode and number of proofs:
    selection_exposures(+list,+list,+compound,-list) - one

---

normalize_strengths/6

Normalizes raw strength values to sum to one while tracking the maximum absolute change from the current strengths.

Compilation flags:
    static

Template:
normalize_strengths(RawStrengths,TotalRawStrength,CurrentStrengths,Strengths,
MaximumDifference0,MaximumDifference)
Mode and number of proofs:
normalize_strengths(+list(number),+number,+list(number),-list(number),+number,-number) - one

---

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.192.8 ranker_common

Shared predicates for ranker score access, diagnostics, and export.

Availability:
logtalk_load(ranking_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-30

Compilation flags:
static

Implements:
public ranker_protocol
Extends:
public options
Uses:
avltree
format
list
type

Remarks:

(none)

Inherited public predicates:
check_option/1 check_options/1 check_ranker/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 print_ranker/1 rank/3 ranker_options/2 scores/2 valid_option/1 valid_options/1 valid_ranker/1

- Public predicates
- Protected predicates
  - ranker_scores_data/2
  - ranker_diagnostics_data/2
  - ranker_export_template/4
  - ranker_term_template/2
  - print_ranker_template/1
  - valid_ranker_metadata/2
  - valid_item_value_pairs/2
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

ranker_scores_data/2

Hook predicate that importing ranker implementations must define in order to expose the learned item-score pairs on the ranker native numeric scale.

Compilation flags:
static

Template:
ranker_scores_data(Ranker,Scores)
Mode and number of proofs:
ranker_scores_data(+compound,-list(pair)) - one

---

ranker_diagnostics_data/2

Hook predicate that importing ranker implementations must define in order to expose diagnostics metadata.

Compilation flags:
    static

Template:
    ranker_diagnostics_data(Ranker,Diagnostics)
Mode and number of proofs:
    ranker_diagnostics_data(+compound,-list(compound)) - one

---

ranker_export_template/4

Hook predicate that importing ranker implementations must define in order to expose the exported ranker template for a given functor.

Compilation flags:
    static

Template:
    ranker_export_template(Dataset,Ranker,Functor,Template)
Mode and number of proofs:
    ranker_export_template(+object_identifier,+compound,+atom,-callable) - one

---

ranker_term_template/2

Hook predicate that importing ranker implementations must define in order to expose the learned ranker term template used by pretty-printing helpers.

Compilation flags:
    static

Template:

ranker_term_template(Ranker,Template)
Mode and number of proofs:
ranker_term_template(+compound,-callable) - one

---

print_ranker_template/1

Pretty-printing helper predicate used by importing ranker implementations to show the learned ranker term template.

Compilation flags:
static

Template:
print_ranker_template(Ranker)
Mode and number of proofs:
print_ranker_template(+compound) - one

---

valid_ranker_metadata/2

True when the diagnostics list contains the expected model identifier, a valid effective options list, and a structurally valid dataset summary.

Compilation flags:
static

Template:
valid_ranker_metadata(Model,Diagnostics)
Mode and number of proofs:
valid_ranker_metadata(+atom,+list(compound)) - zero_or_one

---

valid_item_value_pairs/2

True when the second argument is an ordered list of unique Item-Value pairs aligned with the first argument item list.

Compilation flags:
    static

Template:
    valid_item_value_pairs(Items,Pairs)
Mode and number of proofs:
    valid_item_value_pairs(+list,+list(pair)) - zero_or_one

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.192.9 ranker_protocol

Protocol for machine learning rankers.

Availability:
    logtalk_load(ranking_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
    static

Dependencies:
    (none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - learn/3
  - learn/2
  - rank/3
  - scores/2
  - diagnostics/2
  - diagnostic/2
  - ranker_options/2
  - check_ranker/1
  - valid_ranker/1
  - export_to_clauses/4
  - export_to_file/4
  - print_ranker/1
- Protected predicates
- Private predicates
- Operators

Public predicates

learn/3

Learns a ranker from the given dataset object using the specified options.

Compilation flags:
static

Template:
learn(Dataset,Ranker,Options)
Mode and number of proofs:
learn(+object_identifier,-compound,+list(compound)) - one

learn/2

Learns a ranker from the given dataset object using default options.

Compilation flags:
    static

Template:
    learn(Dataset,Ranker)
Mode and number of proofs:
    learn(+object_identifier,-compound) - one

---

rank/3

Ranks a list of candidate items using the learned ranker and returns the candidates ordered from most preferred to least preferred.

Compilation flags:
    static

Template:
    rank(Ranker,Candidates,Ranking)
Mode and number of proofs:
    rank(+compound,+list,-list) - one

---

scores/2

Returns the learned item-score pairs on the ranker native numeric scale. Depending on the ranker, these values may also be interpreted more specifically as strengths or ratings.

Compilation flags:
    static

Template:
    scores(Ranker,Scores)
Mode and number of proofs:
    scores(+compound,-list(pair)) - one

---

diagnostics/2

Returns diagnostics and metadata associated with a learned ranker in a representation-independent way.

Compilation flags:
    static

Template:
    diagnostics(Ranker,Diagnostics)
Mode and number of proofs:
    diagnostics(+compound,-list(compound)) - one

---

diagnostic/2

Tests or enumerates individual diagnostics metadata terms for a learned ranker.

Compilation flags:
    static

Template:
    diagnostic(Ranker,Diagnostic)
Mode and number of proofs:
    diagnostic(+compound,?compound) - zero_or_more

---

ranker_options/2

Returns the effective training options recorded in a learned ranker diagnostics list.

Compilation flags:
    static

Template:
    ranker_options(Ranker,Options)
Mode and number of proofs:

ranker_options(+compound,-list(compound)) - zero_or_one

---

check_ranker/1

Checks that the argument is a structurally valid learned ranker term for the receiving ranker implementation, throwing an exception on invalid input when applicable.

Compilation flags:
static

Template:
check_ranker(Ranker)
Mode and number of proofs:
check_ranker(@compound) - one_or_error

Exceptions:
Ranker is a variable:
instantiation_error
Ranker is neither a variable nor a valid ranker:
domain_error(ranker,Ranker)

---

valid_ranker/1

True when check_ranker/1 succeeds for the argument without throwing an exception.

Compilation flags:
static

Template:
valid_ranker(Ranker)
Mode and number of proofs:
valid_ranker(@compound) - zero_or_one

---

export_to_clauses/4

Converts a ranker into a list of predicate clauses. Functor is the functor for the generated predicate clauses. When exporting a serialized ranker term, a noun such as ranker or model is usually clearer than a verb or result label.

Compilation flags:
    static

Template:
    export_to_clauses(Dataset,Ranker,Functor,Clauses)
Mode and number of proofs:
    export_to_clauses(+object_identifier,+compound,+callable,-list(clause)) - one

---

export_to_file/4

Exports a ranker to a file. Functor is the functor for the generated predicate clauses. When exporting a serialized ranker term, a noun such as ranker or model is usually clearer than a verb or result label.

Compilation flags:
    static

Template:
    export_to_file(Dataset,Ranker,Functor,File)
Mode and number of proofs:
    export_to_file(+object_identifier,+compound,+callable,+atom) - one

---

print_ranker/1

Prints a ranker to the current output stream in a human-readable format.

Compilation flags:
    static

Template:
    print_ranker(Ranker)
Mode and number of proofs:

print_ranker(+compound) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

bradley_terry_ranker, ranking_dataset_protocol, pairwise_ranking_dataset_protocol

category

1.192.10 ranking_dataset_common

Shared predicates for collecting, analyzing, and validating ranking datasets.

Availability:
logtalk_load(ranking_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-18

Compilation flags:
static

Uses:
avltree
list
numberlist

Remarks:

(none)

Inherited public predicates:

(none)

- Public predicates
  - pairwise_dataset_declared_items/2
  - pairwise_dataset_items/2
  - pairwise_dataset_preferences/2
  - pairwise_dataset_connected_components/2
  - pairwise_dataset_summary/2
  - pairwise_dataset_win_totals/2
  - pairwise_dataset_matchups/2
  - pairwise_measurement_dataset_declared_items/2
  - pairwise_measurement_dataset_items/2
  - pairwise_measurement_dataset_measurements/2
  - pairwise_measurement_dataset_connected_components/2
  - pairwise_measurement_dataset_summary/2
  - temporal_pairwise_dataset_declared_items/2
  - temporal_pairwise_dataset_items/2
  - temporal_pairwise_dataset_periods/2
  - temporal_pairwise_dataset_games/2
  - temporal_pairwise_dataset_games/3
  - temporal_pairwise_dataset_connected_components/2
  - temporal_pairwise_dataset_summary/2
  - grouped_dataset_groups/2
  - grouped_dataset_items/2
  - grouped_dataset_items/3
  - grouped_dataset_summary/2
  - validate_pairwise_dataset/1
  - validate_pairwise_dataset/2
  - validate_pairwise_measurement_dataset/1
  - validate_pairwise_measurement_dataset/2
  - validate_grouped_dataset/1
  - validate_temporal_pairwise_dataset/1

  - validate_temporal_pairwise_dataset/2
  - validate_grouped_dataset/2
- Protected predicates
  - grouped_dataset_item_relevances/4
  - grouped_dataset_relevance_frequencies/2
  - grouped_dataset_item_relevance_scores/3
  - grouped_dataset_tie_blocks/4
- Private predicates
- Operators

Public predicates

pairwise_dataset_declared_items/2

Collects the pairwise dataset declared items preserving declaration order.

Compilation flags:
    static

Template:
    pairwise_dataset_declared_items(Dataset,Items)
Mode and number of proofs:
    pairwise_dataset_declared_items(+object_identifier,-list) - one

---

pairwise_dataset_items/2

Collects the unique pairwise dataset items preserving their first declaration order.

Compilation flags:
    static

Template:
    pairwise_dataset_items(Dataset,Items)
Mode and number of proofs:
    pairwise_dataset_items(+object_identifier,-list) - one

---

pairwise_dataset_preferences/2

Collects the pairwise dataset preferences as p(Winner,Loser,Weight) terms.

Compilation flags:
    static

Template:
    pairwise_dataset_preferences(Dataset,Preferences)
Mode and number of proofs:
    pairwise_dataset_preferences(+object_identifier,-list) - one

---

pairwise_dataset_connected_components/2

Returns the connected components induced by the pairwise preference graph.

Compilation flags:
    static

Template:
    pairwise_dataset_connected_components(Dataset,Components)
Mode and number of proofs:
    pairwise_dataset_connected_components(+object_identifier,-list(list)) - one

---

pairwise_dataset_summary/2

Returns a summary of a pairwise dataset, including item, preference, component, and isolated-item counts.

Compilation flags:
    static

Template:
    pairwise_dataset_summary(Dataset,Summary)
Mode and number of proofs:
    pairwise_dataset_summary(+object_identifier,-list(compound)) - one

---

pairwise_dataset_win_totals/2

Returns per-item total win weights for a pairwise dataset.

Compilation flags:
    static

Template:
    pairwise_dataset_win_totals(Dataset,Totals)
Mode and number of proofs:
    pairwise_dataset_win_totals(+object_identifier,-list(pair)) - one

---

pairwise_dataset_matchups/2

Returns aggregated pairwise matchups as matchup(Item1,Item2,Item1Wins,Item2Wins) terms preserving item declaration order inside each pair.

Compilation flags:
    static

Template:
    pairwise_dataset_matchups(Dataset,Matchups)
Mode and number of proofs:
    pairwise_dataset_matchups(+object_identifier,-list(compound)) - one

---

pairwise_measurement_dataset_declared_items/2

Collects the pairwise measurement dataset declared items preserving declaration order.

Compilation flags:
    static

Template:
    pairwise_measurement_dataset_declared_items(Dataset,Items)
Mode and number of proofs:
    pairwise_measurement_dataset_declared_items(+object_identifier,-list) - one

---

pairwise_measurement_dataset_items/2

Collects the unique pairwise measurement dataset items preserving their first declaration order.

Compilation flags:
    static

Template:
    pairwise_measurement_dataset_items(Dataset,Items)
Mode and number of proofs:
    pairwise_measurement_dataset_items(+object_identifier,-list) - one

---

pairwise_measurement_dataset_measurements/2

Collects the pairwise measurement dataset observations as m(Item1,Item2,Value,Weight) terms preserving enumeration order.

Compilation flags:
    static

Template:
    pairwise_measurement_dataset_measurements(Dataset,Measurements)
Mode and number of proofs:
    pairwise_measurement_dataset_measurements(+object_identifier,-list(compound)) - one

---

pairwise_measurement_dataset_connected_components/2

Returns the connected components induced by the pairwise measurement support graph.

Compilation flags:
    static

Template:
    pairwise_measurement_dataset_connected_components(Dataset,Components)

Mode and number of proofs:
pairwise_measurement_dataset_connected_components(+object_identifier,-list(list)) - one

---

pairwise_measurement_dataset_summary/2

Returns a summary of a pairwise measurement dataset, including item, measurement, component, and isolated-item counts.

Compilation flags:
static

Template:
pairwise_measurement_dataset_summary(Dataset,Summary)
Mode and number of proofs:
pairwise_measurement_dataset_summary(+object_identifier,-list(compound)) - one

---

temporal_pairwise_dataset_declared_items/2

Collects the temporal pairwise dataset declared items preserving declaration order.

Compilation flags:
static

Template:
temporal_pairwise_dataset_declared_items(Dataset,Items)
Mode and number of proofs:
temporal_pairwise_dataset_declared_items(+object_identifier,-list) - one

---

temporal_pairwise_dataset_items/2

Collects the unique temporal pairwise dataset items preserving their first declaration order.

Compilation flags:
    static

Template:
    temporal_pairwise_dataset_items(Dataset,Items)
Mode and number of proofs:
    temporal_pairwise_dataset_items(+object_identifier,-list) - one

---

temporal_pairwise_dataset_periods/2

Collects the unique temporal pairwise dataset periods preserving their first declaration order.

Compilation flags:
    static

Template:
    temporal_pairwise_dataset_periods(Dataset,Periods)
Mode and number of proofs:
    temporal_pairwise_dataset_periods(+object_identifier,-list) - one

---

temporal_pairwise_dataset_games/2

Collects the temporal pairwise dataset games as game(Period,Item1,Item2,Score) terms preserving enumeration order.

Compilation flags:
    static

Template:
    temporal_pairwise_dataset_games(Dataset,Games)
Mode and number of proofs:
    temporal_pairwise_dataset_games(+object_identifier,-list(compound)) - one

---

temporal_pairwise_dataset_games/3

Collects the temporal pairwise dataset games for a single period as game(Item1,Item2,Score) terms preserving enumeration order.

Compilation flags:
    static

Template:
    temporal_pairwise_dataset_games(Dataset,Period,Games)
Mode and number of proofs:
    temporal_pairwise_dataset_games(+object_identifier,+term,-list(compound)) - one

---

temporal_pairwise_dataset_connected_components/2

Returns the connected components induced by the temporal pairwise game graph.

Compilation flags:
    static

Template:
    temporal_pairwise_dataset_connected_components(Dataset,Components)
Mode and number of proofs:
    temporal_pairwise_dataset_connected_components(+object_identifier,-list(list)) - one

---

temporal_pairwise_dataset_summary/2

Returns a summary of a temporal pairwise dataset, including item, period, game, component, and isolated-item counts.

Compilation flags:
    static

Template:

temporal_pairwise_dataset_summary(Dataset,Summary)
Mode and number of proofs:
temporal_pairwise_dataset_summary(+object_identifier,-list(compound)) - one

---

grouped_dataset_groups/2

Collects the unique grouped-dataset groups preserving their first declaration order.

Compilation flags:
static

Template:
grouped_dataset_groups(Dataset,Groups)
Mode and number of proofs:
grouped_dataset_groups(+object_identifier,-list) - one

---

grouped_dataset_items/2

Collects the unique items declared across all groups in a grouped dataset preserving their first declaration order.

Compilation flags:
static

Template:
grouped_dataset_items(Dataset,Items)
Mode and number of proofs:
grouped_dataset_items(+object_identifier,-list) - one

---

grouped_dataset_items/3

Collects the unique items declared for a grouped-dataset group.

Compilation flags:
    static

Template:
    grouped_dataset_items(Dataset,Group,Items)
Mode and number of proofs:
    grouped_dataset_items(+object_identifier,+term,-list) - one

---

grouped_dataset_summary/2

Returns a summary of a grouped dataset, including group, item, and relevance-judgment counts.

Compilation flags:
    static

Template:
    grouped_dataset_summary(Dataset,Summary)
Mode and number of proofs:
    grouped_dataset_summary(+object_identifier,-list(compound)) - one

---

validate_pairwise_dataset/1

Validates a pairwise ranking dataset and throws an error if the dataset is malformed or disconnected.

Compilation flags:
    static

Template:
    validate_pairwise_dataset(Dataset)
Mode and number of proofs:
    validate_pairwise_dataset(+object_identifier) - one

---

validate_pairwise_dataset/2

Validates a pairwise ranking dataset and returns its dataset summary when validation succeeds.

Compilation flags:
    static

Template:
    validate_pairwise_dataset(Dataset,Summary)
Mode and number of proofs:
    validate_pairwise_dataset(+object_identifier,-list(compound)) - one

---

validate_pairwise_measurement_dataset/1

Validates a pairwise measurement dataset and throws an error if the dataset is malformed or disconnected.

Compilation flags:
    static

Template:
    validate_pairwise_measurement_dataset(Dataset)
Mode and number of proofs:
    validate_pairwise_measurement_dataset(+object_identifier) - one

---

validate_pairwise_measurement_dataset/2

Validates a pairwise measurement dataset and returns its dataset summary when validation succeeds.

Compilation flags:
    static

Template:
    validate_pairwise_measurement_dataset(Dataset,Summary)
Mode and number of proofs:
    validate_pairwise_measurement_dataset(+object_identifier,-list(compound)) - one

---

validate_grouped_dataset/1

Validates a grouped ranking dataset and throws an error if the dataset is malformed.

Compilation flags:
    static

Template:
    validate_grouped_dataset(Dataset)
Mode and number of proofs:
    validate_grouped_dataset(+object_identifier) - one

---

validate_temporal_pairwise_dataset/1

Validates a temporal pairwise ranking dataset and throws an error if the dataset is malformed or disconnected.

Compilation flags:
    static

Template:
    validate_temporal_pairwise_dataset(Dataset)
Mode and number of proofs:
    validate_temporal_pairwise_dataset(+object_identifier) - one

---

validate_temporal_pairwise_dataset/2

Validates a temporal pairwise ranking dataset and returns its dataset summary when validation succeeds.

Compilation flags:
    static

Template:
    validate_temporal_pairwise_dataset(Dataset,Summary)
Mode and number of proofs:
    validate_temporal_pairwise_dataset(+object_identifier,-list(compound)) - one

---

validate_grouped_dataset/2

Validates a grouped ranking dataset and returns its dataset summary when validation succeeds.

Compilation flags:
    static

Template:
    validate_grouped_dataset(Dataset,Summary)
Mode and number of proofs:
    validate_grouped_dataset(+object_identifier,-list(compound)) - one

---

Protected predicates

grouped_dataset_item_relevances/4

Collects the declared items for a grouped-dataset group paired with their relevance values under the given missing-relevance policy.

Compilation flags:
    static

Template:
    grouped_dataset_item_relevances(Dataset,Group,MissingRelevance,ItemRelevances)
Mode and number of proofs:
    grouped_dataset_item_relevances(+object_identifier,+term,+atom,-list(pair)) - one

---

grouped_dataset_relevance_frequencies/2

Aggregates grouped item relevances into a relevance-frequency dictionary.

Compilation flags:
    static

Template:
    grouped_dataset_relevance_frequencies(ItemRelevances,FrequencyDictionary)
Mode and number of proofs:

grouped_dataset_relevance_frequencies(+list(pair),-compound) - one

---

grouped_dataset_item_relevance_scores/3

Maps grouped item relevances to per-item score deltas using a relevance-points dictionary.

Compilation flags:
static

Template:
grouped_dataset_item_relevance_scores(ItemRelevances,PointsDictionary,ItemScores)
Mode and number of proofs:
grouped_dataset_item_relevance_scores(+list(pair),+compound,-list(pair)) - one

---

grouped_dataset_tie_blocks/4

Collects grouped-dataset item relevances into descending relevance tie blocks represented as tie_block(Relevance, Items) terms, preserving the declaration order of items inside each block.

Compilation flags:
static

Template:
grouped_dataset_tie_blocks(Dataset,Group,MissingRelevance,TieBlocks)
Mode and number of proofs:
grouped_dataset_tie_blocks(+object_identifier,+term,+atom,-list(compound)) - one

---

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.192.11 ranking_dataset_protocol

Protocol for grouped ranking datasets.

Availability:
logtalk_load(ranking_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-16

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - group/1
  - item/2
  - relevance/3
- Protected predicates
- Private predicates
- Operators

Public predicates

group/1

Enumerates by backtracking the ranking groups or contexts in the dataset.

Compilation flags:
    static

Template:
    group(Group)
Mode and number of proofs:
    group(-atom) - zero_or_more

---

item/2

Enumerates by backtracking the items available inside each ranking group or context.

Compilation flags:
    static

Template:
    item(Group,Item)
Mode and number of proofs:
    item(+atom,-atom) - zero_or_more
    item(-atom,-atom) - zero_or_more

---

relevance/3

Enumerates by backtracking the non-negative integer relevance judgments associated with each group and item.

Compilation flags:
    static

Template:
    relevance(Group,Item,Relevance)

Mode and number of proofs:
relevance(+atom,+atom,-integer) - zero_or_more
relevance(-atom,-atom,-integer) - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

category

1.192.12 score_ranker_common

Shared predicates for score-based rankers that order candidates using precomputed item scores.

Availability:
logtalk_load(ranking_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-25

Compilation flags:
static

Extends:
public ranker_common
Uses:
avltree
list
pairs

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_ranker/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 print_ranker/1 rank/3 ranker_options/2 scores/2 valid_option/1 valid_options/1 valid_ranker/1

- Public predicates
- Protected predicates
  - rank_by_scores/3
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

rank_by_scores/3

Ranks candidates using precomputed Item-Score pairs while validating item existence and duplicate candidates.

Compilation flags:
static

Template:
rank_by_scores(Scores,Candidates,Ranking)
Mode and number of proofs:
rank_by_scores(+list(pair),+list,-list) - one

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.192.13 score_ranker_model_common

Shared predicates for score-ranker term construction, validation, ranking, export, and pretty-printing.

Availability:
logtalk_load(ranking_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-27

Compilation flags:
static

Extends:
public score_ranker_common
Uses:
avltree
format
list

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_ranker/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 print_ranker/1 rank/3 ranker_options/2 scores/2 valid_option/1 valid_options/1 valid_ranker/1

- Public predicates
- Protected predicates
  - score_ranker_model/1
  - score_ranker_label/1
  - score_ranker_term/4
  - valid_score/1
  - build_score_ranker/5
  - score_ranker_data/4
  - valid_score_ranker_diagnostics/3
  - initialize_scores/2
  - accumulate_item_scores/3
  - update_score/4
  - ordered_scores/3
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

score_ranker_model/1

Hook predicate that importing rankers must define to identify the learned score-ranker model.

Compilation flags:
    static

Template:
    score_ranker_model(Model)
Mode and number of proofs:
    score_ranker_model(-atom) - one

score_ranker_label/1

Hook predicate that importing rankers must define to provide a human-readable label used by pretty-printing helpers.

Compilation flags:
    static

Template:
    score_ranker_label(Label)
Mode and number of proofs:
    score_ranker_label(-atom) - one

---

score_ranker_term/4

Hook predicate that importing rankers must define to map between the ranker term representation and its items, scores, and diagnostics payloads.

Compilation flags:
    static

Template:
    score_ranker_term(Items,Scores,Diagnostics,Ranker)
Mode and number of proofs:
    score_ranker_term(?list,?list,?list(compound),?compound) - zero_or_one

---

valid_score/1

Hook predicate that importing rankers must define to validate individual learned score values.

Compilation flags:
    static

Template:
    valid_score(Score)
Mode and number of proofs:
    valid_score(+number) - zero_or_one

---

build_score_ranker/5

Builds a self-describing score-ranker term from ordered items, score pairs, effective options, and dataset summary metadata.

Compilation flags:
    static

Template:
    build_score_ranker(Items,Scores,Options,DatasetSummary,Ranker)
Mode and number of proofs:
    build_score_ranker(+list,+list(pair),+list(compound),+list(compound),-compound) - one

---

score_ranker_data/4

Validates and destructures a score-ranker term into its items, score pairs, and diagnostics payloads.

Compilation flags:
    static

Template:
    score_ranker_data(Ranker,Items,Scores,Diagnostics)
Mode and number of proofs:
    score_ranker_data(+compound,-list,-list(pair),-list(compound)) - one

---

valid_score_ranker_diagnostics/3

Hook predicate that importing rankers can define to validate diagnostics beyond the common model, options, and dataset summary metadata.

Compilation flags:
    static

Template:

valid_score_ranker_diagnostics(Items,Scores,Diagnostics)
Mode and number of proofs:
valid_score_ranker_diagnostics(+list,+list(pair),+list(compound)) - zero_or_one

---

initialize_scores/2

Initializes a score dictionary with zero scores for the given ordered items.

Compilation flags:
static

Template:
initialize_scores(Items,Scores)
Mode and number of proofs:
initialize_scores(+list,-compound) - one

---

accumulate_item_scores/3

Accumulates item score deltas into a score dictionary.

Compilation flags:
static

Template:
accumulate_item_scores(ItemDeltas,Scores,UpdatedScores)
Mode and number of proofs:
accumulate_item_scores(+list(pair),+compound,-compound) - one

---

update_score/4

Updates the score of a single item in a score dictionary by adding the given delta.

Compilation flags:
    static

Template:
    update_score(Scores,Item,Delta,UpdatedScores)
Mode and number of proofs:
    update_score(+compound,+term,+number,-compound) - one

---

ordered_scores/3

Materializes ordered Item-Score pairs from a score dictionary using the given item order.

Compilation flags:
    static

Template:
    ordered_scores(Items,ScoresDictionary,Scores)
Mode and number of proofs:
    ordered_scores(+list,+compound,-list(pair)) - one

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.192.14 temporal_pairwise_ranking_dataset_protocol

Protocol for temporal pairwise ranking datasets with explicit rating periods and per-game outcomes.

Availability:
logtalk_load(ranking_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - item/1
  - period/1
  - game/4
- Protected predicates
- Private predicates
- Operators

Public predicates

item/1

Enumerates by backtracking the items that can be ranked.

Compilation flags:
    static

Template:
    item(Item)
Mode and number of proofs:
    item(-atom) - zero_or_more

---

period/1

Enumerates by backtracking the rating periods in declaration order.

Compilation flags:
    static

Template:
    period(Period)
Mode and number of proofs:
    period(-atom) - zero_or_more

---

game/4

Enumerates by backtracking observed games where Score is the result for Item1 against Item2 in the given Period.

Compilation flags:
    static

Template:
    game(Period,Item1,Item2,Score)
Mode and number of proofs:

game(-atom,-atom,-atom,-number) - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.193 reader

object

1.193.1 reader

Predicates for reading text file and text stream contents to lists of terms, characters, or character codes and for reading binary file and binary stream contents to lists of bytes.

Availability:
logtalk_load(reader(loader))

Author: Paulo Moura
Version: 2:2:0
Date: 2023-11-14

Compilation flags:
static, context_switching_calls

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - file_to_codes/2
  - file_to_codes/3
  - file_to_chars/2
  - file_to_chars/3
  - file_to_terms/2
  - file_to_terms/3
  - file_to_bytes/2
  - file_to_bytes/3
  - stream_to_codes/2
  - stream_to_codes/3
  - stream_to_chars/2
  - stream_to_chars/3
  - stream_to_terms/2
  - stream_to_terms/3
  - stream_to_bytes/2
  - stream_to_bytes/3
  - line_to_chars/2
  - line_to_chars/3
  - line_to_codes/2
  - line_to_codes/3
- Protected predicates
- Private predicates
- Operators

Public predicates

file_to_codes/2

Reads a text file into a list of character codes.

Compilation flags:
static

Template:
file_to_codes(File,Codes)
Mode and number of proofs:
file_to_codes(+atom,-list(character_code)) - one

---

file_to_codes/3

Reads a text file into a list of character codes. The list is terminated by the given tail.

Compilation flags:
static

Template:
file_to_codes(File,Codes,Tail)
Mode and number of proofs:
file_to_codes(+atom,-list(character_code),@term) - one

---

file_to_chars/2

Reads a text file into a list of characters.

Compilation flags:
static

Template:
file_to_chars(File,Chars)
Mode and number of proofs:
file_to_chars(+atom,-list(character)) - one

---

file_to_chars/3

Reads a text file into a list of characters. The list is terminated by the given tail.

Compilation flags:
    static

Template:
    file_to_chars(File,Chars,Tail)
Mode and number of proofs:
    file_to_chars(+atom,-list(character),@term) - one

---

file_to_terms/2

Reads a text file into a list of terms.

Compilation flags:
    static

Template:
    file_to_terms(File,Terms)
Mode and number of proofs:
    file_to_terms(+atom,-list(term)) - one

---

file_to_terms/3

Reads a text file into a list of terms. The list is terminated by the given tail.

Compilation flags:
    static

Template:
    file_to_terms(File,Terms,Tail)
Mode and number of proofs:
    file_to_terms(+atom,-list(term),@term) - one

---

file_to_bytes/2

Reads a binary file into a list of bytes.

Compilation flags:
    static

Template:
    file_to_bytes(File,Bytes)
Mode and number of proofs:
    file_to_bytes(+atom,-list(byte)) - one

---

file_to_bytes/3

Reads a binary file into a list of bytes. The list is terminated by the given tail.

Compilation flags:
    static

Template:
    file_to_bytes(File,Bytes,Tail)
Mode and number of proofs:
    file_to_bytes(+atom,-list(byte),@term) - one

---

stream_to_codes/2

Reads a text stream into a list of character codes. Does not close the stream.

Compilation flags:
    static

Template:
    stream_to_codes(Stream,Codes)
Mode and number of proofs:
    stream_to_codes(+stream_or_alias,-list(character_code)) - one

---

stream_to_codes/3

Reads a text stream into a list of character codes. Does not close the stream. The list is terminated by the given tail.

Compilation flags:
    static

Template:
    stream_to_codes(Stream,Codes,Tail)
Mode and number of proofs:
    stream_to_codes(+stream_or_alias,-list(character_code),@term) - one

---

stream_to_chars/2

Reads a text stream into a list of characters. Does not close the stream.

Compilation flags:
    static

Template:
    stream_to_chars(Stream,Chars)
Mode and number of proofs:
    stream_to_chars(+stream_or_alias,-list(char)) - one

---

stream_to_chars/3

Reads a text stream into a list of characters. Does not close the stream. The list is terminated by the given tail.

Compilation flags:
    static

Template:
    stream_to_chars(Stream,Chars,Tail)
Mode and number of proofs:
    stream_to_chars(+stream_or_alias,-list(char),@term) - one

---

stream_to_terms/2

Reads a text stream into a list of terms. Does not close the stream.

Compilation flags:
    static

Template:
    stream_to_terms(Stream,Terms)
Mode and number of proofs:
    stream_to_terms(+stream_or_alias,-list(term)) - one

---

stream_to_terms/3

Reads a text stream into a list of terms. Does not close the stream. The list is terminated by the given tail.

Compilation flags:
    static

Template:
    stream_to_terms(Stream,Terms,Tail)
Mode and number of proofs:
    stream_to_terms(+stream_or_alias,-list(term),@term) - one

---

stream_to_bytes/2

Reads a binary stream into a list of bytes. Does not close the stream.

Compilation flags:
    static

Template:
    stream_to_bytes(Stream,Bytes)
Mode and number of proofs:

stream_to_bytes(+stream_or_alias,-list(byte)) - one

---

stream_to_bytes/3

Reads a binary stream into a list of bytes. Does not close the stream. The list is terminated by the given tail.

Compilation flags:
static

Template:
stream_to_bytes(Stream,Bytes,Tail)
Mode and number of proofs:
stream_to_bytes(+stream_or_alias,-list(byte),@term) - one

---

line_to_chars/2

Reads a line from a text stream into a list of characters. Discards the end-of-line characters. Unifies Chars with end_of_file at the end of the file.

Compilation flags:
static

Template:
line_to_chars(Stream,Chars)
Mode and number of proofs:
line_to_chars(+stream_or_alias,-types([atom,list(character)])) - one

---

line_to_chars/3

Reads a line from a text stream into a list of characters. Keeps the end-of-line marker normalized to the line feed control character. The list is terminated by the given tail, which is unified with the empty list at the end of the file.

Compilation flags:
    static

Template:
    line_to_chars(Stream,Chars,Tail)
Mode and number of proofs:
    line_to_chars(+stream_or_alias,-list(character),?term) - one

---

line_to_codes/2

Reads a line from a text stream into a list of character codes. Discards the end-of-line character codes. Unifies Codes with end_of_file at the end of the file.

Compilation flags:
    static

Template:
    line_to_codes(Stream,Codes)
Mode and number of proofs:
    line_to_codes(+stream_or_alias,-types([atom,list(character_code)])) - one

---

line_to_codes/3

Reads a line from a text stream into a list of character codes. Keeps the end-of-line marker normalized to the line feed control character code. The list is terminated by the given tail, which is unified with the empty list at the end of the file.

Compilation flags:
    static

Template:

line_to_codes(Stream,Codes,Tail)
Mode and number of proofs:
line_to_codes(+stream_or_alias,-list(character_code),?term) - one

---

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.194 recorded_database

object

1.194.1 recorded_database

Legacy recorded database predicates. Provides an application global database.

Availability:
logtalk_load(recorded_database(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2023-12-17

Compilation flags:
static, context_switching_calls

Imports:
public recorded_database_core

Remarks:
(none)

Inherited public predicates:
erase/1 instance/2 recorda/2 recorda/3 recorded/2 recorded/3 recordz/2 recordz/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.194.2 recorded_database_core

Legacy recorded database predicates. Can be imported into an object to provide a local database.

Availability:
logtalk_load(recorded_database(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2023-12-17

Compilation flags:
static

Dependencies:

(none)

Remarks:

- References: Opaque ground terms.

Inherited public predicates:
(none)

- Public predicates
  - recorda/3
  - recorda/2
  - recordz/3
  - recordz/2
  - recorded/3
  - recorded/2
  - erase/1
  - instance/2
- Protected predicates
- Private predicates
  - record_/3
  - reference_/1
- Operators

Public predicates

recorda/3

Adds a term as the first term for the given key, returning its reference.

Compilation flags:
static

Template:
recorda(Key,Term,Reference)
Mode and number of proofs:
recorda(+recorded_database_key,+term,--recorded_database_reference) - one_or_error

Exceptions:
Key is a variable:
instantiation_error
Key is neither a variable nor an atomic term or compound term:
type_error(recorded_database_key,Key)
Reference is a not a variable:
uninstantiation_error(Reference)

---

recorda/2

Adds a term as the first term for the given key.

Compilation flags:
static

Template:
recorda(Key,Term)
Mode and number of proofs:
recorda(+recorded_database_key,+term) - one_or_error

Exceptions:
Key is a variable:
instantiation_error
Key is neither a variable nor an atomic term or compound term:
type_error(recorded_database_key,Key)

---

recordz/3

Adds a term as the last term for the given key, returning its reference.

Compilation flags:
static

Template:
recordz(Key,Term,Reference)
Mode and number of proofs:
recordz(+recorded_database_key,+term,--recorded_database_reference) - one_or_error

Exceptions:
Key is a variable:
instantiation_error
Key is neither a variable nor an atomic term or compound term:
type_error(recorded_database_key,Key)
Reference is a not a variable:
uninstantiation_error(Reference)

---

recordz/2

Adds a term as the last term for the given key.

Compilation flags:
static

Template:
recordz(Key,Term)
Mode and number of proofs:
recordz(+recorded_database_key,+term) - one_or_error

Exceptions:
Key is a variable:
instantiation_error
Key is neither a variable nor an atomic term or compound term:
type_error(recorded_database_key,Key)

---

recorded/3

Enumerates, by backtracking, all record key-term pairs and their references.

Compilation flags:
static

Template:
recorded(Key,Term,Reference)
Mode and number of proofs:
recorded(?recorded_database_key,?term,-recorded_database_reference) - zero_or_more

recorded(?recorded_database_key,?term,+recorded_database_reference) - zero_or_one

---

recorded/2

Enumerates, by backtracking, all record key-term pairs.

Compilation flags:
    static

Template:
    recorded(Key,Term)
Mode and number of proofs:
    recorded(?recorded_database_key,?term) - zero_or_more

---

erase/1

Erases the record indexed by the given reference. Fails if there is no record with the given reference.

Compilation flags:
    static

Template:
    erase(Reference)
Mode and number of proofs:
    erase(@recorded_database_reference) - zero_or_one_or_error

Exceptions:
    Reference is a variable:
        instantiation_error

---

instance/2

.

Compilation flags:
    static

Template:
    instance(Reference,Term)
Mode and number of proofs:
    instance(@recorded_database_reference,?term) - zero_or_one_or_error

Exceptions:
    Reference is a variable:
        instantiation_error

Protected predicates

(none)

Private predicates

record_/3

Records table.

Compilation flags:
    dynamic

Template:
    record_(Key,Term,Reference)
Mode and number of proofs:
    record_(?recorded_database_key,?term,?recorded_database_reference) - zero_or_more

---

reference_/1

Reference count.

Compilation flags:
dynamic

Template:
reference_(Reference)
Mode and number of proofs:
reference_(?non_negative_integer) - zero_or_one

Operators

(none)

1.195 redis

object

1.195.1 redis

Redis client library with support for strings, keys, hashes, lists, sets, and sorted sets. Inspired by Sean Charles GNU Prolog Redis client.

Availability:
logtalk_load(redis(loader))

Author: Paulo Moura
Version: 0:7:0
Date: 2026-02-10

Compilation flags:
static, context_switching_calls

Provides:
logtalk::message_tokens//2
Uses:
list

logtalk
socket

Remarks:

- Command representation: Use the Redis command name as the functor of a compound term where the arguments are the command arguments.
- Valid arguments: Atoms, integers, and floats. Always use atoms instead of double-quoted "strings". This helps portability by not depending on the value of the double_quotes flag.
- Wrapper predicates: The library provides convenient wrapper predicates for common Redis operations. For operations without a wrapper, use the generic send/3 predicate.

Inherited public predicates:
(none)

- Public predicates
  - connect/1
  - connect/3
  - disconnect/1
  - send/3
  - console/1
  - get/3
  - set/4
  - append/4
  - getrange/5
  - setrange/5
  - strlen/3
  - mget/3
  - mset/3
  - incr/3
  - decr/3
  - incrby/4
  - decrby/4
  - del/3
  - exists/3
  - keys/3

  - ttl/3
  - expire/4
  - persist/3
  - rename/4
  - type/3
  - hset/5
  - hget/4
  - hgetall/3
  - hdel/4
  - hexists/4
  - hkeys/3
  - hvals/3
  - hlen/3
  - lpush/4
  - rpush/4
  - lpop/3
  - rpop/3
  - lrange/5
  - llen/3
  - lrem/5
  - ltrim/5
  - sadd/4
  - srem/4
  - smembers/3
  - sismember/4
  - scard/3
  - zadd/5
  - zrem/4
  - zrange/5
  - zrank/4
  - zcard/3
  - zscore/4
- Protected predicates
- Private predicates
- Operators

Public predicates

connect/1

Connect to a Redis server running on localhost using the default 6379 port.

Compilation flags:
    static

Template:
    connect(Connection)
Mode and number of proofs:
    connect(--ground) - one

---

connect/3

Connect to a Redis server running on the given host and port.

Compilation flags:
    static

Template:
    connect(Host,Port,Connection)
Mode and number of proofs:
    connect(+atom,+integer,--ground) - one

---

disconnect/1

Disconnect from a Redis server.

Compilation flags:
    static

Template:
    disconnect(Connection)
Mode and number of proofs:
    disconnect(++ground) - one

---

send/3

Sends a request to the a Redis server and returns its reply.

Compilation flags:
    static

Template:
    send(Connection,Request,Reply)
Mode and number of proofs:
    send(++ground,++callable,--callable) - one

---

console/1

Sends a request to a Redis server running on localhost at the default 6379 port and prints the reply.

Compilation flags:
    static

Template:
    console(Request)
Mode and number of proofs:
    console(++callable) - one

---

get/3

Gets the value of a key.

Compilation flags:
    static

Template:
    get(Connection,Key,Value)
Mode and number of proofs:

get(+ground,+atom,-atom) - one

---

set/4

Sets the value of a key.

Compilation flags:
static

Template:
set(Connection,Key,Value,Status)
Mode and number of proofs:
set(+ground,+atom,+ground,-atom) - one

---

append/4

Appends a value to a key. Returns the length of the string after the append.

Compilation flags:
static

Template:
append(Connection,Key,Value,Length)
Mode and number of proofs:
append(+ground,+atom,+ground,-integer) - one

---

getrange/5

Gets a substring of the string stored at a key.

Compilation flags:
static

Template:

getrange(Connection,Key,Start,End,Substring)
Mode and number of proofs:
getrange(+ground,+atom,+integer,+integer,-atom) - one

---

setrange/5

Overwrites part of a string at a key starting at the specified offset. Returns the length of the string after modification.

Compilation flags:
static

Template:
setrange(Connection,Key,Offset,Value,Length)
Mode and number of proofs:
setrange(+ground,+atom,+integer,+ground,-integer) - one

---

strlen/3

Gets the length of the value stored at a key.

Compilation flags:
static

Template:
strlen(Connection,Key,Length)
Mode and number of proofs:
strlen(+ground,+atom,-integer) - one

---

mget/3

Gets the values of multiple keys. Returns a list of values.

Compilation flags:
    static

Template:
    mget(Connection,Keys,Values)
Mode and number of proofs:
    mget(+ground,+list(atom),-list) - one

---

mset/3

Sets multiple key-value pairs. Pairs should be provided as a flat list [Key1, Value1, Key2, Value2, ...].

Compilation flags:
    static

Template:
    mset(Connection,Pairs,Status)
Mode and number of proofs:
    mset(+ground,+list,-atom) - one

---

incr/3

Increments the integer value of a key by one. Returns the value after increment.

Compilation flags:
    static

Template:
    incr(Connection,Key,Value)
Mode and number of proofs:
    incr(+ground,+atom,-integer) - one

---

decr/3

Decrements the integer value of a key by one. Returns the value after decrement.

Compilation flags:
    static

Template:
    decr(Connection,Key,Value)
Mode and number of proofs:
    decr(+ground,+atom,-integer) - one

---

incrby/4

Increments the integer value of a key by the specified amount. Returns the value after increment.

Compilation flags:
    static

Template:
    incrby(Connection,Key,Increment,Value)
Mode and number of proofs:
    incrby(+ground,+atom,+integer,-integer) - one

---

decrby/4

Decrements the integer value of a key by the specified amount. Returns the value after decrement.

Compilation flags:
    static

Template:
    decrby(Connection,Key,Decrement,Value)
Mode and number of proofs:
    decrby(+ground,+atom,+integer,-integer) - one

---

del/3

Deletes a key. Returns the number of keys removed.

Compilation flags:
    static

Template:
    del(Connection,Key,Count)
Mode and number of proofs:
    del(+ground,+atom,-integer) - one

---

exists/3

Checks if a key exists. Returns 1 if the key exists, 0 otherwise.

Compilation flags:
    static

Template:
    exists(Connection,Key,Exists)
Mode and number of proofs:
    exists(+ground,+atom,-integer) - one

---

keys/3

Finds all keys matching a pattern. Returns a list of keys.

Compilation flags:
    static

Template:
    keys(Connection,Pattern,Keys)
Mode and number of proofs:
    keys(+ground,+atom,-list) - one

---

ttl/3

Gets the time to live for a key in seconds. Returns -1 if the key has no expiry, -2 if the key does not exist.

Compilation flags:
    static

Template:
    ttl(Connection,Key,Seconds)
Mode and number of proofs:
    ttl(+ground,+atom,-integer) - one

---

expire/4

Sets a timeout on a key in seconds. Returns 1 if the timeout was set, 0 if the key does not exist.

Compilation flags:
    static

Template:
    expire(Connection,Key,Seconds,Result)
Mode and number of proofs:
    expire(+ground,+atom,+integer,-integer) - one

---

persist/3

Removes the expiration from a key. Returns 1 if the timeout was removed, 0 if the key does not exist or has no timeout.

Compilation flags:
    static

Template:
    persist(Connection,Key,Result)
Mode and number of proofs:
    persist(+ground,+atom,-integer) - one

---

rename/4

Renames a key. Returns status OK or error if the key does not exist.

Compilation flags:
    static

Template:
    rename(Connection,OldKey,NewKey,Status)
Mode and number of proofs:
    rename(+ground,+atom,+atom,-atom) - one

---

type/3

Gets the type of the value stored at a key. Returns one of: string, list, set, zset, hash, stream, none.

Compilation flags:
    static

Template:
    type(Connection,Key,Type)
Mode and number of proofs:
    type(+ground,+atom,-atom) - one

---

hset/5

Sets a field in a hash. Returns 1 if a new field was created, 0 if the field was updated.

Compilation flags:
    static

Template:
    hset(Connection,Key,Field,Value,Result)
Mode and number of proofs:

hset(+ground,+atom,+atom,+ground,-integer) - one

---

hget/4

Gets the value of a field in a hash.

Compilation flags:
static

Template:
hget(Connection,Key,Field,Value)
Mode and number of proofs:
hget(+ground,+atom,+atom,-ground) - one

---

hgetall/3

Gets all fields and values in a hash. Returns a flat list of alternating fields and values.

Compilation flags:
static

Template:
hgetall(Connection,Key,FieldsValues)
Mode and number of proofs:
hgetall(+ground,+atom,-list) - one

---

hdel/4

Deletes a field from a hash. Returns the number of fields removed.

Compilation flags:
static

Template:

hdel(Connection,Key,Field,Count)
Mode and number of proofs:
hdel(+ground,+atom,+atom,-integer) - one

---

hexists/4

Checks if a field exists in a hash. Returns 1 if the field exists, 0 otherwise.

Compilation flags:
static

Template:
hexists(Connection,Key,Field,Exists)
Mode and number of proofs:
hexists(+ground,+atom,+atom,-integer) - one

---

hkeys/3

Gets all field names in a hash.

Compilation flags:
static

Template:
hkeys(Connection,Key,Fields)
Mode and number of proofs:
hkeys(+ground,+atom,-list) - one

---

hvals/3

Gets all values in a hash.

Compilation flags:
    static

Template:
    hvals(Connection,Key,Values)
Mode and number of proofs:
    hvals(+ground,+atom,-list) - one

---

hlen/3

Gets the number of fields in a hash.

Compilation flags:
    static

Template:
    hlen(Connection,Key,Count)
Mode and number of proofs:
    hlen(+ground,+atom,-integer) - one

---

lpush/4

Prepends a value to a list. Returns the length of the list after the push.

Compilation flags:
    static

Template:
    lpush(Connection,Key,Value,Length)
Mode and number of proofs:
    lpush(+ground,+atom,+ground,-integer) - one

---

rpush/4

Appends a value to a list. Returns the length of the list after the push.

Compilation flags:
    static

Template:
    rpush(Connection,Key,Value,Length)
Mode and number of proofs:
    rpush(+ground,+atom,+ground,-integer) - one

---

lpop/3

Removes and returns the first element of a list.

Compilation flags:
    static

Template:
    lpop(Connection,Key,Value)
Mode and number of proofs:
    lpop(+ground,+atom,-ground) - one

---

rpop/3

Removes and returns the last element of a list.

Compilation flags:
    static

Template:
    rpop(Connection,Key,Value)
Mode and number of proofs:
    rpop(+ground,+atom,-ground) - one

---

lrange/5

Gets a range of elements from a list. Indices are zero-based.

Compilation flags:
    static

Template:
    lrange(Connection,Key,Start,Stop,Elements)
Mode and number of proofs:
    lrange(+ground,+atom,+integer,+integer,-list) - one

---

llen/3

Gets the length of a list.

Compilation flags:
    static

Template:
    llen(Connection,Key,Length)
Mode and number of proofs:
    llen(+ground,+atom,-integer) - one

---

lrem/5

Removes elements from a list. Count > 0: remove from head, Count < 0: remove from tail, Count $= 0$: remove all. Returns the number of removed elements.

Compilation flags:
    static

Template:
    lrem(Connection,Key,Count,Value,Removed)
Mode and number of proofs:
    lrem(+ground,+atom,+integer,+ground,-integer) - one

---

ltrim/5

Trims a list to the specified range.

Compilation flags:
    static

Template:
    ltrim(Connection,Key,Start,Stop,Status)
Mode and number of proofs:
    ltrim(+ground,+atom,+integer,+integer,-atom) - one

---

sadd/4

Adds a member to a set. Returns the number of elements added.

Compilation flags:
    static

Template:
    sadd(Connection,Key,Member,Count)
Mode and number of proofs:
    sadd(+ground,+atom,+ground,-integer) - one

---

srem/4

Removes a member from a set. Returns the number of elements removed.

Compilation flags:
    static

Template:
    srem(Connection,Key,Member,Count)
Mode and number of proofs:

srem(+ground,+atom,+ground,-integer) - one

---

smembers/3

Gets all members in a set.

Compilation flags:
static

Template:
smembers(Connection,Key,Members)
Mode and number of proofs:
smembers(+ground,+atom,-list) - one

---

sismember/4

Checks if a value is a member of a set. Returns 1 if the member exists, 0 otherwise.

Compilation flags:
static

Template:
sismember(Connection,Key,Member,IsMember)
Mode and number of proofs:
sismember(+ground,+atom,+ground,-integer) - one

---

scard/3

Gets the number of members in a set.

Compilation flags:
static

Template:

scard(Connection,Key,Count)
Mode and number of proofs:
scard(+ground,+atom,-integer) - one

---

zadd/5

Adds a member with a score to a sorted set. Returns the number of elements added.

Compilation flags:
static

Template:
zadd(Connection,Key,Score,Member,Count)
Mode and number of proofs:
zadd(+ground,+atom,+number,+ground,-integer) - one

---

zrem/4

Removes a member from a sorted set. Returns the number of elements removed.

Compilation flags:
static

Template:
zrem(Connection,Key,Member,Count)
Mode and number of proofs:
zrem(+ground,+atom,+ground,-integer) - one

---

zrange/5

Gets a range of members from a sorted set, ordered from lowest to highest score.

Compilation flags:
    static

Template:
    zrange(Connection,Key,Start,Stop,Members)
Mode and number of proofs:
    zrange(+ground,+atom,+integer,+integer,-list) - one

---

zrank/4

Gets the rank (index) of a member in a sorted set, ordered from lowest to highest score. Returns nil if the member does not exist.

Compilation flags:
    static

Template:
    zrank(Connection,Key,Member,Rank)
Mode and number of proofs:
    zrank(+ground,+atom,+ground,-integer) - one

---

zcard/3

Gets the number of members in a sorted set.

Compilation flags:
    static

Template:
    zcard(Connection,Key,Count)
Mode and number of proofs:
    zcard(+ground,+atom,-integer) - one

zscore/4

Gets the score of a member in a sorted set.

Compilation flags:
    static

Template:
    zscore(Connection,Key,Member,Score)
Mode and number of proofs:
    zscore(+ground,+atom,+ground,-ground) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.196 regression_protocols

protocol

1.196.1 regression_dataset_protocol

Protocol for datasets used with regression algorithms.

Availability:
    logtalk_load(regression_protocols(loader))

Author: Paulo Moura
Version: 1:0:0

Date: 2026-04-21

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - attribute_values/2
  - target/1
  - example/3
- Protected predicates
- Private predicates
- Operators

Public predicates

attribute_values/2

Enumerates by backtracking the attributes and their possible values. For discrete attributes, Values is a list of possible values. For continuous (numeric) attributes, Values is the atom continuous.

Compilation flags:
static

Template:
attribute_values(Attribute,Values)
Mode and number of proofs:
attribute_values(?atom,-list(atom)) - zero_or_more
attribute_values(?atom,-atom) - zero_or_more

target/1

Returns the name of the numeric target attribute.

Compilation flags:
    static

Template:
    target(Target)
Mode and number of proofs:
    target(-atom) - one

example/3

Enumerates by backtracking the examples in the dataset. Each example has an Id, a numeric Target value, and a list of Attribute-Value pairs.

Compilation flags:
    static

Template:
    example(Id,Target,AttributeValues)
Mode and number of proofs:
    example(-integer,-number,-list(pair)) - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

category

1.196.2 regressor_common

Shared predicates for regressor learning defaults, diagnostics, validation, dataset validation, export, and pretty-print helpers.

Availability:
logtalk_load(regression_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-04

Compilation flags:
static

Implements:
public regressor_protocol
Extends:
public options
Uses:
format
linear_algebra
list
numberlist
population
type

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_regressor/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2

option/3 predict/3 print_regressor/1 regressor_options/2 valid_option/1 valid_options/1 valid_regressor/1

- Public predicates
- Protected predicates
  - regressor_diagnostics_data/2
  - regressor_export_template/4
  - regressor_term_template/2
  - dataset_attributes/2
  - dataset_examples/2
  - check_examples/2
  - print_regressor_template/1
  - base_regressor_diagnostics/6
  - valid_attribute_names/1
  - valid_attribute_declarations/1
  - valid_discrete_values/1
  - valid_regression_encoders/1
  - valid_regressor_options/1
  - valid_regressor_metadata/2
  - valid_diagnostic_count/3
  - valid_linear_model_diagnostics/1
  - valid_encoded_rows/2
  - encoded_feature_count/2
  - continuous_stats/5
  - examples_to_rows/3
  - encode_instance/3
  - fit_linear_model/7
  - fit_ridge_model/7
  - valid_feature_labels/1
  - valid_regression_tree/2
- Private predicates
  - check_attribute_declarations/1
  - known_attribute_values/3
  - check_attribute_bindings/2
  - check_declared_attribute_bindings/2

- check_undeclared_attribute_bindings/2
- attribute_occurrences/4
- declared_attribute_names/2
- encoder_attribute_names/2
- encode_instance_checked/3
- valid_linear_model_convergence/1
- normalize_continuous/4
- check_categorical_value/3
- one_hot_encode/4
- missing_one_hot_encode/3
- zero_vector_from_values/3
- build_linear_encoders/4
- train_linear_model/6
- rows_to_design_matrix/3
- residual_sum_of_squares/4
- train_ridge_model/6
- ridge_feature_statistics/3
- accumulate_feature_statistics/5
- add_squared_vector/3
- feature_penalty_profiles/5
- active_feature_count/2
- compress_rows/3
- compress_features/3
- compress_vector/3
- build_ridge_system/5
- accumulate_ridge_system/5
- add_outer_product/4
- regularize_ridge_matrix/4
- solve_linear_system/4
- augment_rows/3
- triangularize/2
- select_pivot_row/3
- leading_magnitude/2
- ensure_non_zero/1
- eliminate_rows/5

  - scaled_row_difference/4
  - back_substitution/2
  - maximum_linear_system_residual/4
  - expand_weights/3
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

regressor_diagnostics_data/2

Default hook predicate for exposing diagnostics metadata from a regressor term. Importing implementations may override it when using a non-standard regressor representation.

Compilation flags:
    static

Template:
    regressor_diagnostics_data(Regressor,Diagnostics)
Mode and number of proofs:
    regressor_diagnostics_data(+compound,-list(compound)) - one

---

regressor_export_template/4

Hook predicate that importing regressor implementations must define in order to expose the exported regressor template for a given functor.

Compilation flags:
    static

Template:
    regressor_export_template(Dataset,Regressor,Functor,Template)
Mode and number of proofs:
    regressor_export_template(+object_identifier,+compound,+atom,-callable) - one

---

regressor_term_template/2

Hook predicate that importing regressor implementations must define in order to expose the learned regressor term template used by pretty-printing helpers.

Compilation flags:
    static

Template:
    regressor_term_template(Regressor,Template)
Mode and number of proofs:
    regressor_term_template(+compound,-callable) - one

---

dataset_attributes/2

Collects the dataset attribute declarations as Attribute-Values pairs.

Compilation flags:
    static

Template:
    dataset_attributes(Dataset,Attributes)
Mode and number of proofs:
    dataset_attributes(+object_identifier,-list(pair)) - one

---

dataset_examples/2

Collects the dataset examples as example(Id, TargetValue, AttributeValues) terms.

Compilation flags:
    static

Template:
    dataset_examples(Dataset,Examples)

Mode and number of proofs:
dataset_examples(+object_identifier,-list(compound)) - one

---

check_examples/2

Validates that the collected examples list is non-empty, only contains numeric targets, and only uses declared attributes without duplicate bindings.

Compilation flags:
static

Template:
check_examples(Dataset,Examples)
Mode and number of proofs:
check_examples(+object_identifier,+list) - one_or_error

Exceptions:
Examples is the empty list:
domain_error(non_empty_dataset,Dataset)
An example target is not numeric:
type_error(number,Target)
An example repeats a declared attribute binding:
domain_error(attribute_occurrences,Attribute)
An example contains an undeclared attribute binding:
domain_error(declared_attribute,Attribute)

---

print_regressor_template/1

Pretty-printing helper predicate used by importing regressor implementations to show the learned regressor term template.

Compilation flags:
static

Template:
print_regressor_template(Regressor)
Mode and number of proofs:
print_regressor_template(+compound) - one

---

base_regressor_diagnostics/6

Builds common diagnostics metadata terms for a learned regressor and appends regressor-specific diagnostics terms.

Compilation flags:
    static

Template:
    base_regressor_diagnostics(Model,Target,TrainingExampleCount,Options,ExtraDiagnostics,
    Diagnostics)
Mode and number of proofs:
    base_regressor_diagnostics(+atom,+atom,+integer,+list(compound),+list(compound),
    -list(compound)) - one

---

valid_attribute_names/1

True when a list of attribute names is a proper list of distinct atoms.

Compilation flags:
    static

Template:
    valid_attribute_names(AttributeNames)
Mode and number of proofs:
    valid_attribute_names(+list(atom)) - zero_or_one

---

valid_attribute_declarations/1

True when a list of attribute declarations is a proper list of distinct Attribute-Values pairs where values are either continuous or a valid discrete value list.

Compilation flags:
    static

Template:
valid_attribute_declarations(Attributes)
Mode and number of proofs:
valid_attribute_declarations(+list(pair)) - zero_or_one

---

valid_discrete_values/1

True when a list of categorical values is non-empty, contains only nonvar terms, and has no duplicates.

Compilation flags:
static

Template:
valid_discrete_values(Values)
Mode and number of proofs:
valid_discrete_values(+list) - zero_or_one

---

valid_regression_encoders/1

True when a list of encoders only contains valid continuous/3 or categorical/2 encoder terms with distinct attributes.

Compilation flags:
static

Template:
valid_regression_encoders(Encoders)
Mode and number of proofs:
valid_regression_encoders(+list(compound)) - zero_or_one

---

valid_regressor_options/1

True when a list of options is structurally valid for the receiving regressor implementation.

Compilation flags:
    static

Template:
    valid_regressor_options(Options)
Mode and number of proofs:
    valid_regressor_options(+list(compound)) - zero_or_one

---

valid_regressor_metadata/2

True when diagnostics metadata contains the expected model term and records a structurally valid effective options list.

Compilation flags:
    static

Template:
    valid_regressor_metadata(Model,Diagnostics)
Mode and number of proofs:
    valid_regressor_metadata(+atom,+list(compound)) - zero_or_one

---

valid_diagnostic_count/3

True when diagnostics contains a count term with the given functor and integer value.

Compilation flags:
    static

Template:
    valid_diagnostic_count(Functor,Diagnostics,Count)
Mode and number of proofs:
    valid_diagnostic_count(+atom,+list(compound),+integer) - zero_or_one

---

valid_linear_model_diagnostics/1

True when diagnostics contains structurally valid linear-model optimization metadata terms for convergence, completed iterations, and final parameter delta.

Compilation flags:
    static

Template:
    valid_linear_model_diagnostics(Diagnostics)
Mode and number of proofs:
    valid_linear_model_diagnostics(+list(compound)) - zero_or_one

---

valid_encoded_rows/2

True when encoded training rows match the feature count induced by the encoders and carry numeric targets.

Compilation flags:
    static

Template:
    valid_encoded_rows(Encoders,Rows)
Mode and number of proofs:
    valid_encoded_rows(+list(compound),+list) - zero_or_one

---

encoded_feature_count/2

Counts the number of numeric features induced by a list of continuous and categorical encoders, including missing-value indicator features.

Compilation flags:
    static

Template:

encoded_feature_count(Encoders,FeatureCount)
Mode and number of proofs:
encoded_feature_count(+list(compound),-integer) - one

---

continuous_stats/5

Computes the mean and scaling factor used to encode a continuous attribute from the training examples according to the effective feature scaling option.

Compilation flags:
static

Template:
continuous_stats(Attribute,Examples,Options,Mean,Scale)
Mode and number of proofs:
continuous_stats(+atom,+list(compound),+list(compound),-float,-positive_float) - one

---

examples_to_rows/3

Encodes dataset examples as numeric feature-vector and target pairs using a list of encoders.

Compilation flags:
static

Template:
examples_to_rows(Examples,Encoders,Rows)
Mode and number of proofs:
examples_to_rows(+list(compound),+list(compound),-list(pair)) - one

---

encode_instance/3

Validates and encodes an attribute-value list as a numeric feature vector using a list of continuous and categorical encoders.

Compilation flags:
static

Template:
encode_instance(Encoders,AttributeValues,Features)
Mode and number of proofs:
encode_instance(+list(compound),+list(pair),-list(float)) - one

---

fit_linear_model/7

Builds linear-model encoders from the training dataset, encodes the examples, fits an ordinary least-squares bias plus weight vector using the linear_algebra least-squares solver, and returns solver diagnostics terms.

Compilation flags:
static

Template:
fit_linear_model(Dataset,Options,Encoders,TrainingExampleCount,Bias,Weights,
TrainingDiagnostics)
Mode and number of proofs:
fit_linear_model(+object_identifier,+list(compound),-list(compound),-integer,-float,-list(float),
-list(compound)) - one

---

fit_ridge_model/7

Builds linear-model encoders from the training dataset, encodes the examples, fits a bias plus weight vector by solving the ridge normal equations with partial pivoting, and returns ridge-specific diagnostics terms.

Compilation flags:
static

Template:

fit_ridge_model(Dataset,Options,Encoders,TrainingExampleCount,Bias,Weights,
TrainingDiagnostics)

Mode and number of proofs:

fit_ridge_model(+object_identifier,+list(compound),-list(compound),-integer,-float,-list(float),
-list(compound)) - one

---

valid_feature_labels/1

True when a list of regression-tree feature labels only contains valid feature/2 terms.

Compilation flags:

static

Template:

valid_feature_labels(FeatureLabels)

Mode and number of proofs:

valid_feature_labels(+list(compound)) - zero_or_one

---

valid_regression_tree/2

True when a regression tree only contains valid leaf/1 and node/5 terms using feature indexes within bounds.

Compilation flags:

static

Template:

valid_regression_tree(Tree,FeatureCount)

Mode and number of proofs:

valid_regression_tree(+compound,+positive_integer) - zero_or_one

---

Private predicates

check_attribute_declarations/1

Checks that dataset attribute declarations use distinct attribute names and valid value domains.

Compilation flags:
    static

Template:
    check_attribute_declarations(Attributes)
Mode and number of proofs:
    check_attribute_declarations(+list(pair)) - one_or_error

Exceptions:
    An attribute is declared more than once or uses an invalid value domain:
        domain_error(attribute_declarations,Attribute)

---

known_attribute_values/3

Collects the known numeric values for a continuous attribute from the training examples, skipping omitted and variable values.

Compilation flags:
    static

Template:
    known_attribute_values(Examples,Attribute,Values)
Mode and number of proofs:
    known_attribute_values(+list(compound),+atom,-list(number)) - one

---

check_attribute_bindings/2

Checks that an attribute-value list contains only declared attributes and does not repeat any declared attribute; omitted declared attributes are allowed and treated as missing values.

Compilation flags:
    static

Template:
    check_attribute_bindings(AttributeNames,AttributeValues)
Mode and number of proofs:
    check_attribute_bindings(+list(atom),+list(pair)) - one

---

check_declared_attribute_bindings/2

Checks that no declared attribute appears more than once in an attribute-value list.

Compilation flags:
    static

Template:
    check_declared_attribute_bindings(AttributeNames,AttributeValues)
Mode and number of proofs:
    check_declared_attribute_bindings(+list(atom),+list(pair)) - one

---

check_undeclared_attribute_bindings/2

Checks that every attribute occurring in an attribute-value list is declared.

Compilation flags:
    static

Template:
    check_undeclared_attribute_bindings(AttributeValues,AttributeNames)
Mode and number of proofs:
    check_undeclared_attribute_bindings(+list(pair),+list(atom)) - one

---

attribute_occurrences/4

Counts the number of times an attribute occurs in an attribute-value list.

Compilation flags:
    static

Template:
    attribute_occurrences(AttributeValues,Attribute,Count0,Count)
Mode and number of proofs:
    attribute_occurrences(+list(pair),+atom,+integer,-integer) - one

---

declared_attribute_names/2

Collects the declared attribute names from dataset attribute declarations.

Compilation flags:
    static

Template:
    declared_attribute_names(Attributes,AttributeNames)
Mode and number of proofs:
    declared_attribute_names(+list(pair),-list(atom)) - one

---

encoder_attribute_names/2

Collects the declared attribute names from a list of regression encoders.

Compilation flags:
    static

Template:
    encoder_attribute_names(Encoders,AttributeNames)
Mode and number of proofs:

encoder_attribute_names(+list(compound),-list(atom)) - one

---

encode_instance_checked/3

Encodes an already validated attribute-value list as a numeric feature vector using a list of continuous and categorical encoders.

Compilation flags:
static

Template:
encode_instance_checked(Encoders,AttributeValues,Features)
Mode and number of proofs:
encode_instance_checked(+list(compound),+list(pair),-list(float)) - one

---

valid_linear_model_convergence/1

True when a linear-model optimization stop reason is recognized.

Compilation flags:
static

Template:
valid_linear_model_convergence(Convergence)
Mode and number of proofs:
valid_linear_model_convergence(+atom) - zero_or_one

---

normalize_continuous/4

Normalizes a continuous value using a stored mean and scale.

Compilation flags:
static

Template:
normalize_continuous(Value,Mean,Scale,Feature)
Mode and number of proofs:
normalize_continuous(+number,+float,+positive_float,-float) - one

---

check_categorical_value/3

Validates that a categorical value is declared for an attribute.

Compilation flags:
static

Template:
check_categorical_value(Attribute,Values,Value)
Mode and number of proofs:
check_categorical_value(+atom,+list,+nonvar) - one

---

one_hot_encode/4

Encodes a declared categorical value using reference-level dummy coding plus a trailing missing-value indicator feature set to zero.

Compilation flags:
static

Template:
one_hot_encode(Values,Value,Encoded,Tail)
Mode and number of proofs:
one_hot_encode(+list,+nonvar,-list(float),+list(float)) - one

---

missing_one_hot_encode/3

Encodes a missing categorical value as reference-level dummy zeroes plus a trailing missing-value indicator feature set to one.

Compilation flags:
    static

Template:
    missing_one_hot_encode(Values,Encoded,Tail)
Mode and number of proofs:
    missing_one_hot_encode(+list,-list(float),+list(float)) - one

---

zero_vector_from_values/3

Creates a zero vector with one element per supplied categorical value slot.

Compilation flags:
    static

Template:
    zero_vector_from_values(Values,Zeroes,Tail)
Mode and number of proofs:
    zero_vector_from_values(+list,-list(float),+list(float)) - one

---

build_linear_encoders/4

Builds the encoder list used by linear models from dataset attribute declarations, training examples, and the effective feature scaling option.

Compilation flags:
    static

Template:
    build_linear_encoders(Attributes,Examples,Options,Encoders)
Mode and number of proofs:
    build_linear_encoders(+list(pair),+list(compound),+list(compound),-list(compound)) - one

---

train_linear_model/6

Fits an ordinary least-squares linear model bias and weight vector from encoded rows by delegating the solve and rank estimation to the linear_algebra library and returns solver diagnostics terms.

Compilation flags:
    static

Template:
    train_linear_model(Rows,FeatureCount,Options,Bias,Weights,TrainingDiagnostics)
Mode and number of proofs:
    train_linear_model(+list(pair),+integer,+list(compound),-float,-list(float),-list(compound)) - one

---

rows_to_design_matrix/3

Transforms encoded training rows into a row-oriented design matrix with a leading intercept column and a target vector.

Compilation flags:
    static

Template:
    rows_to_design_matrix(Rows,DesignMatrix,Targets)
Mode and number of proofs:
    rows_to_design_matrix(+list(pair),-list(list(float)),-list(float)) - one

---

residual_sum_of_squares/4

Computes the residual sum of squares for encoded training rows and a learned intercept plus weight vector.

Compilation flags:
    static

Template:

residual_sum_of_squares(Rows,Bias,Weights,ResidualSumOfSquares)
Mode and number of proofs:
residual_sum_of_squares(+list(pair),+float,+list(float),-float) - one

---

train_ridge_model/6

Fits a ridge model bias and weight vector from encoded rows by solving a direct weighted linear system and returns diagnostics terms describing the solve.

Compilation flags:
static

Template:
train_ridge_model(Rows,FeatureCount,Options,Bias,Weights,TrainingDiagnostics)
Mode and number of proofs:
train_ridge_model(+list(pair),+integer,+list(compound),-float,-list(float),-list(compound)) - one

---

ridge_feature_statistics/3

Computes per-feature ridge penalty weights and active-feature flags from encoded training rows, dropping zero-variance columns from the direct solve.

Compilation flags:
static

Template:
ridge_feature_statistics(Rows,PenaltyWeights,ActiveFlags)
Mode and number of proofs:
ridge_feature_statistics(+list(pair),-list(float),-list(atom)) - one

---

accumulate_feature_statistics/5

Accumulates per-feature sums and squared sums over encoded training rows.

Compilation flags:
    static

Template:
    accumulate_feature_statistics(Rows,Sums0,SumSquares0,Sums,SumSquares)
Mode and number of proofs:
    accumulate_feature_statistics(+list(pair),+list(float),+list(float),-list(float),-list(float)) - one

---

add_squared_vector/3

Adds the element-wise squares of a feature vector to an accumulated vector.

Compilation flags:
    static

Template:
    add_squared_vector(Features,AccumulatedSquares0,AccumulatedSquares)
Mode and number of proofs:
    add_squared_vector(+list(float),+list(float),-list(float)) - one

---

feature_penalty_profiles/5

Builds ridge penalty weights and active-feature flags from per-feature sums and squared sums.

Compilation flags:
    static

Template:
    feature_penalty_profiles(Sums,SumSquares,Count,PenaltyWeights,ActiveFlags)
Mode and number of proofs:
    feature_penalty_profiles(+list(float),+list(float),+integer,-list(float),-list(atom)) - one

---

active_feature_count/2

Counts the number of active encoded features retained for the direct ridge solve.

Compilation flags:
    static

Template:
    active_feature_count(ActiveFlags,Count)
Mode and number of proofs:
    active_feature_count(+list(atom),-integer) - one

---

compress_rows/3

Filters encoded training rows down to the active feature subset selected for the direct ridge solve.

Compilation flags:
    static

Template:
    compress_rows(Rows,ActiveFlags,CompressedRows)
Mode and number of proofs:
    compress_rows(+list(pair),+list(atom),-list(pair)) - one

---

compress_features/3

Filters a feature vector down to the active feature subset selected for the direct ridge solve.

Compilation flags:
    static

Template:
    compress_features(Features,ActiveFlags,CompressedFeatures)
Mode and number of proofs:
    compress_features(+list(float),+list(atom),-list(float)) - one

---

compress_vector/3

Filters a numeric vector down to the active positions selected for the direct ridge solve.

Compilation flags:
    static

Template:
    compress_vector(Vector,ActiveFlags,CompressedVector)
Mode and number of proofs:
    compress_vector(+list(float),+list(atom),-list(float)) - one

---

build_ridge_system/5

Builds the regularized linear system for the intercept plus active ridge coefficients.

Compilation flags:
    static

Template:
    build_ridge_system(Rows,PenaltyWeights,Regularization,Matrix,Vector)
Mode and number of proofs:
    build_ridge_system(+list(pair),+list(float),+float,-list(list(float)),-list(float)) - one

---

accumulate_ridge_system/5

Accumulates the unregularized normal-equation matrix and target vector for a set of active encoded rows.

Compilation flags:
    static

Template:
    accumulate_ridge_system(Rows,Matrix0,Vector0,Matrix,Vector)
Mode and number of proofs:
    accumulate_ridge_system(+list(pair),+list(list(float)),+list(float),-list(list(float)),-list(float)) - one

---

add_outer_product/4

Adds the outer product of two vectors to an accumulated matrix.

Compilation flags:
    static

Template:
    add_outer_product(Vector1,Vector2,Matrix0,Matrix)
Mode and number of proofs:
    add_outer_product(+list(float),+list(float),+list(list(float)),-list(list(float))) - one

---

regularize_ridge_matrix/4

Adds the ridge penalty weights to the feature-feature diagonal block of a linear system matrix.

Compilation flags:
    static

Template:
    regularize_ridge_matrix(Matrix0,PenaltyWeights,Regularization,Matrix)
Mode and number of proofs:
    regularize_ridge_matrix(+list(list(float)),+list(float),+float,-list(list(float))) - one

---

solve_linear_system/4

Solves a square linear system using partial pivoting Gaussian elimination and returns the solver name.

Compilation flags:
    static

Template:
    solve_linear_system(Matrix,Vector,Solution,Solver)
Mode and number of proofs:
    solve_linear_system(+list(list(float)),+list(float),-list(float),-atom) - one_or_error

---

augment_rows/3

Pairs each matrix row with its corresponding right-hand-side value for elimination.

Compilation flags:
    static

Template:
    augment_rows(Matrix,Vector,Rows)
Mode and number of proofs:
    augment_rows(+list(list(float)),+list(float),-list(compound)) - one

---

triangularize/2

Transforms an augmented matrix into upper-triangular form using partial pivoting.

Compilation flags:
    static

Template:
    triangularize(Rows0,UpperRows)
Mode and number of proofs:
    triangularize(+list(compound),-list(compound)) - one_or_error

---

select_pivot_row/3

Selects the pivot row with the largest leading magnitude and returns the remaining rows.

Compilation flags:
    static

Template:
    select_pivot_row(Rows0,PivotRow,RemainingRows)
Mode and number of proofs:
    select_pivot_row(+list(compound),-compound,-list(compound)) - one

---

leading_magnitude/2

Returns the absolute leading coefficient magnitude of an augmented row.

Compilation flags:
    static

Template:
    leading_magnitude(Row,Magnitude)
Mode and number of proofs:
    leading_magnitude(+compound,-float) - one

---

ensure_non_zero/1

Checks that a pivot coefficient is numerically non-zero.

Compilation flags:
    static

Template:
    ensure_non_zero(Value)
Mode and number of proofs:
    ensure_non_zero(+float) - one_or_error

Exceptions:
    A pivot coefficient is numerically zero:
        evaluation_error(zero_divisor)

---

eliminate_rows/5

Eliminates the leading coefficient from remaining augmented rows using a pivot row.

Compilation flags:
    static

Template:

eliminate_rows(Pivot,PivotTail,PivotValue,Rows0,Rows)
Mode and number of proofs:
eliminate_rows(+float,+list(float),+float,+list(compound),-list(compound)) - one

---

scaled_row_difference/4

Subtracts a scaled pivot tail from another row tail.

Compilation flags:
static

Template:
scaled_row_difference(PivotTail,RowTail,Factor,Difference)
Mode and number of proofs:
scaled_row_difference(+list(float),+list(float),+float,-list(float)) - one

---

back_substitution/2

Performs back-substitution on an upper-triangular augmented matrix.

Compilation flags:
static

Template:
back_substitution(UpperRows,Solution)
Mode and number of proofs:
back_substitution(+list(compound),-list(float)) - one_or_error

---

maximum_linear_system_residual/4

Computes the maximum absolute residual of a solved linear system.

Compilation flags:
    static

Template:
    maximum_linear_system_residual(Matrix,Vector,Solution,MaximumResidual)
Mode and number of proofs:
    maximum_linear_system_residual(+list(list(float)),+list(float),+list(float),-float) - one

---

expand_weights/3

Expands active ridge coefficients back to the full encoded feature vector, inserting zeroes for dropped zero-variance features.

Compilation flags:
    static

Template:
    expand_weights(ActiveFlags,ActiveWeights,Weights)
Mode and number of proofs:
    expand_weights(+list(atom),+list(float),-list(float)) - one

---

Operators

(none)

protocol

1.196.3 regressor_protocol

Protocol for machine learning regressors.

Availability:
logtalk_load(regression_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-01

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - learn/3
  - learn/2
  - predict/3
  - check_regressor/1
  - valid_regressor/1
  - diagnostics/2
  - diagnostic/2
  - regressor_options/2
  - export_to_clauses/4
  - export_to_file/4
  - print_regressor/1
- Protected predicates
- Private predicates

- Operators

Public predicates

learn/3

Learns a regressor from the given dataset object using the specified options.

Compilation flags:
    static

Template:
    learn(Dataset,Regressor,Options)
Mode and number of proofs:
    learn(+object_identifier,-compound,+list(compound)) - one

---

learn/2

Learns a regressor from the given dataset object using default options.

Compilation flags:
    static

Template:
    learn(Dataset,Regressor)
Mode and number of proofs:
    learn(+object_identifier,-compound) - one

---

predict/3

Predicts the numeric target value for a new instance using the learned regressor. The instance is a list of Attribute-Value pairs.

Compilation flags:
    static

Template:
predict(Regressor,Instance,Target)
Mode and number of proofs:
predict(+compound,+list,-number) - one

---

check_regressor/1

Checks that a learned regressor term is structurally valid for the receiving implementation. Throws an exception when the term is not a valid regressor representation.

Compilation flags:
static

Template:
check_regressor(Regressor)
Mode and number of proofs:
check_regressor(@compound) - one_or_error

Exceptions:
Regressor is a variable:
instantiation_error
Regressor is neither a variable nor a valid regressor:
domain_error(regressor,Regressor)

---

valid_regressor/1

True when a learned regressor term is structurally valid for the receiving implementation. Succeeds iff check_regressor/1 succeeds without throwing an exception.

Compilation flags:
static

Template:
valid_regressor(Regressor)
Mode and number of proofs:
valid_regressor(@compound) - zero_or_one

---

diagnostics/2

Returns diagnostics metadata for a learned regressor.

Compilation flags:
    static

Template:
    diagnostics(Regressor,Diagnostics)
Mode and number of proofs:
    diagnostics(+compound,-list(compound)) - one

---

diagnostic/2

Enumerates individual diagnostics metadata terms for a learned regressor.

Compilation flags:
    static

Template:
    diagnostic(Regressor,Diagnostic)
Mode and number of proofs:
    diagnostic(+compound,?compound) - zero_or_more

---

regressor_options/2

Returns the effective options stored in a learned regressor diagnostics metadata.

Compilation flags:
    static

Template:
    regressor_options(Regressor,Options)
Mode and number of proofs:
    regressor_options(+compound,-list(compound)) - one

---

export_to_clauses/4

Converts a regressor into a list of predicate clauses. Functor is the functor for the generated predicate clauses. When exporting a serialized regressor term, a noun such as regressor or model is usually clearer than a verb such as predict.

Compilation flags:
    static

Template:
    export_to_clauses(Dataset,Regressor,Functor,Clauses)
Mode and number of proofs:
    export_to_clauses(+object_identifier,+compound,+callable,-list(clause)) - one

---

export_to_file/4

Exports a regressor to a file. Functor is the functor for the generated predicate clauses. When exporting a serialized regressor term, a noun such as regressor or model is usually clearer than a verb such as predict.

Compilation flags:
    static

Template:
    export_to_file(Dataset,Regressor,Functor,File)
Mode and number of proofs:
    export_to_file(+object_identifier,+compound,+callable,+atom) - one

---

print_regressor/1

Prints a regressor to the current output stream in a human-readable format.

Compilation flags:
    static

Template:
    print_regressor(Regressor)
Mode and number of proofs:

print_regressor(+compound) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

linear_regression, regression_dataset_protocol

1.197 regression_tree

object

1.197.1 regression_tree

Regression tree regressor supporting continuous and mixed-feature datasets using recursive variance-reduction splits.

Availability:
logtalk_load(regression_tree(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:
static, context_switching_calls

Imports:
public regressor_common
Uses:

fast_random(Algorithm)
format
list
population
type

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_regressor/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 print_regressor/1 regressor_options/2 valid_option/1 valid_options/1 valid_regressor/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

linear_regression, knn_regression, random_forest_regression, gradient_boosting_regression

1.198 regularized_bradley_terry_ranker

object

1.198.1 regularized_bradley_terry_ranker

Regularized Bradley-Terry MAP ranker. Learns one positive strength parameter per item from a dataset object implementing the pairwise_ranking_dataset_protocol protocol using a Bradley-Terry likelihood with an explicit independent Gamma prior, and returns a self-describing ranker term with diagnostics that can be used for ranking and export.

Availability:
logtalk_load(regularized_bradley_terry_ranker(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public ranking_dataset_common
public pairwise_strength_ranker_common
public score_ranker_common
Uses:
avltree
format
list

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_ranker/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 grouped_dataset_groups/2 grouped_dataset_items/2 grouped_dataset_items/3 grouped_dataset_summary/2 learn/2 learn/3 option/2 option/3 pairwise_dataset_connected_components/2 pairwise_dataset_declared_items/2 pairwise_dataset_items/2 pairwise_dataset_matchups/2 pairwise_dataset_preferences/2 pairwise_dataset_summary/2 pairwise_dataset_win_totals/2 pairwise_measurement_dataset_connected_components/2 pairwise_measurement_dataset_declared_items/2 pairwise_measurement_dataset_items/2 pairwise_measurement_dataset_measurements/2 pairwise_measurement_dataset_summary/2 print_ranker/1 rank/3 ranker_options/2 scores/2

temporal_pairwise_dataset_connected_components/2
temporal_pairwise_dataset_declared_items/2 temporal_pairwise_dataset_games/2
temporal_pairwise_dataset_games/3 temporal_pairwise_dataset_items/2
temporal_pairwise_dataset_periods/2 temporal_pairwise_dataset_summary/2 valid_option/1
valid_options/1 valid_ranker/1 validate_grouped_dataset/1 validate_grouped_dataset/2
validate_pairwise_dataset/1 validate_pairwise_dataset/2
validate_pairwise_measurement_dataset/1 validate_pairwise_measurement_dataset/2
validate_temporal_pairwise_dataset/1 validate_temporal_pairwise_dataset/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

pairwise_ranking_dataset_protocol, ranker_protocol, bradley_terry_ranker

1.199 ridge_regression

object

1.199.1 ridge_regression

Ridge regression regressor supporting continuous and mixed-feature datasets using a direct weighted linear-system solve with L2 regularization. Learns from a dataset object implementing the regression_dataset_protocol protocol and returns a regressor term that can be used for prediction and exported as predicate clauses.

Availability:
logtalk_load(ridge_regression(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-05

Compilation flags:
static, context_switching_calls

Imports:
public regressor_common
Uses:
format
list
numberlist
type

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_regressor/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 print_regressor/1 regressor_options/2 valid_option/1 valid_options/1 valid_regressor/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

linear_regression, knn_regression, regression_tree, random_forest_regression, gradient_boosting_regression

1.200 sarif

object

1.200.1 sarif

Shared SARIF report generator for tools implementing the diagnostics protocol.

Availability:
logtalk_load(sarif(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-04-19

Compilation flags:
static, context_switching_calls

Uses:
git
json(ObjectRepresentation,PairRepresentation,StringRepresentation)
list

logtalk
os
reader
term_io
type
url(Representation)
user
uuid

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - term/2
  - term/4
  - generate/2
  - generate/4
- Protected predicates
- Private predicates
- Operators

Public predicates

term/2

Returns a SARIF 2.1.0 report term for the given list of diagnostics tool specifications. Each specification must be a tool_spec(Tool, Target, Options) term.

Compilation flags:
static

Template:
term(Specs,Term)
Mode and number of proofs:
term(+list(compound),-compound) - one

---

term/4

Returns a SARIF 2.1.0 report term for a diagnostics tool target using the given options.

Compilation flags:
    static

Template:
    term(Tool,Target,Term,Options)
Mode and number of proofs:
    term(+object_identifier,+nonvar,-compound,+list(compound)) - one

---

generate/2

Generates a SARIF 2.1.0 report for the given list of diagnostics tool specifications and sink accepted by the json library. Each specification must be a tool_spec(Tool, Target, Options) term.

Compilation flags:
    static

Template:
    generate(Specs,Sink)
Mode and number of proofs:
    generate(+list(compound),++compound) - one

---

generate/4

Generates a SARIF 2.1.0 report for a diagnostics tool target using the given options and sink accepted by the json library.

Compilation flags:
    static

Template:

generate(Tool,Target,Sink,Options)
Mode and number of proofs:
generate(+object_identifier,+nonvar,++compound,+list(compound)) - one

---

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.201 sbom

object

1.201.1 sbom

This tool generates a Software Bill of Materials (SBOM) for an application.

Availability:
logtalk_load(sbom(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-02

Compilation flags:
static, context_switching_calls

Imports:
public options
Uses:
json
json_schema
list

os
packs
term_io
url(Representation)
user
uuid

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 default_option/1 default_options/1 option/2 option/3 valid_option/1 valid_options/1

- Public predicates
  - document/2
  - document/1
  - export/2
  - export/1
- Protected predicates
- Private predicates
  - spdx_license_schema_/1
- Operators

Public predicates

document/2

Returns an SPDX 2.3 or a CycloneDX 1.6 JSON term describing the currently loaded application using the given options. The JSON term represents objects using curly terms, pairs using a dash, and strings using atoms.

Compilation flags:
static

Template:
document(Document,Options)
Mode and number of proofs:
document(-compound,+list(compound)) - one

---

document/1

Returns an SPDX 2.3 JSON term describing the currently loaded application using default options. The JSON term represents objects using curly terms, pairs using a dash, and strings using atoms.

Compilation flags:
    static

Template:
    document(Document)
Mode and number of proofs:
    document(-compound) - one

---

export/2

Exports an SPDX 2.3 or a CycloneDX 1.6 JSON document describing the currently loaded application to the specified sink using the given options. Valid sinks are codes(List), stream(Stream), file(Path), chars(List), and atom(Atom).

Compilation flags:
    static

Template:
    export(Sink,Options)
Mode and number of proofs:
    export(++compound,+list(compound)) - one

---

export/1

Exports an SPDX 2.3 JSON document describing the currently loaded application to the specified sink using default options. Valid sinks are codes(List), stream(Stream), file(Path), chars(List), and atom(Atom).

Compilation flags:
    static

Template:
    export(Sink)
Mode and number of proofs:
    export(++compound) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

spdx_license_schema_/1

Caches the parsed schema used to validate SPDX license identifiers for CycloneDX exports.

Compilation flags:
    dynamic

Template:
    spdx_license_schema_(Schema)
Mode and number of proofs:
    spdx_license_schema_(-term) - zero_or_one

Operators

(none)

1.202 schulze_ranker

object

1.202.1 schulze_ranker

Schulze pairwise preference ranker. Learns one deterministic score per item from a dataset object implementing the pairwise_ranking_dataset_protocol protocol by computing strongest paths over aggregated head-to-head outcomes and returns a self-describing ranker term with diagnostics that can be used for ranking and export.

Availability:
logtalk_load(schulze_ranker(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-07

Compilation flags:
static, context_switching_calls

Imports:
public ranking_dataset_common
public score_ranker_model_common
public condorcet_victory_common
Uses:
avltree
list

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_ranker/1 default_option/1 default_options/1
diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 grouped_dataset_groups/2
grouped_dataset_items/2 grouped_dataset_items/3 grouped_dataset_summary/2 learn/2
learn/3 option/2 option/3 pairwise_dataset_connected_components/2
pairwise_dataset_declared_items/2 pairwise_dataset_items/2 pairwise_dataset_matchups/2
pairwise_dataset_preferences/2 pairwise_dataset_summary/2 pairwise_dataset_win_totals/2
pairwise_measurement_dataset_connected_components/2
pairwise_measurement_dataset_declared_items/2 pairwise_measurement_dataset_items/2
pairwise_measurement_dataset_measurements/2 pairwise_measurement_dataset_summary/2
print_ranker/1 rank/3 ranker_options/2 scores/2
temporal_pairwise_dataset_connected_components/2
temporal_pairwise_dataset_declared_items/2 temporal_pairwise_dataset_games/2
temporal_pairwise_dataset_games/3 temporal_pairwise_dataset_items/2
temporal_pairwise_dataset_periods/2 temporal_pairwise_dataset_summary/2 valid_option/1
valid_options/1 valid_ranker/1 validate_grouped_dataset/1 validate_grouped_dataset/2
validate_pairwise_dataset/1 validate_pairwise_dataset/2

validate_pairwise_measurement_dataset/1 validate_pairwise_measurement_dataset/2
validate_temporal_pairwise_dataset/1 validate_temporal_pairwise_dataset/2

- Public predicates
  - strongest_paths/2
- Protected predicates
- Private predicates
- Operators

Public predicates

strongest_paths/2

Returns the labeled strongest-path relation as path(Item1,Item2,Strength) terms for all ordered pairs of distinct learned items.

Compilation flags:
static

Template:
strongest_paths(Ranker,StrongestPaths)
Mode and number of proofs:
strongest_paths(+compound,-list(compound)) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

pairwise_ranking_dataset_protocol, ranker_protocol, copeland_ranker, borda_ranker

1.203 sequential_pattern_mining_protocols

protocol

1.203.1 sequence_dataset_protocol

Protocol for ordered sequence datasets used with sequential pattern mining algorithms. Sequence identifiers are expected to be unique within a dataset.

Availability:
logtalk_load(sequential_pattern_mining_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-29

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - items/1
  - sequence/2

- Protected predicates
- Private predicates
- Operators

Public predicates

items/1

Returns the declared sequence item domain as a canonical sorted list of unique items.

Compilation flags:
    static

Template:
    items(Items)
Mode and number of proofs:
    items(-list(atom)) - one

---

sequence/2

Enumerates by backtracking the sequences in the dataset. The sequence identifier is expected to be unique within the dataset. Each sequence is represented as an ordered list of canonical sorted itemsets with unique declared items.

Compilation flags:
    static

Template:
    sequence(Id,Sequence)
Mode and number of proofs:
    sequence(-integer,-list(list(atom))) - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

category

1.203.2 sequential_pattern_mining_common

Shared predicates for sequential pattern miner dataset validation, support counting, and pattern ordering helpers.

Availability:
logtalk_load(sequential_pattern_mining_protocols(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-29

Compilation flags:
static

Extends:
public pattern_miner_common
Uses:
list
type

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_pattern_miner/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 mine/2 mine/3 option/2 option/3 pattern_miner_options/2 print_pattern_miner/1 valid_option/1 valid_options/1 valid_pattern_miner/1

- Public predicates
- Protected predicates
  - check_sequences/4
  - pattern_length/2
  - sort_patterns/2
  - filter_patterns/3
  - count_items/4
  - select_frequent_item_supports/3
  - sort_item_supports/2
  - valid_sequence_patterns/2
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

check_sequences/4

Validates the collected dataset sequences and returns the maximum total sequence length.

Compilation flags:
    static

Template:
    check_sequences(Dataset,ItemDomain,Sequences,MaxSequenceLength)
Mode and number of proofs:
    check_sequences(+object_identifier,+list(atom),+list(pair(integer,list(list(atom)))),-integer) - one

pattern_length/2

Computes the total number of items in a sequential pattern.

Compilation flags:
    static

Template:
    pattern_length(Pattern,PatternLength)
Mode and number of proofs:
    pattern_length(+list(list(atom)),-integer) - one

---

sort_patterns/2

Sorts sequence patterns first by total item count and then lexicographically.

Compilation flags:
    static

Template:
    sort_patterns(Patterns0,Patterns)
Mode and number of proofs:
    sort_patterns(+list(compound),-list(compound)) - one

---

filter_patterns/3

Filters sequence patterns by minimum total item count.

Compilation flags:
    static

Template:
    filter_patterns(Patterns0,MinimumPatternLength,Patterns)
Mode and number of proofs:
    filter_patterns(+list(compound),+integer,-list(compound)) - one

---

count_items/4

Accumulates support counts for a list of items.

Compilation flags:
    static

Template:
    count_items(Items,Count,ItemCounts0,ItemCounts)
Mode and number of proofs:
    count_items(+list(atom),+integer,+list(pair(atom,integer)),-list(pair(atom,integer))) - one

---

select_frequent_item_supports/3

Selects the item supports that satisfy the minimum support count.

Compilation flags:
    static

Template:
    select_frequent_item_supports(ItemCounts,SupportCount,FrequentItemSupports)
Mode and number of proofs:
    select_frequent_item_supports(+list(pair(atom,integer)),+integer,-list(compound)) - one

---

sort_item_supports/2

Sorts item supports lexicographically by item.

Compilation flags:
    static

Template:
    sort_item_supports(ItemSupports0,ItemSupports)
Mode and number of proofs:
    sort_item_supports(+list(compound),-list(compound)) - one

---

valid_sequence_patterns/2

True when the patterns are valid sequence_pattern(Pattern, Support) terms over the given item domain with positive integer supports.

Compilation flags:
    static

Template:
    valid_sequence_patterns(ItemDomain,Patterns)
Mode and number of proofs:
    valid_sequence_patterns(+list(atom),+list(compound)) - zero_or_one

---

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.204 sets

object

1.204.1 set

Set predicates implemented using ordered lists. Uses ==/2 for element comparison and standard term ordering.

Availability:
    logtalk_load(sets(loader))

Author: Richard O'Keefe (main predicates); adapted to Logtalk by Paulo Moura.
Version: 2:0:0
Date: 2025-07-08

Compilation flags:
    static, context_switching_calls

Implements:
public setp
Extends:
public compound
Aliases:
setp size/2 as length/2

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 as_list/2 as_set/2 check/1 delete/3 depth/2 disjoint/2 empty/1 equal/2 ground/1 insert/3 insert_all/3 intersect/2 intersection/3 intersection/4 member/2 memberchk/2 new/1 numbervars/1 numbervars/3 occurs/2 powerset/2 product/3 select/3 selectchk/3 singletons/2 size/2 subset/2 subsumes/2 subterm/2 subtract/3 symdiff/3 union/3 union/4 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

set(Type), treap_set

object

1.204.2 set(Type)

Set predicates with elements constrained to a single type and custom comparing rules.

Availability:
logtalk_load(sets(loader))

Author: Paulo Moura and Adrian Arroyo
Version: 1:24:0
Date: 2022-02-03

Compilation flags:
static, context_switching_calls

Extends:
public set
Uses:
list

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 as_list/2 as_set/2 check/1 delete/3 depth/2 disjoint/2 empty/1 equal/2 ground/1 insert/3 insert_all/3 intersect/2 intersection/3 intersection/4 member/2 memberchk/2 new/1 numbervars/1 numbervars/3 occurs/2 powerset/2 product/3 select/3 selectchk/3 singletons/2 size/2 subset/2 subsumes/2 subterm/2 subtract/3 symdiff/3 union/3 union/4 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
- Protected predicates

- Private predicates
  - sort/2
  - partition/4
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

sort/2

Sorts a list in ascending order.

Compilation flags:
    static

Template:
    sort(List,Sorted)
Mode and number of proofs:
    sort(+list,-list) - one

---

partition/4

List partition in two sub-lists using a pivot.

Compilation flags:
    static

Template:
    partition(List,Pivot,Lowers,Biggers)
Mode and number of proofs:
    partition(+list,+nonvar,-list,-list) - one

Operators

(none)

See also

set, treap_set

protocol

1.204.3 setp

Set protocol.

Availability:
logtalk_load(sets(loader))

Author: Paulo Moura
Version: 2:0:0
Date: 2025-07-08

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - as_set/2
  - as_list/2
  - delete/3

  - disjoint/2
  - equal/2
  - empty/1
  - insert/3
  - insert_all/3
  - intersect/2
  - intersection/3
  - intersection/4
  - size/2
  - member/2
  - memberchk/2
  - powerset/2
  - product/3
  - select/3
  - selectchk/3
  - subset/2
  - subtract/3
  - symdiff/3
  - union/3
  - union/4
- Protected predicates
- Private predicates
- Operators

Public predicates

as_set/2

Returns a set with all unique elements from the given list.

Compilation flags:
    static

Template:
    as_set(List,Set)
Mode and number of proofs:
    as_set(@list,-set) - one

---

as_list/2

Returns a list with all elements of the given set.

Compilation flags:
    static

Template:
    as_list(Set,List)
Mode and number of proofs:
    as_list(@set,-list) - one

---

delete/3

Deletes an element from a set returning the set of the remaining elements.

Compilation flags:
    static

Template:
    delete(Set,Element,Remaining)
Mode and number of proofs:
    delete(@set,@term,-set) - one

---

disjoint/2

True when the two sets have no element in common.

Compilation flags:
    static

Template:
    disjoint(Set1,Set2)
Mode and number of proofs:

disjoint(@set,@set) - zero_or_one

---

equal/2

True when the two sets are equal.

Compilation flags:
static

Template:
equal(Set1,Set2)
Mode and number of proofs:
equal(@set,@set) - zero_or_one

---

empty/1

True when the set is empty.

Compilation flags:
static

Template:
empty(Set)
Mode and number of proofs:
empty(@set) - zero_or_one

---

insert/3

Inserts an element in a set, returning the resulting set.

Compilation flags:
static

Template:

insert(In,Element,Out)
Mode and number of proofs:
insert(@set,@term,-set) - one

---

insert_all/3

Inserts all the elements of the list into a set, returning the resulting set.

Compilation flags:
static

Template:
insert_all(List,In,Out)
Mode and number of proofs:
insert_all(@list,@set,-set) - one

---

intersect/2

True if the two sets have at least one element in common.

Compilation flags:
static

Template:
intersect(Set1,Set2)
Mode and number of proofs:
intersect(@set,@set) - zero_or_one

---

intersection/3

Computes the intersection of Set1 and Set2.

Compilation flags:
    static

Template:
    intersection(Set1,Set2,Intersection)
Mode and number of proofs:
    intersection(@set,@set,-set) - one

---

intersection/4

Computes the intersection and the difference between Set2 and Set1.

Compilation flags:
    static

Template:
    intersection(Set1,Set2,Intersection,Difference)
Mode and number of proofs:
    intersection(@set,@set,-set,-set) - one

---

size/2

Number of set elements.

Compilation flags:
    static

Template:
    size(Set,Size)
Mode and number of proofs:
    size(@set,?integer) - zero_or_one

---

member/2

Element is a member of set Set.

Compilation flags:
    static

Template:
    member(Element,Set)
Mode and number of proofs:
    member(?term,+set) - zero_or_more

---

memberchk/2

True when a term is a member of a set.

Compilation flags:
    static

Template:
    memberchk(Element,Set)
Mode and number of proofs:
    memberchk(@term,@set) - zero_or_one

---

powerset/2

Returns the power set of a set, represented as a list of sets.

Compilation flags:
    static

Template:
    powerset(Set,Powerset)
Mode and number of proofs:
    powerset(@set,-list(set)) - one

---

product/3

Returns the cartesian product of two sets.

Compilation flags:
    static

Template:
    product(Set1,Set2,Product)
Mode and number of proofs:
    product(@set,@set,-set) - one

---

select/3

Selects an element from a set, returning the set of remaining elements.

Compilation flags:
    static

Template:
    select(Element,Set,Remaining)
Mode and number of proofs:
    select(?term,?set,?set) - zero_or_more

---

selectchk/3

True if an element can be selected from a set, returning the set of remaining elements.

Compilation flags:
    static

Template:
    selectchk(Element,Set,Remaining)
Mode and number of proofs:
    selectchk(@term,@set,-set) - zero_or_one

---

subset/2

True if Subset is a subset of Set.

Compilation flags:
    static

Template:
    subset(Subset,Set)
Mode and number of proofs:
    subset(@set,@set) - zero_or_one

---

subtract/3

Computes the set of all the elements of Set1 which are not also in Set2.

Compilation flags:
    static

Template:
    subtract(Set1,Set2,Difference)
Mode and number of proofs:
    subtract(@set,@set,-set) - one

---

symdiff/3

Computes the symmetric difference of Set1 and Set2, containing all elements that are not in the sets intersection.

Compilation flags:
    static

Template:
    symdiff(Set1,Set2,Difference)
Mode and number of proofs:
    symdiff(@set,@set,-set) - one

---

union/3

Computes the union of Set1 and Set2.

Compilation flags:
    static

Template:
    union(Set1,Set2,Union)
Mode and number of proofs:
    union(@set,@set,-set) - one

---

union/4

Computes the union of Set1 and Set2 and the difference between Set2 and Set1.

Compilation flags:
    static

Template:
    union(Set1,Set2,Union,Difference)
Mode and number of proofs:
    union(@set,@set,-set,-set) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

set, set(Type)

object

1.204.4 treap_set

Set predicates implemented using treaps (tree heaps). A treap is a binary search tree with randomly assigned priorities. Uses ==/2 for element comparison and standard term ordering.

Availability:
logtalk_load(sets(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-01-28

Compilation flags:
static, context_switching_calls

Implements:
public setp
Extends:
public compound
Uses:
fast_random(Algorithm)
list
set

Remarks:
(none)

Inherited public predicates:

(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 as_list/2 as_set/2 check/1 delete/3 depth/2 disjoint/2 empty/1 equal/2 ground/1 insert/3 insert_all/3 intersect/2 intersection/3 intersection/4 member/2 memberchk/2 new/1 numbervars/1 numbervars/3 occurs/2 powerset/2 product/3 select/3 selectchk/3 singletons/2 size/2 subset/2 subsumes/2 subterm/2 subtract/3 symdiff/3 union/3 union/4 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

set, set(Type)

1.205 sgd_classifier

object

1.205.1 sgd_classifier

Stochastic gradient descent classifier supporting one-vs-rest linear models with configurable losses including log_loss, hinge, squared_hinge, modified_huber, and perceptron.

Availability:
logtalk_load(sgd_classifier(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-11

Compilation flags:
static, context_switching_calls

Imports:
public probabilistic_classifier_common
Uses:
format
linear_algebra
list
numberlist
type

Remarks:
(none)

Inherited public predicates:
check_classifier/1 check_option/1 check_options/1 classifier_options/2 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 predict_probabilities/3 print_classifier/1 valid_classifier/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

dataset_protocol, logistic_regression_classifier, linear_svm_classifier, knn_classifier, nearest_centroid_classifier

1.206 simulated_annealing

object

1.206.1 simulated_annealing(Problem)

- Problem - Problem object implementing simulated_annealing_protocol.

Simulated annealing optimization algorithm using the Xoshiro128++ random number generator. Convenience object that extends simulated_annealing/2 with the random algorithm bound to xoshiro128pp.

Availability:
logtalk_load(simulated_annealing(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-22

Compilation flags:
static, context_switching_calls

Extends:
public simulated_annealing(Problem,xoshiro128pp)

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 default_option/1 default_options/1 estimate_temperature/1
estimate_temperature/2 option/2 option/3 run/2 run/3 run/4 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

simulated_annealing(Problem,RandomAlgorithm), simulated_annealing_protocol

object

1.206.2 simulated_annealing(Problem,RandomAlgorithm)

- Problem - Problem object implementing simulated_annealing_protocol.
- RandomAlgorithm - Random number generator algorithm for the fast_random library (e.g. xoshiro128pp, xoshiro256ss, well512a, …).

Simulated annealing optimization algorithm. Parameterized by a problem object implementing the simulated_annealing_protocol protocol and by a random number generator algorithm for the fast_random library. The algorithm minimizes the energy (cost) function defined by the problem. Custom cooling schedules, stop conditions, delta-energy neighbor generation, progress reporting, and reheating restarts can be defined by the problem object or configured via options; suitable defaults are used otherwise.

Availability:
logtalk_load(simulated_annealing(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-23

Compilation flags:
static, context_switching_calls

Imports:
public options
Uses:
fast_random(Algorithm)
type

Remarks:

- Algorithm: Simulated annealing is a probabilistic metaheuristic for global optimization. It explores the solution space by iteratively generating neighbor states and accepting them based on an energy-dependent probability that decreases over time as the temperature cools.
- Acceptance criterion: Uses the standard Boltzmann acceptance criterion: a worse neighbor is accepted with probability exp(-DeltaE / Temperature).
- Default cooling schedule: Geometric cooling: NewTemp is Temp * 0.995. Override by defining cooling_schedule/3 in the problem object.
- Default stop condition: The search stops when the maximum number of steps is reached or the temperature drops below the minimum temperature. Override by defining stop_condition/3 in the problem object.
- Delta-energy optimization: If the problem object defines neighbor_state/3, the algorithm uses the returned delta energy directly instead of calling state_energy/2 on the neighbor. This is useful when computing the energy change is cheaper than recomputing the full energy.

- Progress reporting: If the problem object defines progress/5, it is called periodically with the current step, temperature, best energy, acceptance rate, and improvement rate. The reporting interval is controlled by the updates(N) option. A final report is always produced when the loop terminates.
- Best state tracking: The algorithm tracks the best state found across all iterations and across all restart cycles, not just the final state.
- Seed control: The seed(S) option initializes the random number generator for reproducible runs.
- Reheating restart: The restarts(N) option runs N additional SA cycles after the first. Each restart reheats the temperature to the initial value and begins from the best state found so far, allowing the search to escape local minima. Statistics accumulate across all cycles.
- Auto-temperature estimation: The estimate_temperature/1-2 predicates sample random neighbor transitions and compute an initial temperature that would produce a target acceptance rate. This avoids manual tuning of the initial temperature.

Inherited public predicates:
check_option/1 check_options/1 default_option/1 default_options/1 option/2 option/3 valid_option/1 valid_options/1

- Public predicates
  - estimate_temperature/1
  - estimate_temperature/2
  - run/2
  - run/3
  - run/4
- Protected predicates
- Private predicates
- Operators

Public predicates

estimate_temperature/1

Estimates an initial temperature for the problem using default options (200 samples, 80% target acceptance rate).

Compilation flags:
static

Template:
estimate_temperature(Temperature)
Mode and number of proofs:

estimate_temperature(-float) - one

---

estimate_temperature/2

Estimates an initial temperature by sampling random neighbor transitions. The temperature is computed so that the Boltzmann acceptance probability for the average uphill move equals the target acceptance rate. Sampling starts from the problem initial state and follows a random walk.

Compilation flags:
static

Template:
estimate_temperature(Temperature,Options)
Mode and number of proofs:
estimate_temperature(-float,+list(compound)) - one

Remarks:

- samples(N) option: Number of random neighbor transitions to sample (default: 200).
- acceptance_rate(P) option: Target initial acceptance rate as an integer percentage between 1 and 99 (default: 80).

---

run/2

Runs the simulated annealing algorithm using default options and returns the best state found and its energy.

Compilation flags:
static

Template:
run(BestState,BestEnergy)
Mode and number of proofs:
run(-term,-number) - one

---

run/3

Runs the simulated annealing algorithm using the given options and returns the best state found and its energy.

Compilation flags:
    static

Template:
    run(BestState,BestEnergy,Options)
Mode and number of proofs:
    run(-term,-number,+list(compound)) - one

Remarks:

- max_steps(N) option: Maximum number of iterations per cycle (default: 10000).
- min_temperature(T) option: Minimum temperature floor; search stops when temperature drops below this value (default: 0.001).
- updates(N) option: Number of progress reports during the run. Set to 0 to disable. Progress is reported by calling progress/5 on the problem object (default: 0).
- seed(S) option: Positive integer seed for the random number generator, enabling reproducible runs (default: none).
- restarts(N) option: Number of additional SA cycles after the first. Each restart reheats the temperature and begins from the best state found so far (default: 0).

---

run/4

Runs the simulated annealing algorithm using the given options, returns the best state found and its energy, and returns run statistics.

Compilation flags:
    static

Template:
    run(BestState,BestEnergy,Statistics,Options)
Mode and number of proofs:
    run(-term,-number,-list(compound),+list(compound)) - one

Remarks:

- Statistics list: A list of Key(Value) pairs: steps(N) is the number of steps executed, acceptances(A) is the number of accepted moves, improvements(I) is the number of moves that improved the best energy, and final_temperature(T) is the temperature at termination.

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

simulated_annealing(Problem), simulated_annealing_protocol

protocol

1.206.3 simulated_annealing_protocol

Protocol for simulated annealing problem definitions. A problem object must define the four required predicates and may optionally define predicates to override cooling, stopping, neighbor generation with delta energy, and progress reporting defaults.

Availability:
logtalk_load(simulated_annealing(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-22

Compilation flags:
static

Dependencies:
(none)

Remarks:

(none)

Inherited public predicates:
(none)

- Public predicates
  - initial_state/1
  - neighbor_state/2
  - neighbor_state/3
  - state_energy/2
  - initial_temperature/1
  - cooling_schedule/3
  - stop_condition/3
  - progress/5
- Protected predicates
- Private predicates
- Operators

Public predicates

initial_state/1

Returns an initial state for the optimization problem.

Compilation flags:
static

Template:
initial_state(State)
Mode and number of proofs:
initial_state(-term) - one

neighbor_state/2

Generates a neighboring state from the given state. This is the most problem-specific predicate and its definition determines the quality of the search.

Compilation flags:
    static

Template:
    neighbor_state(State,Neighbor)
Mode and number of proofs:
    neighbor_state(+term,-term) - one

---

neighbor_state/3

Generates a neighboring state and returns the energy change (delta) directly, avoiding a full energy recomputation. Optional. When not defined by the problem, the algorithm calls neighbor_state/2 and state_energy/2 instead.

Compilation flags:
    static

Template:
    neighbor_state(State,Neighbor,DeltaEnergy)
Mode and number of proofs:
    neighbor_state(+term,-term,-number) - one

---

state_energy/2

Computes the energy (cost) of the given state. The algorithm minimizes this value.

Compilation flags:
    static

Template:
    state_energy(State,Energy)
Mode and number of proofs:

state_energy(+term,-number) - one

---

initial_temperature/1

Returns the initial temperature for the annealing schedule. Higher temperatures increase the probability of accepting worse solutions early on.

Compilation flags:
static

Template:
initial_temperature(Temperature)
Mode and number of proofs:
initial_temperature(-number) - one

---

cooling_schedule/3

Computes the next temperature from the current temperature and step number. Optional. When not defined by the problem, a default geometric cooling schedule is used (NewTemp is Temp * 0.995).

Compilation flags:
static

Template:
cooling_schedule(Temperature,Step,NewTemperature)
Mode and number of proofs:
cooling_schedule(+number,+non_negative_integer,-number) - one

---

stop_condition/3

True when the search should stop given the current step, temperature, and best energy found so far. Optional. When not defined by the problem, the search runs until the maximum number of steps is reached or the minimum temperature is reached.

Compilation flags:
    static

Template:
    stop_condition(Step,Temperature,BestEnergy)
Mode and number of proofs:
    stop_condition(+non_negative_integer,+number,+number) - zero_or_one

---

progress/5

Called periodically to report optimization progress. Optional. When not defined by the problem, progress reporting is skipped. The acceptance and improvement rates are values between 0.0 and 1.0 computed over the interval since the last progress report.

Compilation flags:
    static

Template:
    progress(Step,Temperature,BestEnergy,AcceptanceRate,ImprovementRate)
Mode and number of proofs:
    progress(+non_negative_integer,+number,+number,+number,+number) - zero_or_one

---

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

simulated_annealing(Problem)

1.207 snowflakeid

object

1.207.1 snowflakeid

Snowflake ID generator using the Twitter-style profile and atom representation.

Availability:
logtalk_load(snowflakeid(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-26

Compilation flags:
static, context_switching_calls

Extends:
public snowflakeid_twitter(atom)

Remarks:
(none)

Inherited public predicates:
generate/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

snowflakeid(Representation,EpochMilliseconds,TimeUnitMilliseconds,TimestampBits,NodeBits,SequenceBits,Node), snowflakeid_twitter, snowflakeid_sonyflake, snowflakeid_instagram

object

1.207.2 snowflakeid(Representation,EpochMilliseconds,TimeUnitMilliseconds, TimestampBits,NodeBits,SequenceBits,Node)

- Representation - Snowflake ID representation. Possible values are integer, atom, chars, and codes.
- EpochMilliseconds - Custom epoch in Unix milliseconds.
- TimeUnitMilliseconds - Timestamp unit in milliseconds.
- TimestampBits - Number of timestamp bits.
- NodeBits - Number of node bits.
- SequenceBits - Number of sequence bits.
- Node - Node identifier value.

Snowflake ID generic generator.

Availability:

logtalk_load(snowflakeid(loader))

Author: Paulo Moura
Version: 1:0:1
Date: 2026-03-13

Compilation flags:
static, context_switching_calls

Implements:
public snowflakeid_protocol
Uses:
iso8601
os

Remarks:
(none)

Inherited public predicates:
generate/1

- Public predicates
- Protected predicates
- Private predicates
  - last_time_sequence_/2
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

last_time_sequence_/2

Last time and sequence.

Compilation flags:
dynamic

Template:
last_time_sequence_(Time,Sequence)
Mode and number of proofs:
last_time_sequence_(?integer,?integer) - zero_or_one

Operators

(none)

See also

snowflakeid, snowflakeid_twitter, snowflakeid_sonyflake, snowflakeid_instagram, ksuid(Representation,Alphabet), cuid2(Representation,Size,Alphabet), nanoid(Representation,Size,Alphabet)

object

1.207.3 snowflakeid_instagram

Instagram-style Snowflake profile using atom representation.

Availability:
logtalk_load(snowflakeid(loader))

Author: Paulo Moura

Version: 1:0:0
Date: 2026-02-26

Compilation flags:
static, context_switching_calls

Extends:
public snowflakeid_instagram(atom)

Remarks:
(none)

Inherited public predicates:
generate/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

snowflakeid_instagram(Representation), snowflakeid_twitter, snowflakeid_sonyflake

object

1.207.4 snowflakeid_instagram(Representation)

- Representation - Snowflake ID representation. Possible values are integer, atom, chars, and codes.

Instagram-style Snowflake profile.

Availability:
logtalk_load(snowflakeid(loader))

Author: Paulo Moura
Version: 1:0:1
Date: 2026-02-27

Compilation flags:
static, context_switching_calls

Extends:
public snowflakeid(Representation,1314220021721,1,41,13,10,1)

Remarks:
(none)

Inherited public predicates:
generate/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

snowflakeid(Representation,EpochMilliseconds,TimeUnitMilliseconds,TimestampBits,NodeBits,SequenceBits,Node), snowflakeid_twitter, snowflakeid_sonyflake, snowflakeid_instagram

protocol

1.207.5 snowflakeid_protocol

Snowflake ID generator protocol.

Availability:
logtalk_load(snowflakeid(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-26

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - generate/1
- Protected predicates
- Private predicates
- Operators

Public predicates

generate/1

Returns a Snowflake ID.

Compilation flags:
static

Template:
generate(ID)
Mode and number of proofs:
generate(--ground) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

object

1.207.6 snowflakeid_sonyflake

Sonyflake-style Snowflake profile using atom representation.

Availability:
logtalk_load(snowflakeid(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-26

Compilation flags:
static, context_switching_calls

Extends:
public snowflakeid_sonyflake(atom)

Remarks:
(none)

Inherited public predicates:
generate/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

snowflakeid_sonyflake(Representation), snowflakeid_twitter, snowflakeid_instagram

object

1.207.7 snowflakeid_sonyflake(Representation)

- Representation - Snowflake ID representation. Possible values are integer, atom, chars, and codes.

Sonyflake-style Snowflake profile.

Availability:
logtalk_load(snowflakeid(loader))

Author: Paulo Moura
Version: 1:0:1
Date: 2026-02-27

Compilation flags:
static, context_switching_calls

Extends:
public snowflakeid(Representation,1409529600000,10,39,16,8,1)

Remarks:
(none)

Inherited public predicates:
generate/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

snowflakeid(Representation,EpochMilliseconds,TimeUnitMilliseconds,TimestampBits,NodeBits,SequenceBits,Node), snowflakeid_twitter, snowflakeid_sonyflake, snowflakeid_instagram

object

1.207.8 snowflakeid_twitter

Twitter-style Snowflake profile using atom representation.

Availability:
logtalk_load(snowflakeid(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-26

Compilation flags:
static, context_switching_calls

Extends:
public snowflakeid_twitter(atom)

Remarks:
(none)

Inherited public predicates:
generate/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

snowflakeid_twitter(Representation), snowflakeid_sonyflake, snowflakeid_instagram

object

1.207.9 snowflakeid_twitter(Representation)

- Representation - Snowflake ID representation. Possible values are integer, atom, chars, and codes.

Twitter-style Snowflake profile.

Availability:
logtalk_load(snowflakeid(loader))

Author: Paulo Moura
Version: 1:0:1
Date: 2026-02-27

Compilation flags:
static, context_switching_calls

Extends:
public snowflakeid(Representation,1288834974657,1,41,10,12,1)

Remarks:
(none)

Inherited public predicates:
generate/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

snowflakeid(Representation,EpochMilliseconds,TimeUnitMilliseconds,TimestampBits,NodeBits,SequenceBits,Node), snowflakeid_twitter, snowflakeid_sonyflake, snowflakeid_instagram

1.208 sockets

object

1.208.1 socket

Portable abstraction over TCP sockets. Provides a high-level API for client and server socket operations that works with selected backend Prolog systems.

Availability:
logtalk_load(sockets(loader))

Author: Paulo Moura
Version: 0:13:0
Date: 2026-04-22

Compilation flags:
static, context_switching_calls

Imports:
public options
Uses:
list

Remarks:

- Supported backends: ECLiPSe, GNU Prolog, SICStus Prolog, SWI-Prolog, and Trealla Prolog.
- Design rationale: Some backends (notably SICStus Prolog) do not provide low-level socket creation predicates that can be separated from binding or connecting. This library therefore provides a higher-level API with client_open/5 and server_open/3 that abstracts over these differences.
- Stream handling: Predicates client_open/5 and server_accept/5 return separate input and output streams opened in binary mode. For backends where the same stream is used for bidirectional communication, the same stream handle is returned in both arguments. Use standard stream predicates (put_byte/2, get_byte/2, read/2, write/2, etc.) to communicate.
- Options: Currently options are only defined for the server_open/3 predicate. This is expected to change in future versions for the other predicates that have an options argument.

Inherited public predicates:

check_option/1 check_options/1 default_option/1 default_options/1 option/2 option/3 valid_option/1 valid_options/1

- Public predicates
  - client_open/5
  - client_open/4
  - server_open/4
  - server_open/3
  - server_open/2
  - server_accept/5
  - server_accept/4
  - server_close/1
  - close/2
  - current_host/1
- Protected predicates
- Private predicates
- Operators

Public predicates

client_open/5

Opens a client connection to the specified host and port using the given options. Returns separate input and output streams for bidirectional communication. The streams are opened by default in binary mode.

Compilation flags:

static

Template:
client_open(Host,Port,InputStream,OutputStream,Options)
Mode and number of proofs:
client_open(+atom,+integer,--stream,--stream,+list) - one_or_error

Exceptions:
Connection refused or host not found:
socket_error(Error)

Remarks:

- Option type(binary): Open the streams in binary mode. This is the default.
- Option type(text): Open the streams in text mode.

---

client_open/4

Opens a client connection to the specified host and port using default options. Returns separate input and output streams for bidirectional communication. The streams are opened in binary mode.

Compilation flags:
static

Template:
client_open(Host,Port,InputStream,OutputStream)
Mode and number of proofs:
client_open(+atom,+integer,--stream,--stream) - one_or_error

Exceptions:
Connection refused or host not found:
socket_error(Error)

---

server_open/4

Opens a server socket bound to the specified host and port using the given options. If Port is a variable, binds to an available port and unifies Port with the port number. Returns a ServerSocket handle to use with server_accept/4. The default backlog (queue length) for pending connections is 5. Use the option backlog(N) to override. This option is not supported and thus ignored by the SICStus Prolog and Trealla Prolog backends.

Compilation flags:
    static

Template:
    server_open(Host,Port,ServerSocket,Options)
Mode and number of proofs:
    server_open(+atom,?integer,--compound,+list) - one_or_error

Exceptions:
    Port already in use:
        socket_error(Error)

---

server_open/3

Opens a server socket bound to the specified port using the given options. If Port is a variable, binds to an available port and unifies Port with the port number. Returns a ServerSocket handle to use with server_accept/4. The default backlog (queue length) for pending connections is 5. Use the option backlog(N) to override. This option is not supported and thus ignored by the SICStus Prolog and Trealla Prolog backends.

Compilation flags:
    static

Template:
    server_open(Port,ServerSocket,Options)
Mode and number of proofs:
    server_open(?integer,--compound,+list) - one_or_error

Exceptions:
    Port already in use:
        socket_error(Error)

---

server_open/2

Opens a server socket bound to the specified port using default options. If Port is a variable, binds to an available port and unifies Port with the port number. Returns a ServerSocket handle to use with server_accept/4. The default backlog (queue length) for pending connections is 5.

Compilation flags:
    static

Template:
    server_open(Port,ServerSocket)
Mode and number of proofs:
    server_open(?integer,--compound) - one_or_error

Exceptions:
    Port already in use:
        socket_error(Error)

---

server_accept/5

Accepts an incoming connection on the server socket, blocking until a client connects, using the given options. Returns separate input and output streams for bidirectional communication and client information as client(Host, Port) or client(Address) depending on backend. The streams are opened by default in binary mode.

Compilation flags:
    static

Template:
    server_accept(ServerSocket,InputStream,OutputStream,ClientInfo,Options)
Mode and number of proofs:
    server_accept(+compound,--stream,--stream,--compound,+list) - one_or_error

Exceptions:
    Invalid server socket:
        socket_error(Error)

Remarks:

- Option type(binary): Open the streams in binary mode. This is the default.

- Option type(text): Open the streams in text mode.

---

server_accept/4

Accepts an incoming connection on the server socket, blocking until a client connects, using default options. Returns separate input and output streams for bidirectional communication and client information as client(Host, Port) or client(Address) depending on backend. The streams are opened in binary mode.

Compilation flags:
    static

Template:
    server_accept(ServerSocket,InputStream,OutputStream,ClientInfo)
Mode and number of proofs:
    server_accept(+compound,--stream,--stream,--compound) - one_or_error

Exceptions:
    Invalid server socket:
        socket_error(Error)

---

server_close/1

Closes a server socket.

Compilation flags:
    static

Template:
    server_close(ServerSocket)
Mode and number of proofs:
    server_close(+compound) - one_or_error

---

close/2

Closes a client or accepted connection by closing both the input and output streams. If the same stream is used for both, it is closed only once.

Compilation flags:
    static

Template:
    close(InputStream,OutputStream)
Mode and number of proofs:
    close(+stream,+stream) - one_or_error

---

current_host/1

Returns the hostname of the current machine.

Compilation flags:
    static

Template:
    current_host(Host)
Mode and number of proofs:
    current_host(-atom) - one_or_error

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.209 spade_pattern_miner

object

1.209.1 spade_pattern_miner

SPADE sequential pattern miner for sequence datasets using vertical occurrence lists.

Availability:
logtalk_load(spade_pattern_miner(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public sequential_pattern_mining_common
Uses:
avltree
format
list

Remarks:
(none)

Inherited public predicates:
check_option/1 check_options/1 check_pattern_miner/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 mine/2 mine/3 option/2 option/3 pattern_miner_options/2 print_pattern_miner/1 valid_option/1 valid_options/1 valid_pattern_miner/1

- Public predicates
- Protected predicates

- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

pattern_miner_protocol, sequence_dataset_protocol, prefix_span_pattern_miner, gsp_pattern_miner, clo_span_pattern_miner

1.210 statistics

object

1.210.1 population

Statistical population represented as a list of numbers.

Availability:
logtalk_load(statistics(loader))

Author: Paulo Moura
Version: 1:4:0
Date: 2026-02-20

Compilation flags:
static, context_switching_calls

Imports:
public statistics

Remarks:
(none)

Inherited public predicates:
arithmetic_mean/2 average_deviation/3 central_moment/3 coefficient_of_variation/2 correlation/3 covariance/3 fractile/3 frequency_distribution/2 geometric_mean/2 harmonic_mean/2 interquartile_range/2 kurtosis/2 max/2 mean_deviation/2 mean_squared_error/3 median/2 median_deviation/2 min/2 min_max/3 min_max_normalization/2 modes/2 percentile/3 product/2 quartiles/4 range/2 rank_correlation/3 relative_standard_deviation/2 root_mean_squared_error/3 skewness/2 standard_deviation/2 standard_error/2 sum/2 sum_of_squares/2 trimmed_mean/3 valid/1 variance/2 weighted_mean/3 z_normalization/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

sample

object

1.210.2 sample

Statistical sample represented as a list of numbers.

Availability:
logtalk_load(statistics(loader))

Author: Paulo Moura
Version: 1:5:0
Date: 2026-02-20

Compilation flags:
static, context_switching_calls

Imports:
public statistics

Remarks:
(none)

Inherited public predicates:
arithmetic_mean/2 average_deviation/3 central_moment/3 coefficient_of_variation/2 correlation/3 covariance/3 fractile/3 frequency_distribution/2 geometric_mean/2 harmonic_mean/2 interquartile_range/2 kurtosis/2 max/2 mean_deviation/2 mean_squared_error/3 median/2 median_deviation/2 min/2 min_max/3 min_max_normalization/2 modes/2 percentile/3 product/2 quartiles/4 range/2 rank_correlation/3 relative_standard_deviation/2 root_mean_squared_error/3 skewness/2 standard_deviation/2 standard_error/2 sum/2 sum_of_squares/2 trimmed_mean/3 valid/1 variance/2 weighted_mean/3 z_normalization/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

population

category

1.210.3 statistics

Statistical calculations over a list of numbers.

Availability:
logtalk_load(statistics(loader))

Author: Paulo Moura
Version: 1:8:0
Date: 2026-02-20

Compilation flags:
static

Implements:
public statisticsp
Uses:
list
numberlist

Remarks:

(none)

Inherited public predicates:

arithmetic_mean/2 average_deviation/3 central_moment/3 coefficient_of_variation/2 correlation/3 covariance/3 fractile/3 frequency_distribution/2 geometric_mean/2 harmonic_mean/2 interquartile_range/2 kurtosis/2 max/2 mean_deviation/2 mean_squared_error/3 median/2 median_deviation/2 min/2 min_max/3 min_max_normalization/2 modes/2 percentile/3 product/2 quartiles/4 range/2 rank_correlation/3 relative_standard_deviation/2 root_mean_squared_error/3 skewness/2 standard_deviation/2 standard_error/2 sum/2 sum_of_squares/2 trimmed_mean/3 valid/1 variance/2 weighted_mean/3 z_normalization/2

- Public predicates
- Protected predicates
- Private predicates
  - arithmetic_mean/5
  - squares_and_cubes/6
  - squares_and_hypers/6
  - variance/6
  - cross_deviation_sum/6
  - sorted_median/3
  - sum_of_squares/4
  - central_moment_sum/7
  - min_max_normalize/4
  - count_frequencies/4
  - squared_error_sum/6
  - compute_ranks/3
  - count_less_equal/6
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

arithmetic_mean/5

Auxiliary predicate for computing the arithmetic mean.

Compilation flags:
    static

Template:
    arithmetic_mean(List,Length0,Length,Sum,Mean)
Mode and number of proofs:
    arithmetic_mean(+list(number),+integer,-integer,+number,-float) - one

---

squares_and_cubes/6

Auxiliary predicate for computing the skewness.

Compilation flags:
    static

Template:
    squares_and_cubes(List,Mean,Squares0,Squares,Cubes0,Cubes)
Mode and number of proofs:
    squares_and_cubes(+list(number),+float,+float,-float,+float,-float) - one

---

squares_and_hypers/6

Auxiliary predicate for computing the kurtosis.

Compilation flags:
    static

Template:
    squares_and_hypers(List,Mean,Squares0,Squares,Hypers0,Hypers)
Mode and number of proofs:
    squares_and_hypers(+list(number),+float,+float,-float,+float,-float) - one

---

variance/6

Auxiliary predicate for computing the variance.

Compilation flags:
    static

Template:
    variance(List,Length0,Length,Mean,M20,M2)
Mode and number of proofs:
    variance(+list(number),+integer,-integer,+float,+float,-float) - one

---

cross_deviation_sum/6

Auxiliary predicate for computing the cross-deviation sum for covariance.

Compilation flags:
    static

Template:
    cross_deviation_sum(List1,List2,Mean1,Mean2,Sum0,Sum)
Mode and number of proofs:
    cross_deviation_sum(+list(number),+list(number),+float,+float,+float,-float) - one

---

sorted_median/3

Auxiliary predicate for computing the median of an already sorted list with known length.

Compilation flags:
    static

Template:
    sorted_median(Sorted,Length,Median)
Mode and number of proofs:
    sorted_median(+list(number),+integer,-number) - one

---

sum_of_squares/4

Auxiliary predicate for computing the sum of squared deviations.

Compilation flags:
    static

Template:
    sum_of_squares(List,Mean,Sum0,Sum)
Mode and number of proofs:
    sum_of_squares(+list(number),+float,+float,-float) - one

---

central_moment_sum/7

Auxiliary predicate for computing central moments.

Compilation flags:
    static

Template:
    central_moment_sum(List,K,Mean,N0,Sum0,N,Sum)
Mode and number of proofs:
    central_moment_sum(+list(number),+positive_integer,+float,+integer,+float,-integer,-float) - one

---

min_max_normalize/4

Auxiliary predicate for min-max normalization.

Compilation flags:
    static

Template:
    min_max_normalize(List,Min,Range,NormalizedList)
Mode and number of proofs:
    min_max_normalize(+list(number),+number,+number,-list(float)) - one

---

count_frequencies/4

Auxiliary predicate for computing frequency distribution.

Compilation flags:
    static

Template:
    count_frequencies(List,CurrentValue,CurrentCount,Distribution)
Mode and number of proofs:
    count_frequencies(+list(number),+number,+integer,-list(pair(number,integer))) - one

---

squared_error_sum/6

Auxiliary predicate for computing mean squared error.

Compilation flags:
    static

Template:
    squared_error_sum(List1,List2,N0,Sum0,N,Sum)
Mode and number of proofs:
    squared_error_sum(+list(number),+list(number),+integer,+float,-integer,-float) - one

---

compute_ranks/3

Auxiliary predicate for computing ranks of elements in a list.

Compilation flags:
    static

Template:
    compute_ranks(List,All,Ranks)
Mode and number of proofs:
    compute_ranks(+list(number),+list(number),-list(float)) - one

---

count_less_equal/6

Auxiliary predicate for counting elements less than and equal to a value.

Compilation flags:
    static

Template:
    count_less_equal(List,Value,Less0,Less,Equal0,Equal)
Mode and number of proofs:
    count_less_equal(+list(number),+number,+integer,-integer,+integer,-integer) - one

---

Operators

(none)

protocol

1.210.4 statisticsp

Statistical calculations over a list of numbers protocol.

Availability:
    logtalk_load(statistics(loader))

Author: Paulo Moura

Version: 1:4:0
Date: 2026-02-20

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - product/2
  - sum/2
  - min/2
  - max/2
  - min_max/3
  - range/2
  - arithmetic_mean/2
  - geometric_mean/2
  - harmonic_mean/2
  - weighted_mean/3
  - median/2
  - modes/2
  - average_deviation/3
  - mean_deviation/2
  - median_deviation/2
  - standard_deviation/2
  - coefficient_of_variation/2
  - relative_standard_deviation/2
  - skewness/2
  - kurtosis/2

  - variance/2
  - z_normalization/2
  - fractile/3
  - percentile/3
  - quartiles/4
  - interquartile_range/2
  - covariance/3
  - correlation/3
  - rank_correlation/3
  - trimmed_mean/3
  - sum_of_squares/2
  - central_moment/3
  - min_max_normalization/2
  - frequency_distribution/2
  - standard_error/2
  - mean_squared_error/3
  - root_mean_squared_error/3
  - valid/1
- Protected predicates
- Private predicates
- Operators

Public predicates

product/2

Calculates the product of all list numbers. Fails if the list is empty.

Compilation flags:
    static

Template:
    product(List,Product)
Mode and number of proofs:
    product(+list(number),-number) - zero_or_one

sum/2

Calculates the sum of all list numbers. Fails if the list is empty.

Compilation flags:
    static

Template:
    sum(List,Sum)
Mode and number of proofs:
    sum(+list(number),-number) - zero_or_one

---

min/2

Determines the minimum value in a list of numbers. Fails if the list is empty.

Compilation flags:
    static

Template:
    min(List,Minimum)
Mode and number of proofs:
    min(+list,-number) - zero_or_one

---

max/2

Determines the list maximum value in a list of numbers. Fails if the list is empty.

Compilation flags:
    static

Template:
    max(List,Maximum)
Mode and number of proofs:
    max(+list,-number) - zero_or_one

---

min_max/3

Determines the minimum and maximum values in a list of numbers. Fails if the list is empty.

Compilation flags:
    static

Template:
    min_max(List,Minimum,Maximum)
Mode and number of proofs:
    min_max(+list(number),-number,-number) - zero_or_one

---

range/2

Range is the length of the smallest interval which contains all the numbers in List. Fails if the list is empty.

Compilation flags:
    static

Template:
    range(List,Range)
Mode and number of proofs:
    range(+list,-number) - zero_or_one

---

arithmetic_mean/2

Calculates the arithmetic mean of a list of numbers. Fails if the list is empty.

Compilation flags:
    static

Template:
    arithmetic_mean(List,Mean)
Mode and number of proofs:
    arithmetic_mean(+list(number),-float) - zero_or_one

---

geometric_mean/2

Calculates the geometric mean of a list of numbers. Fails if the list is empty.

Compilation flags:
    static

Template:
    geometric_mean(List,Mean)
Mode and number of proofs:
    geometric_mean(+list(number),-float) - zero_or_one

---

harmonic_mean/2

Calculates the harmonic mean of a list of numbers. Fails if the list is empty.

Compilation flags:
    static

Template:
    harmonic_mean(List,Mean)
Mode and number of proofs:
    harmonic_mean(+list(number),-float) - zero_or_one

---

weighted_mean/3

Calculates the weighted mean of a list of numbers. Fails if the list is empty or if the two lists have different lengths. Weights are assumed to be non-negative.

Compilation flags:
    static

Template:
    weighted_mean(Weights,List,Mean)
Mode and number of proofs:
    weighted_mean(+list(number),+list(number),-float) - zero_or_one

---

median/2

Calculates the median of a list of numbers. Fails if the list is empty.

Compilation flags:
    static

Template:
    median(List,Median)
Mode and number of proofs:
    median(+list(number),-float) - zero_or_one

---

modes/2

Returns the list of modes of a list of numbers in ascending order. Fails if the list is empty.

Compilation flags:
    static

Template:
    modes(List,Modes)
Mode and number of proofs:
    modes(+list(number),-list(number)) - zero_or_one

---

average_deviation/3

Calculates the average absolute deviation of a list of numbers given a central tendency (e.g., mean, median, or mode). Fails if the list is empty.

Compilation flags:
    static

Template:
    average_deviation(List,CentralTendency,Deviation)

Mode and number of proofs:
average_deviation(+list(number),+float,-float) - zero_or_one

---

mean_deviation/2

Calculates the mean absolute deviation of a list of numbers. Fails if the list is empty.

Compilation flags:
static

Template:
mean_deviation(List,Deviation)
Mode and number of proofs:
mean_deviation(+list(number),-float) - zero_or_one

---

median_deviation/2

Calculates the median absolute deviation of a list of numbers. Fails if the list is empty.

Compilation flags:
static

Template:
median_deviation(List,Deviation)
Mode and number of proofs:
median_deviation(+list(number),-float) - zero_or_one

---

standard_deviation/2

Calculates the standard deviation of a list of numbers. Fails if the list is empty.

Compilation flags:
static

Template:
standard_deviation(List,Deviation)
Mode and number of proofs:
standard_deviation(+list(number),-float) - zero_or_one

---

coefficient_of_variation/2

Calculates the coefficient of variation of a list of numbers. Fails if the list is empty.

Compilation flags:
static

Template:
coefficient_of_variation(List,Coefficient)
Mode and number of proofs:
coefficient_of_variation(+list(number),-float) - zero_or_one

---

relative_standard_deviation/2

Calculates the relative standard deviation of a list of numbers. Fails if the list is empty.

Compilation flags:
static

Template:
relative_standard_deviation(List,Percentage)
Mode and number of proofs:
relative_standard_deviation(+list(number),-float) - zero_or_one

---

skewness/2

Calculates the (moment) skewness of a list of numbers. Fails if the list is empty.

Compilation flags:
    static

Template:
    skewness(List,Skewness)
Mode and number of proofs:
    skewness(+list(number),-float) - zero_or_one

---

kurtosis/2

Calculates the (excess) kurtosis of a list of numbers. Fails if the list is empty.

Compilation flags:
    static

Template:
    kurtosis(List,Kurtosis)
Mode and number of proofs:
    kurtosis(+list(number),-float) - zero_or_one

---

variance/2

Calculates the unbiased variance of a list of numbers. Fails if the list is empty.

Compilation flags:
    static

Template:
    variance(List,Variance)
Mode and number of proofs:
    variance(+list(number),-float) - zero_or_one

---

z_normalization/2

Normalizes a list of number such that for the resulting list the mean of is close to zero and the standard deviation is close to 1. Fails if the list is empty.

Compilation flags:
    static

Template:
    z_normalization(List,NormalizedList)
Mode and number of proofs:
    z_normalization(+list(number),-list(float)) - zero_or_one

---

fractile/3

Calculates the smallest value in a list of numbers such that the list elements in its fraction P are less or equal to that value (with P in the open interval (0.0, 1.0)). Fails if the list is empty.

Compilation flags:
    static

Template:
    fractile(P,List,Fractile)
Mode and number of proofs:
    fractile(+float,+list(integer),-integer) - zero_or_one
    fractile(+float,+list(float),-float) - zero_or_one

---

percentile/3

Calculates the P-th percentile of a list of numbers (with P in the open interval (0, 100)). Fails if the list is empty.

Compilation flags:
    static

Template:
    percentile(P,List,Percentile)

Mode and number of proofs:
percentile(+number,+list(number),-number) - zero_or_one

---

quartiles/4

Calculates the quartiles (Q1, Q2, Q3) of a list of numbers. Q2 is the median. Q1 and Q3 are the medians of the lower and upper halves, respectively. Fails if the list has fewer than two elements.

Compilation flags:
static

Template:
quartiles(List,Q1,Q2,Q3)
Mode and number of proofs:
quartiles(+list(number),-number,-number,-number) - zero_or_one

---

interquartile_range/2

Calculates the interquartile range (Q3 - Q1) of a list of numbers. Fails if the list has fewer than two elements.

Compilation flags:
static

Template:
interquartile_range(List,IQR)
Mode and number of proofs:
interquartile_range(+list(number),-number) - zero_or_one

---

covariance/3

Calculates the covariance of two lists of numbers. Fails if the lists are empty or have different lengths.

Compilation flags:
    static

Template:
    covariance(List1,List2,Covariance)
Mode and number of proofs:
    covariance(+list(number),+list(number),-float) - zero_or_one

---

correlation/3

Calculates the Pearson correlation coefficient of two lists of numbers. Fails if the lists are empty or have different lengths.

Compilation flags:
    static

Template:
    correlation(List1,List2,Correlation)
Mode and number of proofs:
    correlation(+list(number),+list(number),-float) - zero_or_one

---

rank_correlation/3

Calculates the Spearman rank correlation coefficient of two lists of numbers. Handles ties using average ranks. Fails if the lists are empty or have different lengths.

Compilation flags:
    static

Template:
    rank_correlation(List1,List2,Correlation)
Mode and number of proofs:
    rank_correlation(+list(number),+list(number),-float) - zero_or_one

---

trimmed_mean/3

Calculates the trimmed mean of a list of numbers by removing a fraction of extreme values from both ends (with the fraction in the half-open interval [0.0, 0.5)). Fails if the list is empty or if too many elements would be trimmed.

Compilation flags:
    static

Template:
    trimmed_mean(Fraction,List,Mean)
Mode and number of proofs:
    trimmed_mean(+float,+list(number),-float) - zero_or_one

---

sum_of_squares/2

Calculates the sum of squared deviations from the arithmetic mean of a list of numbers. Fails if the list is empty.

Compilation flags:
    static

Template:
    sum_of_squares(List,Sum)
Mode and number of proofs:
    sum_of_squares(+list(number),-float) - zero_or_one

---

central_moment/3

Calculates the K-th central moment of a list of numbers. The K-th central moment is the mean of the deviations from the mean raised to the power K (with $K > 0$). Fails if the list is empty.

Compilation flags:
    static

Template:
central_moment(K,List,Moment)
Mode and number of proofs:
central_moment(+positive_integer,+list(number),-float) - zero_or_one

---

min_max_normalization/2

Normalizes a list of numbers to the interval [0.0, 1.0] using min-max normalization. Fails if the list is empty or if all values are equal.

Compilation flags:
static

Template:
min_max_normalization(List,NormalizedList)
Mode and number of proofs:
min_max_normalization(+list(number),-list(float)) - zero_or_one

---

frequency_distribution/2

Computes the frequency distribution of a list of numbers, returning a list of Value-Count pairs in ascending order of value. Fails if the list is empty.

Compilation flags:
static

Template:
frequency_distribution(List,Distribution)
Mode and number of proofs:
frequency_distribution(+list(number),-list(pair(number,integer))) - zero_or_one

---

standard_error/2

Calculates the standard error of the mean of a list of numbers. Fails if the list is empty.

Compilation flags:
    static

Template:
    standard_error(List,Error)
Mode and number of proofs:
    standard_error(+list(number),-float) - zero_or_one

---

mean_squared_error/3

Calculates the mean squared error between two lists of numbers. Fails if the lists are empty or have different lengths.

Compilation flags:
    static

Template:
    mean_squared_error(List1,List2,Error)
Mode and number of proofs:
    mean_squared_error(+list(number),+list(number),-float) - zero_or_one

---

root_mean_squared_error/3

Calculates the root mean squared error between two lists of numbers. Fails if the lists are empty or have different lengths.

Compilation flags:
    static

Template:
    root_mean_squared_error(List1,List2,Error)
Mode and number of proofs:
    root_mean_squared_error(+list(number),+list(number),-float) - zero_or_one

---

valid/1

Term is a closed list of numbers.

Compilation flags:
    static

Template:
    valid(Term)
Mode and number of proofs:
    valid(@nonvar) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

statistics, sample, population

1.211 stemming

object

1.211.1 lovins_stemmer(Representation)

- Representation - Word representation. Valid values are atom, codes, and chars.

Lovins stemmer algorithm implementation for English words.

Availability:
logtalk_load(stemming(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-05

Compilation flags:
static, context_switching_calls

Implements:
public stemmer_protocol
Uses:
list

Remarks:

- Algorithm: The Lovins stemming algorithm (Lovins, 1968) removes the longest suffix from a word using a list of 294 endings, each associated with a condition for removal. It then applies transformation rules to fix spelling.
- Reference: Lovins, J.B. (1968). Development of a stemming algorithm. Mechanical Translation and Computational Linguistics, 11(1-2), 22-31.

Inherited public predicates:
stem/2 stems/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

stemmer_protocol, porter_stemmer(Representation)

object

1.211.2 porter_stemmer(Representation)

- Representation - Word representation. Valid values are atom, codes, and chars.

Porter stemmer algorithm implementation for English words.

Availability:
logtalk_load(stemming(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-05

Compilation flags:
static, context_switching_calls

Implements:
public stemmer_protocol
Uses:
integer
list

Remarks:

- Algorithm: The Porter stemming algorithm (Porter, 1980) is a widely used algorithm for reducing English words to their root form by applying a series of rules that remove common suffixes.
- Reference: Porter, M.F. (1980). An algorithm for suffix stripping. Program, 14(3), 130-137.

Inherited public predicates:
stem/2 stems/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

stemmer_protocol, lovins_stemmer(Representation)

protocol

1.211.3 stemmer_protocol

Stemmer protocol for reducing words to their stems.

Availability:
logtalk_load(stemming(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-05

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - stem/2
  - stems/2
- Protected predicates
- Private predicates
- Operators

Public predicates

stem/2

Stems a single word, returning its root form.

Compilation flags:
static

Template:
stem(Word,Stem)
Mode and number of proofs:
stem(+text,-text) - one

---

stems/2

Stems a list of words, returning a list of their root forms.

Compilation flags:
static

Template:
stems(Words,Stems)
Mode and number of proofs:
stems(+list(text),-list(text)) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

porter_stemmer(Representation), lovins_stemmer(Representation)

1.212 stomp

object

1.212.1 stomp

Portable STOMP 1.2 (Simple Text Orientated Messaging Protocol) client. Uses the sockets library for TCP communication.

Availability:
logtalk_load(stomp(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-09

Compilation flags:
static, context_switching_calls

Uses:
list
socket
term_io
user
uuid(Representation)

Remarks:

- Supported backends: ECLiPSe, GNU Prolog, SICStus Prolog, and SWI-Prolog (same as the sockets library).
- Protocol version: Implements STOMP 1.2 specification.
- Heartbeat: Supports heartbeat negotiation. Automatic heartbeat sending is not implemented; use send_heartbeat/1 manually if needed.
- Subscriptions: Supports multiple concurrent subscriptions with unique IDs.
- Transactions: Supports STOMP transactions with BEGIN, COMMIT, and ABORT.
- Frame encoding: Properly encodes/decodes header values according to STOMP 1.2 escaping rules.

Inherited public predicates:
(none)

- Public predicates
  - connect/4
  - disconnect/2
  - connection_alive/1
  - send/4
  - subscribe/4
  - unsubscribe/3
  - receive/3
  - ack/3
  - nack/3
  - begin_transaction/3
  - commit_transaction/3
  - abort_transaction/3
  - send_heartbeat/1
  - frame_command/2
  - frame_header/3
  - frame_headers/2
  - frame_body/2
- Protected predicates
- Private predicates
- Operators

Public predicates

connect/4

Connects to a STOMP server and performs the STOMP handshake. Returns a connection handle for subsequent operations.

Compilation flags:
    static

Template:
    connect(Host,Port,Connection,Options)
Mode and number of proofs:
    connect(+atom,+integer,--compound,+list) - one_or_error

Exceptions:
    Connection refused or network error:
        stomp_error(connection_failed)
    Server rejected connection:
        stomp_error(protocol_error(Message))

Remarks:

- Option login(Login): Username for authentication.
- Option passcode(Passcode): Password for authentication.
- Option host(VirtualHost): Virtual host name. Defaults to the Host parameter.
- Option heartbeat(ClientMs, ServerMs): Heartbeat timing in milliseconds. Default is 0,0 (no heartbeat).

---

disconnect/2

Gracefully disconnects from the STOMP server. Sends DISCONNECT frame and waits for RECEIPT if requested.

Compilation flags:
    static

Template:
    disconnect(Connection,Options)
Mode and number of proofs:
    disconnect(+compound,+list) - one_or_error

Remarks:

- Option receipt(ReceiptId): Request receipt confirmation. Automatically generated if not specified.

---

connection_alive/1

Checks if the connection is still open and valid.

Compilation flags:
    static

Template:

connection_alive(Connection)
Mode and number of proofs:
connection_alive(+compound) - zero_or_one

---

send/4

Sends a message to the specified destination.

Compilation flags:
static

Template:
send(Connection,Destination,Body,Options)
Mode and number of proofs:
send(+compound,+atom,+term,+list) - one_or_error

Remarks:

- Option content_type(MimeType): MIME type of the body.
- Option content_length(Length): Body length in bytes. Auto-calculated if omitted for atom/string bodies.
- Option transaction(TransactionId): Include message in the named transaction.
- Option receipt(ReceiptId): Request receipt confirmation.
- Option header(Name, Value): Add custom header (can be repeated).

---

subscribe/4

Subscribes to a destination to receive messages.

Compilation flags:
static

Template:
subscribe(Connection,Destination,SubscriptionId,Options)
Mode and number of proofs:
subscribe(+compound,+atom,+atom,+list) - one_or_error

Remarks:

- Option ack(Mode): Acknowledgment mode: auto (default), client, or client_individual.

---

unsubscribe/3

Unsubscribes from a destination.

Compilation flags:
    static

Template:
    unsubscribe(Connection,SubscriptionId,Options)
Mode and number of proofs:
    unsubscribe(+compound,+atom,+list) - one_or_error

---

receive/3

Receives a frame from the server. Returns MESSAGE, RECEIPT, or ERROR frames.

Compilation flags:
    static

Template:
    receive(Connection,Frame,Options)
Mode and number of proofs:
    receive(+compound,-compound,+list) - zero_or_one_or_error

Remarks:

- Option timeout(Milliseconds): Timeout in milliseconds. 0 for non-blocking, -1 for infinite wait. Default is -1.

---

ack/3

Acknowledges receipt of a message.

Compilation flags:
    static

Template:
    ack(Connection,AckId,Options)
Mode and number of proofs:
    ack(+compound,+atom,+list) - one_or_error

Remarks:

- Option transaction(TransactionId): Include acknowledgment in the named transaction.

---

nack/3

Negatively acknowledges a message (tells server the message was not consumed).

Compilation flags:
    static

Template:
    nack(Connection,AckId,Options)
Mode and number of proofs:
    nack(+compound,+atom,+list) - one_or_error

Remarks:

- Option transaction(TransactionId): Include negative acknowledgment in the named transaction.

---

begin_transaction/3

Begins a new transaction.

Compilation flags:
    static

Template:
    begin_transaction(Connection,TransactionId,Options)
Mode and number of proofs:
    begin_transaction(+compound,+atom,+list) - one_or_error

---

commit_transaction/3

Commits a transaction, making all its operations permanent.

Compilation flags:
    static

Template:
    commit_transaction(Connection,TransactionId,Options)
Mode and number of proofs:
    commit_transaction(+compound,+atom,+list) - one_or_error

---

abort_transaction/3

Aborts a transaction, rolling back all its operations.

Compilation flags:
    static

Template:
    abort_transaction(Connection,TransactionId,Options)
Mode and number of proofs:
    abort_transaction(+compound,+atom,+list) - one_or_error

---

send_heartbeat/1

Sends a heartbeat (EOL) to the server to keep the connection alive.

Compilation flags:
    static

Template:
    send_heartbeat(Connection)
Mode and number of proofs:
    send_heartbeat(+compound) - one_or_error

---

frame_command/2

Extracts the command from a frame.

Compilation flags:
    static

Template:
    frame_command(Frame,Command)
Mode and number of proofs:
    frame_command(+compound,-atom) - one

---

frame_header/3

Extracts a header value from a frame. Fails if header is not present.

Compilation flags:
    static

Template:
    frame_header(Frame,HeaderName,Value)
Mode and number of proofs:
    frame_header(+compound,+atom,-atom) - zero_or_one

---

frame_headers/2

Extracts all headers from a frame as a list of Name-Value pairs.

Compilation flags:
    static

Template:
    frame_headers(Frame,Headers)
Mode and number of proofs:
    frame_headers(+compound,-list) - one

---

frame_body/2

Extracts the body from a frame. Returns empty atom if no body.

Compilation flags:
    static

Template:
    frame_body(Frame,Body)
Mode and number of proofs:
    frame_body(+compound,-term) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.213 string_distance

object

1.213.1 string_distance(Representation)

- Representation - String representation. Valid values are atom, codes, and chars.

String distance predicates.

Availability:
logtalk_load(string_distance(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-05

Compilation flags:
static, context_switching_calls

Uses:
integer
list
set

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - levenshtein/3
  - damerau_levenshtein/3
  - hamming/3

  - jaro/3
  - jaro_winkler/3
  - edit_similarity/3
  - edit_similarity/4
  - longest_common_subsequence_length/3
  - longest_common_subsequence/3
  - longest_common_substring/3
  - cosine_similarity/3
  - jaccard_index/3
  - soundex/2
  - soundex_match/2
  - metaphone/2
  - metaphone_match/2
  - double_metaphone/3
  - double_metaphone_match/2
- Protected predicates
- Private predicates
- Operators

Public predicates

levenshtein/3

Computes the Levenshtein distance between two strings.

Compilation flags:
    static

Template:
    levenshtein(String1,String2,Distance)
        String1 - First input string.
        String2 - Second input string.
        Distance - Minimum number of single-character edits (insertions, deletions, substitutions) to transform String1 into String2.
Mode and number of proofs:
    levenshtein(+text,+text,-integer) - one

damerau_levenshtein/3

Computes the Damerau-Levenshtein distance between two strings.

Compilation flags:
    static

Template:
    damerau_levenshtein(String1,String2,Distance)
        String1 - First input string.
        String2 - Second input string.
        Distance - Minimum number of edits (insertions, deletions, substitutions, and adjacent transpositions) to transform String1 into String2.
Mode and number of proofs:
    damerau_levenshtein(+text,+text,-integer) - one

---

hamming/3

Computes the Hamming distance between two strings of equal length. Fails if the strings differ in length.

Compilation flags:
    static

Template:
    hamming(String1,String2,Distance)
        String1 - First input string.
        String2 - Second input string (must have the same length as String1).
        Distance - Number of positions at which the corresponding characters differ.
Mode and number of proofs:
    hamming(+text,+text,-integer) - zero_or_one

---

jaro/3

Computes the Jaro similarity score between two strings.

Compilation flags:
    static

Template:
    jaro(String1,String2,Similarity)
        String1 - First input string.
        String2 - Second input string.
        Similarity - A value between 0.0 (completely different) and 1.0 (identical), based on matching characters and transpositions.
Mode and number of proofs:
    jaro(+text,+text,-float) - one

---

jaro_winkler/3

Computes the Jaro-Winkler similarity score between two strings.

Compilation flags:
    static

Template:
    jaro_winkler(String1,String2,Similarity)
        String1 - First input string.
        String2 - Second input string.
        Similarity - A value between 0.0 and 1.0. Extends Jaro similarity with a prefix bonus: strings sharing a common prefix are scored higher.
Mode and number of proofs:
    jaro_winkler(+text,+text,-float) - one

---

edit_similarity/3

Computes the edit similarity score between two strings using Levenshtein distance.

Compilation flags:
    static

Template:
    edit_similarity(String1,String2,Similarity)
        String1 - First input string.
        String2 - Second input string.
        Similarity - A value between 0.0 and 1.0 computed as 1 - (edit distance / max length of the two strings).
Mode and number of proofs:
    edit_similarity(+text,+text,-float) - one

---

edit_similarity/4

Computes the edit similarity score between two strings using the given algorithm.

Compilation flags:
    static

Template:
    edit_similarity(Algorithm,String1,String2,Similarity)
        Algorithm - Edit distance algorithm. Valid values are levenshtein, damerau_levenshtein, hamming, and longest_common_subsequence.
        String1 - First input string.
        String2 - Second input string.
        Similarity - A value between 0.0 and 1.0 computed as 1 - (edit distance / max length of the two strings).
Mode and number of proofs:
    edit_similarity(+atom,+text,+text,-float) - one

---

longest_common_subsequence_length/3

Computes the length of the Longest Common Subsequence between two strings.

Compilation flags:
    static

Template:
    longest_common_subsequence_length(String1,String2,Length)
        String1 - First input string.
        String2 - Second input string.
        Length - Length of the longest subsequence common to both strings (characters need not be contiguous).
Mode and number of proofs:
    longest_common_subsequence_length(+text,+text,-integer) - one

---

longest_common_subsequence/3

Computes the Longest Common Subsequence itself between two strings.

Compilation flags:
    static

Template:
    longest_common_subsequence(String1,String2,Subsequence)
        String1 - First input string.
        String2 - Second input string.
        Subsequence - The longest subsequence common to both strings (characters need not be contiguous). If multiple exist, one is returned nondeterministically.
Mode and number of proofs:
    longest_common_subsequence(+text,+text,-atom) - one

---

longest_common_substring/3

Computes the longest contiguous common substring between two strings.

Compilation flags:
    static

Template:
    longest_common_substring(String1,String2,Substring)
        String1 - First input string.
        String2 - Second input string.
        Substring - The longest contiguous substring shared by both strings. If multiple exist, one is returned nondeterministically.
Mode and number of proofs:
    longest_common_substring(+text,+text,-atom) - one

---

cosine_similarity/3

Computes the cosine similarity between two token lists.

Compilation flags:
    static

Template:
    cosine_similarity(Tokens1,Tokens2,Similarity)
        Tokens1 - First token list (e.g., list of words or character n-grams).
        Tokens2 - Second token list.
        Similarity - A value between 0.0 and 1.0 representing the cosine of the angle between the two token vectors.
Mode and number of proofs:
    cosine_similarity(+list(text),+list(text),-float) - one

---

jaccard_index/3

Computes the Jaccard index (similarity) between two token lists.

Compilation flags:
    static

Template:
    jaccard_index(Tokens1,Tokens2,Index)
        Tokens1 - First token list (e.g., list of words or character n-grams).
        Tokens2 - Second token list.
        Index - A value between 0.0 (no overlap) and 1.0 (identical sets), computed as |intersection| / |union|.
Mode and number of proofs:
    jaccard_index(+list(text),+list(text),-float) - one

---

soundex/2

Computes the Soundex phonetic encoding for a string.

Compilation flags:
    static

Template:
    soundex(String,Encoding)
        String - Input string (typically a name).
        Encoding - A four-character Soundex code representing the phonetic encoding.
Mode and number of proofs:
    soundex(+text,-atom) - one

---

soundex_match/2

Succeeds if two strings share the same Soundex code.

Compilation flags:
    static

Template:
soundex_match(String1,String2)
String1 - First input string.
String2 - Second input string.
Mode and number of proofs:
soundex_match(+text,+text) - one

---

metaphone/2

Computes the Metaphone phonetic key for a string.

Compilation flags:
static

Template:
metaphone(String,Encoding)
String - Input string (typically a name).
Encoding - The Metaphone phonetic encoding, a more accurate phonetic encoding than Soundex.
Mode and number of proofs:
metaphone(+text,-atom) - one

---

metaphone_match/2

Succeeds if two strings share the same Metaphone key.

Compilation flags:
static

Template:
metaphone_match(String1,String2)
String1 - First input string.
String2 - Second input string.
Mode and number of proofs:
metaphone_match(+text,+text) - one

---

double_metaphone/3

Computes the Double Metaphone encoding of a text, returning both primary and alternative encodings.

Compilation flags:
    static

Template:
    double_metaphone(Text,Primary,Alternative)
        Text - Input string (typically a name).
        Primary - Primary Double Metaphone encoding.
        Alternative - Alternative Double Metaphone encoding.
Mode and number of proofs:
    double_metaphone(+text,-atom,-atom) - one

---

double_metaphone_match/2

Succeeds if the Double Metaphone encodings of two texts match (either primary or alternative encodings).

Compilation flags:
    static

Template:
    double_metaphone_match(String1,String2)
        String1 - First input string.
        String2 - Second input string.
Mode and number of proofs:
    double_metaphone_match(+text,+text) - zero_or_one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.214 strings

object

1.214.1 string(Representation)

- Representation - String representation. Valid values are atom, codes, and chars.

String manipulation predicates supporting different string representations.

Availability:
logtalk_load(strings(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-04-17

Compilation flags:
static, context_switching_calls

Uses:
list

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - atom_string/2
  - number_string/2

  - string_chars/2
  - string_codes/2
  - string_concat/3
  - string_length/2
  - sub_string/5
  - string_upper/2
  - string_lower/2
  - split_string/4
  - atomics_to_string/2
  - atomics_to_string/3
  - trim/2
  - trim/3
  - trim_left/2
  - trim_left/3
  - trim_right/2
  - trim_right/3
- Protected predicates
- Private predicates
- Operators

Public predicates

atom_string/2

Converts between an atom and a string.

Compilation flags:
    static

Template:
    atom_string(Atom,String)
Mode and number of proofs:
    atom_string(+atom,?text) - zero_or_one
    atom_string(-atom,+text) - zero_or_one

number_string/2

Converts between a number and a string. Fails if the string does not represent a valid number.

Compilation flags:
    static

Template:
    number_string(Number,String)
Mode and number of proofs:
    number_string(+number,?text) - zero_or_one
    number_string(-number,+text) - zero_or_one

---

string_chars/2

Converts between a string and a list of characters.

Compilation flags:
    static

Template:
    string_chars(String,Chars)
Mode and number of proofs:
    string_chars(+text,?list(character)) - zero_or_one
    string_chars(-text,+list(character)) - zero_or_one

---

string_codes/2

Converts between a string and a list of character codes.

Compilation flags:
    static

Template:
    string_codes(String,Codes)
Mode and number of proofs:
    string_codes(+text,?list(character_code)) - zero_or_one

string_codes(-text,+list(character_code)) - zero_or_one

---

string_concat/3

Concatenates two strings.

Compilation flags:
static

Template:
string_concat(String1,String2,String3)
Mode and number of proofs:
string_concat(+text,+text,?text) - zero_or_one
string_concat(?text,?text,+text) - zero_or_more

---

string_length/2

Returns the length of a string.

Compilation flags:
static

Template:
string_length(String,Length)
Mode and number of proofs:
string_length(+text,?integer) - zero_or_one

---

sub_string/5

Extracts a substring from a string.

Compilation flags:
static

Template:
sub_string(String,Before,Length,After,SubString)
Mode and number of proofs:
sub_string(+text,?integer,?integer,?integer,?text) - zero_or_more

---

string_upper/2

Converts a string to uppercase (ASCII only).

Compilation flags:
static

Template:
string_upper(String,UpperString)
Mode and number of proofs:
string_upper(+text,?text) - zero_or_one

---

string_lower/2

Converts a string to lowercase (ASCII only).

Compilation flags:
static

Template:
string_lower(String,LowerString)
Mode and number of proofs:
string_lower(+text,?text) - zero_or_one

---

split_string/4

Decomposes String into SubStrings according to separators SepChars and padding characters PadChars. The string is split at the separators, and any padding characters around the resulting sub-strings are removed. Characters in both SepChars and PadChars are treated as separators where sequences count as one separator, and are ignored at string boundaries.

Compilation flags:
    static

Template:
    split_string(String,SepChars,PadChars,SubStrings)
Mode and number of proofs:
    split_string(+text,+text,+text,-list(text)) - one

---

atomics_to_string/2

Concatenates the atomic terms in List into String. The list may contain numbers, atoms, and strings (in the current representation).

Compilation flags:
    static

Template:
    atomics_to_string(List,String)
Mode and number of proofs:
    atomics_to_string(++list(atomic),-text) - one

---

atomics_to_string/3

Concatenates the atomic terms in List into String, with Separator inserted between each element. The list may contain numbers, atoms, and strings (in the current representation).

Compilation flags:
    static

Template:

atomics_to_string(List,Separator,String)
Mode and number of proofs:
atomics_to_string(++list(atomic),+text,-text) - one

---

trim/2

Trims string by deleting all leading and trailing whitespace.

Compilation flags:
static

Template:
trim(String,Trimmed)
Mode and number of proofs:
trim(+text,-text) - one

---

trim/3

Trims string by deleting all occurrences of the characters in Elements from the beginning and end of the string.

Compilation flags:
static

Template:
trim(String,Elements,Trimmed)
Mode and number of proofs:
trim(+text,+text,-text) - one

---

trim_left/2

Trims string by deleting all leading whitespace.

Compilation flags:
    static

Template:
    trim_left(String,Trimmed)
Mode and number of proofs:
    trim_left(+text,-text) - one

---

trim_left/3

Trims string by deleting all occurrences of the characters in Elements from the beginning of the string.

Compilation flags:
    static

Template:
    trim_left(String,Elements,Trimmed)
Mode and number of proofs:
    trim_left(+text,+text,-text) - one

---

trim_right/2

Trims string by deleting all trailing whitespace.

Compilation flags:
    static

Template:
    trim_right(String,Trimmed)
Mode and number of proofs:
    trim_right(+text,-text) - one

---

trim_right/3

Trims string by deleting all occurrences of the characters in Elements from the end of the string.

Compilation flags:
    static

Template:
    trim_right(String,Elements,Trimmed)
Mode and number of proofs:
    trim_right(+text,+text,-text) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.215 subsequences

object

1.215.1 subsequences

Implementation of subsequence operations over lists.

Availability:
    logtalk_load(subsequences(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-26

Compilation flags:
static, context_switching_calls

Implements:
public subsequences_protocol
Uses:
fast_random(Algorithm)
list

Remarks:
(none)

Inherited public predicates:
alternating_subsequence/2 alternating_subsequences/2 common_subsequence/3 common_subsequences/3 count_distinct_subsequences/3 count_subsequences/2 init/2 init1/2 init_tail/2 init_tails/2 inits/2 inits1/2 is_prefix_of/2 is_subsequence_of/2 is_suffix_of/2 k_distinct_subsequence/3 k_distinct_subsequences/3 longest_common_increasing_subsequence/3 longest_common_subsequence/3 longest_decreasing_subsequence/2 longest_increasing_subsequence/2 longest_repeating_subsequence/2 nonempty_subsequence/2 nonempty_subsequences/2 power_set/2 proper_subsequence/2 random_subsequence/2 sliding_window/3 subsequence/2 subsequence/3 subsequence_at_indices/3 subsequence_length/2 subsequences/2 subsequences/3 subsequences_with_min_span/3 subslices/2 tail/2 tail1/2 tails/2 tails1/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.215.2 subsequences_protocol

Protocol for subsequence operations over lists.

Availability:
logtalk_load(subsequences(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-26

Compilation flags:
static

Dependencies:
(none)

Remarks:

- Generation operations: Predicates for generating all subsequences or variants thereof.
- Ordering variants: Predicates that support an additional Order argument (default, lexicographic, or shortlex) for controlling output order.
- Searching and matching: Predicates for finding specific subsequences with desired properties.
- Prefix and suffix operations: Predicates for checking and finding prefixes and suffixes.
- Contiguous subsequences: Predicates for working with contiguous subsequences (subslices, sliding windows).
- Random selection: Predicates for randomly selecting subsequences.
- Constrained operations: Predicates for generating subsequences with specific constraints.
- Utility predicates: Helper predicates for subsequence operations.

Inherited public predicates:
(none)

- Public predicates
  - subsequences/2
  - subsequence/2
  - subsequences/3
  - subsequence/3
  - nonempty_subsequences/2
  - nonempty_subsequence/2
  - power_set/2
  - inits/2
  - init/2
  - tails/2
  - tail/2
  - inits1/2
  - init1/2
  - tails1/2
  - tail1/2
  - init_tails/2
  - init_tail/2
  - longest_common_subsequence/3
  - longest_increasing_subsequence/2
  - longest_decreasing_subsequence/2
  - longest_common_increasing_subsequence/3
  - longest_repeating_subsequence/2
  - is_subsequence_of/2
  - proper_subsequence/2
  - subsequence_at_indices/3
  - common_subsequences/3
  - common_subsequence/3
  - count_distinct_subsequences/3
  - is_prefix_of/2
  - is_suffix_of/2
  - subslices/2
  - sliding_window/3
  - random_subsequence/2

  - subsequences_with_min_span/3
  - alternating_subsequences/2
  - alternating_subsequence/2
  - k_distinct_subsequences/3
  - k_distinct_subsequence/3
  - count_subsequences/2
  - subsequence_length/2
- Protected predicates
- Private predicates
- Operators

Public predicates

subsequences/2

Generates all subsequences of a list using default order. A subsequence maintains the relative order of elements but need not be contiguous. The empty list is included.

Compilation flags:
    static

Template:
    subsequences(List,Subsequences)
Mode and number of proofs:
    subsequences(+list,-list) - one

Examples:
    All subsequences
        subsequences([a,b,c],Subsequences)
        Subsequences=[[],[a],[b],[a,b],[c],[a,c],[b,c],[a,b,c]]

subsequence/2

True iff the second argument is a subsequence of the first argument. Subsequences of a list using default order. A subsequence maintains the relative order of elements but need not be contiguous. The empty list is included.

Compilation flags:
    static

Template:
    subsequence(List,Subsequence)
Mode and number of proofs:
    subsequence(+list,-list) - one_or_more

Examples:
    A subsequence
        subsequence([1,2],Subsequence)
        Subsequence=[]

---

subsequences/3

Generates all subsequences of a list with specified ordering: default (as naturally produced), lexicographic, or shortlex (by length first, then lexicographically).

Compilation flags:
    static

Template:
    subsequences(List,Order,Subsequences)
Mode and number of proofs:
    subsequences(+list,+atom,-list) - one

Examples:
    Shortlex order
        subsequences([a,b],shortlex,Subsequences)
        Subsequences=[[],[a],[b],[a,b]]

---

subsequence/3

True iff the third argument is a subsequence of the first argument with specified ordering: default (as naturally produced), lexicographic, or shortlex (by length first, then lexicographically).

Compilation flags:
    static

Template:
    subsequence(List,Order,Subsequence)
Mode and number of proofs:
    subsequence(+list,+atom,-list) - one_or_more

Examples:
    Shortlex order
        subsequence([a,b],shortlex,Subsequence)
        Subsequence=[]

---

nonempty_subsequences/2

Generates all non-empty subsequences of a list.

Compilation flags:
    static

Template:
    nonempty_subsequences(List,Subsequences)
Mode and number of proofs:
    nonempty_subsequences(+list,-list) - one

Examples:
    Non-empty subsequences
        nonempty_subsequences([a,b],Subsequences)
        Subsequences=[[a],[b],[a,b]]

---

nonempty_subsequence/2

True iff the second argument is a non-empty subsequence of the first argument.

Compilation flags:
    static

Template:
    nonempty_subsequence(List,Subsequence)
Mode and number of proofs:
    nonempty_subsequence(+list,-list) - one_or_more

Examples:
    Non-empty subsequence
        nonempty_subsequence([a,b],Subsequence)
        Subsequence=[a]

---

power_set/2

Generates the power set of a list (all possible subsequences). Alias for subsequences/2 when first argument is ground.

Compilation flags:
    static

Template:
    power_set(List,PowerSet)
Mode and number of proofs:
    power_set(+list,-list) - one

Examples:
    Power set
        power_set([a,b],PowerSet)
        PowerSet=[[],[a],[b],[a,b]]

---

inits/2

Generates all initial segments (prefixes) of a list, shortest first. Includes the empty list.

Compilation flags:
    static

Template:
    inits(List,Inits)
Mode and number of proofs:
    inits(+list,-list) - one

Examples:
    All prefixes
        inits([a,b,c],Inits)
        Inits=[[],[a],[a,b],[a,b,c]]

---

init/2

True iff the second argument is one of the initial segments (prefixes) of a list.

Compilation flags:
    static

Template:
    init(List,Inits)
Mode and number of proofs:
    init(+list,-term) - zero_or_more
    init(+list,+term) - zero_or_one

Examples:
    Check prefix
        init([a,b,c],[a,b])
        true

---

tails/2

Generates all final segments (suffixes) of a list, longest first. Includes the empty list.

Compilation flags:
    static

Template:
    tails(List,Tails)
Mode and number of proofs:
    tails(+list,-list) - one

Examples:
    All suffixes
        tails([a,b,c],Tails)
        Tails=[[a,b,c],[b,c],[c],[]]

---

tail/2

True iff the second argument is one of the final segments (suffixes) of a list.

Compilation flags:
    static

Template:
    tail(List,Tails)
Mode and number of proofs:
    tail(+list,-term) - zero_or_more
    tail(+list,+term) - zero_or_one

Examples:
    Check suffix
        tail([a,b,c],[b,c])
        true

---

inits1/2

Generates all non-empty initial segments (prefixes) of a list, shortest first.

Compilation flags:
    static

Template:
    inits1(List,Inits)
Mode and number of proofs:
    inits1(+list,-list) - one

Examples:
    Non-empty prefixes
        inits1([a,b,c],Inits)
        Inits=[[a],[a,b],[a,b,c]]

---

init1/2

True iff the second argument is a non-empty initial segment (prefix) of a list, shortest first.

Compilation flags:
    static

Template:
    init1(List,Init)
Mode and number of proofs:
    init1(+list,-term) - one_or_more

Examples:
    Non-empty prefix
        init1([a,b,c],Init)
        Init=[a]

---

tails1/2

Generates all non-empty final segments (suffixes) of a list, longest first.

Compilation flags:
    static

Template:
    tails1(List,Tails)
Mode and number of proofs:
    tails1(+list,-list) - one

Examples:
    Non-empty suffix
        tails1([a,b,c],Tails)
        Tails=[[a,b,c],[b,c],[c]]

---

tail1/2

True iff the second argument is a non-empty final segment (suffix) of a list, longest first.

Compilation flags:
    static

Template:
    tail1(List,Tail)
Mode and number of proofs:
    tail1(+list,-list) - one

Examples:
    Non-empty suffix
        tail1([a,b,c],Tail)
        Tail=[a,b,c]

---

init_tails/2

Generates all pairs of initial and final segments. Each pair Init-Tail represents a split of the list where Init+Tail equals the original list. When the second argument is bound, checks if it is a valid split.

Compilation flags:
    static

Template:
    init_tails(List,InitTailPairs)
Mode and number of proofs:
    init_tails(+list,-list) - one

Examples:
    All splits
        init_tails([a,b],InitTailPairs)
        InitTailPairs=['-'([],[a,b]),'-'([a],[b]),'-'([a,b],[])]

---

init_tail/2

True iff (Init,Tail) represents a split of the list where Init+Tail equals the original list.

Compilation flags:
    static

Template:
    init_tail(List,InitTailPairs)
Mode and number of proofs:
    init_tail(+list,-term) - one_or_more

Examples:
    Check split
        init_tail([a,b,c],'-'([a],[b,c]))
        true

---

longest_common_subsequence/3

Finds the longest common subsequence (LCS) between two lists. Uses dynamic programming.

Compilation flags:
    static

Template:
    longest_common_subsequence(List1,List2,LCS)
Mode and number of proofs:
    longest_common_subsequence(+list,+list,-list) - one

Examples:
    LCS example
        longest_common_subsequence([a,b,c,d,e],[a,c,e,f],LCS)
        LCS=[a,c,e]

---

longest_increasing_subsequence/2

Finds the longest strictly increasing subsequence in a list. Elements must be comparable.

Compilation flags:
    static

Template:
    longest_increasing_subsequence(List,LIS)
Mode and number of proofs:
    longest_increasing_subsequence(+list,-list) - one

Examples:
    LIS example
        longest_increasing_subsequence([3,1,4,1,5,9,2,6],LIS)
        LIS=[1,4,5,9]

---

longest_decreasing_subsequence/2

Finds the longest strictly decreasing subsequence in a list. Elements must be comparable.

Compilation flags:
    static

Template:
    longest_decreasing_subsequence(List,LDS)
Mode and number of proofs:
    longest_decreasing_subsequence(+list,-list) - one

Examples:
    LDS example
        longest_decreasing_subsequence([9,5,2,8,3,1],LDS)
        LDS=[9,5,2,1]

---

longest_common_increasing_subsequence/3

Finds the longest subsequence that is both common to two lists and strictly increasing.

Compilation flags:
    static

Template:
    longest_common_increasing_subsequence(List1,List2,LCIS)
Mode and number of proofs:
    longest_common_increasing_subsequence(+list,+list,-list) - one

Examples:
    LCIS example
        longest_common_increasing_subsequence([1,4,2,5],[4,1,3,5],LCIS)
        LCIS=[1,5]

---

longest_repeating_subsequence/2

Finds the longest subsequence that appears at least twice in the list (at different positions).

Compilation flags:
    static

Template:
    longest_repeating_subsequence(List,LRS)
Mode and number of proofs:
    longest_repeating_subsequence(+list,-list) - one

Examples:
    LRS example
        longest_repeating_subsequence([a,a,b,a,b],LRS)
        LRS=[a,b]

---

is_subsequence_of/2

Checks if the first list is a subsequence of the second list. All elements must occur in order.

Compilation flags:
    static

Template:
    is_subsequence_of(Subsequence,List)
Mode and number of proofs:
    is_subsequence_of(+list,+list) - zero_or_one

Examples:
    Valid subsequence
        is_subsequence_of([a,c],[a,b,c])
        true
    Invalid subsequence
        is_subsequence_of([c,a],[a,b,c])
        false

---

proper_subsequence/2

Checks if the first list is a proper subsequence of the second list (i.e., subsequence and not equal).

Compilation flags:
    static

Template:
    proper_subsequence(Subsequence,List)
Mode and number of proofs:
    proper_subsequence(+list,+list) - zero_or_one

Examples:
    Proper subsequence
        proper_subsequence([a,c],[a,b,c])
        true
    Not proper (equal lists)
        proper_subsequence([a,b,c],[a,b,c])
        false

---

subsequence_at_indices/3

Extracts a subsequence using a strictly increasing list of 1-based indices.

Compilation flags:
    static

Template:
    subsequence_at_indices(List,Indices,Subsequence)
Mode and number of proofs:
    subsequence_at_indices(+list,+list,-list) - zero_or_one

Examples:
    Indices selection
        subsequence_at_indices([a,b,c,d],[1,3],Subsequence)
        Subsequence=[a,c]

---

common_subsequences/3

Generates all subsequences that are common to both lists.

Compilation flags:
    static

Template:
    common_subsequences(List1,List2,CommonSubsequences)
Mode and number of proofs:
    common_subsequences(+list,+list,-list) - one

Examples:
    All common subsequences
        common_subsequences([a,b,c],[a,c,d],CommonSubsequences)
        CommonSubsequences=[[a,c],[a],[c],[]]

---

common_subsequence/3

True iff the third argument is a common subsequence of both lists.

Compilation flags:
    static

Template:
    common_subsequence(List1,List2,CommonSubsequence)
Mode and number of proofs:
    common_subsequence(+list,+list,-list) - one_or_more

Examples:
    Check common subsequence
        common_subsequence([a,b,c],[a,c,d],[a,c])
        true

---

count_distinct_subsequences/3

Counts the number of distinct occurrences of a pattern as a subsequence in a list.

Compilation flags:
    static

Template:
    count_distinct_subsequences(Pattern,List,Count)
Mode and number of proofs:
    count_distinct_subsequences(+list,+list,-integer) - one

Examples:
    Count occurrences
        count_distinct_subsequences([a,b],[a,a,b,b],Count)
        Count=4

---

is_prefix_of/2

Checks if the first list is a prefix of the second list.

Compilation flags:
    static

Template:
    is_prefix_of(Prefix,List)
Mode and number of proofs:
    is_prefix_of(+list,+list) - zero_or_one

Examples:
    Valid prefix
        is_prefix_of([a,b],[a,b,c])
        true
    Invalid prefix
        is_prefix_of([b,c],[a,b,c])
        false

---

is_suffix_of/2

Checks if the first list is a suffix of the second list.

Compilation flags:
    static

Template:
    is_suffix_of(Suffix,List)
Mode and number of proofs:
    is_suffix_of(+list,+list) - zero_or_one

Examples:
    Valid suffix
        is_suffix_of([b,c],[a,b,c])
        true
    Invalid suffix
        is_suffix_of([a,b],[a,b,c])
        false

---

subslices/2

Generates all contiguous non-empty subslices (sublists) of a list.

Compilation flags:
    static

Template:
    subslices(List,Subslice)
Mode and number of proofs:
    subslices(+list,-list) - one
    subslices(+list,?list) - zero_or_more

Examples:
    All subslices
        subslices([a,b,c],Subslice)
        Subslice=[[a],[a,b],[a,b,c],[b],[b,c],[c]]

---

sliding_window/3

Generates all contiguous windows of size N from a list. Fails if N is larger than the list length.

Compilation flags:
    static

Template:
    sliding_window(N,List,Window)
Mode and number of proofs:
    sliding_window(+integer,+list,-list) - one
    sliding_window(+integer,+list,?list) - zero_or_more

Examples:
    Windows of size 2
        sliding_window(2,[a,b,c,d],Window)
        Window=[[a,b],[b,c],[c,d]]

---

random_subsequence/2

Randomly selects one subsequence uniformly from all 2^N possible subsequences.

Compilation flags:
    static

Template:
    random_subsequence(List,Subsequence)
Mode and number of proofs:
    random_subsequence(+list,-list) - one

Examples:
    Random subsequence
        random_subsequence([a,b,c],Subsequence)
        Subsequence=[a,c]

---

subsequences_with_min_span/3

Generates subsequences where consecutive elements are at least MinSpan positions apart in the original list.

Compilation flags:
    static

Template:
    subsequences_with_min_span(MinSpan,List,Subsequence)
Mode and number of proofs:
    subsequences_with_min_span(+integer,+list,-list) - one
    subsequences_with_min_span(+integer,+list,?list) - zero_or_more

Examples:
    Min span of 2
        subsequences_with_min_span(2,[a,b,c,d],Subsequence)
        Subsequence=[a,c]

---

alternating_subsequences/2

Generates all subsequences that alternate between increasing and decreasing (or vice versa). Elements must be comparable.

Compilation flags:
    static

Template:
    alternating_subsequences(List,AlternatingSubsequences)
Mode and number of proofs:
    alternating_subsequences(+list,-list) - one

Examples:
    All alternating subsequences
        alternating_subsequences([1,3,2,4],AlternatingSubsequences)
        AlternatingSubsequences=[[1,2],[1,3],[1,4],[2,3],[2,4],[3,4]]

---

alternating_subsequence/2

True iff the second argument is a subsequence that alternates between increasing and decreasing (or vice versa). Elements must be comparable.

Compilation flags:
    static

Template:
    alternating_subsequence(List,AlternatingSubsequence)
Mode and number of proofs:
    alternating_subsequence(+list,-list) - one_or_more

Examples:
    Alternating
        alternating_subsequence([1,3,2,4],AlternatingSubsequence)
        AlternatingSubsequence=[[1,2],[1,3],[1,4],[2,3],[2,4],[3,4]]

---

k_distinct_subsequences/3

Generates all K-element subsequences where all elements are distinct (no duplicates in the subsequence itself).

Compilation flags:
    static

Template:
    k_distinct_subsequences(K,List,DistinctSubsequences)
Mode and number of proofs:
    k_distinct_subsequences(+integer,+list,-list) - one

Examples:
    All distinct only
        k_distinct_subsequences(2,[a,a,b],DistinctSubsequences)
        DistinctSubsequences=[[a,b],[a,c],[b,c]]

---

k_distinct_subsequence/3

True iff the third argument is a subsequence of the first argument that is a K-element subsequence where all elements are distinct (no duplicates in the subsequence itself).

Compilation flags:
static

Template:
k_distinct_subsequence(K,List,DistinctSubsequence)
Mode and number of proofs:
k_distinct_subsequence(+integer,+list,-list) - one

Examples:
A distinct only
k_distinct_subsequence(2,[a,a,b],DistinctSubsequence)
DistinctSubsequence=[a,b]

---

count_subsequences/2

Counts the total number of subsequences (always 2^N for a list of length N).

Compilation flags:
static

Template:
count_subsequences(List,Count)
Mode and number of proofs:
count_subsequences(+list,-integer) - one

Examples:
Count
count_subsequences([a,b,c],Count)
Count=8

---

subsequence_length/2

Returns the length of a subsequence (same as list length, provided for consistency).

Compilation flags:
    static

Template:
    subsequence_length(Subsequence,Length)
Mode and number of proofs:
    subsequence_length(+list,-integer) - one

Examples:
    Length
        subsequence_length([a,b,c],Length)
        Length=3

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.216 term_io

object

1.216.1 term_io

Term input/output from/to atom, chars, and codes.

Availability:
logtalk_load(term_io(loader))

Author: Paulo Moura
Version: 1:3:0
Date: 2023-11-14

Compilation flags:
static, context_switching_calls

Implements:
public term_io_protocol
Uses:
os

Remarks:
(none)

Inherited public predicates:
format_to_atom/3 format_to_chars/3 format_to_chars/4 format_to_codes/3
format_to_codes/4 read_from_atom/2 read_from_chars/2 read_from_codes/2
read_term_from_atom/3 read_term_from_chars/3 read_term_from_chars/4
read_term_from_codes/3 read_term_from_codes/4 with_output_to/2 write_term_to_atom/3
write_term_to_chars/3 write_term_to_chars/4 write_term_to_codes/3
write_term_to_codes/4 write_to_atom/2 write_to_chars/2 write_to_codes/2

- Public predicates
- Protected predicates
- Private predicates
  - temporary_file_/1
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

temporary_file_/1

Logtalk session and term_io specific temporary file path.

Compilation flags:
    dynamic

Template:
    temporary_file_(Path)
Mode and number of proofs:
    temporary_file_(-atom) - one

Operators

(none)

protocol

1.216.2 term_io_protocol

Predicates for term input/output from/to atom, chars, and codes. The predicates are declared as synchronized when the library is compiled using a backend supporting threads.

Availability:
    logtalk_load(term_io(loader))

Author: Paulo Moura
Version: 1:3:0
Date: 2021-10-04

Compilation flags:

static

Dependencies:
(none)

Remarks:

- Portability notes: To keep calls to these library predicates portable, use only standard read/write options and specify output formats using atoms.

Inherited public predicates:
(none)

- Public predicates
  - read_term_from_atom/3
  - read_from_atom/2
  - read_term_from_chars/3
  - read_term_from_chars/4
  - read_from_chars/2
  - read_term_from_codes/3
  - read_term_from_codes/4
  - read_from_codes/2
  - write_term_to_atom/3
  - write_to_atom/2
  - write_term_to_chars/3
  - write_term_to_chars/4
  - write_to_chars/2
  - write_term_to_codes/3
  - write_term_to_codes/4
  - write_to_codes/2
  - format_to_atom/3
  - format_to_chars/3
  - format_to_chars/4
  - format_to_codes/3
  - format_to_codes/4
  - with_output_to/2

- Protected predicates
- Private predicates
- Operators

Public predicates

read_term_from_atom/3

Reads a term from an atom using the given read options. A period at the end of the atom is optional. Valid options are those supported by the standard read_term/3 predicate.

Compilation flags:
    static, synchronized

Template:
    read_term_from_atom(Atom,Term,Options)
Mode and number of proofs:
    read_term_from_atom(+atom,-term,+list(read_option)) - one_or_error

---

read_from_atom/2

Reads a term from an atom using default read options. Shorthand for read_term_from_atom(Atom,Term,[]). A period at the end of the atom is optional.

Compilation flags:
    static

Template:
    read_from_atom(Atom,Term)
Mode and number of proofs:
    read_from_atom(+atom,-term) - one_or_error

---

read_term_from_chars/3

Reads a term from a list of characters using the given read options. A period at the end of the list is optional. Valid options are those supported by the standard read_term/3 predicate.

Compilation flags:
    static, synchronized

Template:
    read_term_from_chars(Chars,Term,Options)
Mode and number of proofs:
    read_term_from_chars(+list(character),-term,+list(read_option)) - one_or_error

---

read_term_from_chars/4

Reads a term from a list of characters using the given read options, also returning the remaining characters. A period at the end of the term is required. Valid options are those supported by the standard read_term/3 predicate.

Compilation flags:
    static

Template:
    read_term_from_chars(Chars,Term,Tail,Options)
Mode and number of proofs:
    read_term_from_chars(+list(character),-term,-list(character),+list(read_option)) - one_or_error

---

read_from_chars/2

Reads a term from a list of characters using default read options. Shorthand for read_term_from_chars(Chars,Term,[]). A period at the end of the list is optional.

Compilation flags:
    static

Template:
    read_from_chars(Chars,Term)

Mode and number of proofs:
read_from_chars(+list(character),-term) - one_or_error

---

read_term_from_codes/3

Reads a term from a list of character codes using the given read options. A period at the end of the list is optional. Valid options are those supported by the standard read_term/3 predicate.

Compilation flags:
static, synchronized

Template:
read_term_from_codes(Codes,Term,Options)
Mode and number of proofs:
read_term_from_codes(+list(character_code),-term,+list(read_option)) - one_or_error

---

read_term_from_codes/4

Reads a term from a list of character codes using the given read options, also returning the remaining character codes. A period at the end of the term is required. Valid options are those supported by the standard read_term/3 predicate.

Compilation flags:
static

Template:
read_term_from_codes(Codes,Term,Tail,Options)
Mode and number of proofs:
read_term_from_codes(+list(character_code),-term,-list(character_code),+list(read_option)) - one_or_error

---

read_from_codes/2

Reads a term from a list of character codes using default read options. Shorthand for read_term_from_codes(Codes,Term,[]). A period at the end of the list is optional.

Compilation flags:
    static

Template:
    read_from_codes(Codes,Term)
Mode and number of proofs:
    read_from_codes(+list(character_code),-term) - one_or_error

---

write_term_to_atom/3

Writes a term to an atom using the given write options. Valid options are those supported by the standard write_term/3 predicate.

Compilation flags:
    static, synchronized

Template:
    write_term_to_atom(Term,Atom,Options)
Mode and number of proofs:
    write_term_to_atom(@term,-atom,+list(write_option)) - one

---

write_to_atom/2

Writes a term to an atom using default write options. Shorthand for write_term_to_atom(Term,Atom,[]).

Compilation flags:
    static

Template:
    write_to_atom(Term,Atom)
Mode and number of proofs:
    write_to_atom(@term,-atom) - one

---

write_term_to_chars/3

Writes a term to a list of characters using the given write options. Shorthand for write_term_to_chars(Term, Chars,[],Options). Valid options are those supported by the standard write_term/3 predicate.

Compilation flags:
    static

Template:
    write_term_to_chars(Term,Chars,Options)
Mode and number of proofs:
    write_term_to_chars(@term,-list(character),+list(write_option)) - one

---

write_term_to_chars/4

Writes a term to a list of characters with the given tail using the given write options. Valid options are those supported by the standard write_term/3 predicate.

Compilation flags:
    static, synchronized

Template:
    write_term_to_chars(Term,Chars,Tail,Options)
Mode and number of proofs:
    write_term_to_chars(@term,-list(character),@term,+list(write_option)) - one

---

write_to_chars/2

Writes a term to a list of characters using default write options. Shorthand for write_term_to_chars(Term, Chars,[],[]).

Compilation flags:
    static

Template:

write_to_chars(Term,Chars)
Mode and number of proofs:
write_to_chars(@term,-list(character)) - one

---

write_term_to_codes/3

Writes a term to a list of character codes using the given write options. Shorthand for write_term_to_codes(Term,Codes,[],Options). Valid options are those supported by the standard write_term/3 predicate.

Compilation flags:
static

Template:
write_term_to_codes(Term,Codes,Options)
Mode and number of proofs:
write_term_to_codes(@term,-list(character_code),+list(write_option)) - one

---

write_term_to_codes/4

Writes a term to a list of character codes with the given tail using the given write options. Valid options are those supported by the standard write_term/3 predicate.

Compilation flags:
static, synchronized

Template:
write_term_to_codes(Term,Codes,Tail,Options)
Mode and number of proofs:
write_term_to_codes(@term,-list(character_code),@term,+list(write_option)) - one

---

write_to_codes/2

Writes a term to a list of character codes using default write options. Shorthand for write_term_to_chars(Term,Codes,[],[]).

Compilation flags:
static

Template:
write_to_codes(Term,Codes)
Mode and number of proofs:
write_to_codes(@term,-list(character_code)) - one

---

format_to_atom/3

Writes a list of arguments to an atom using the given format (specified as in the de facto standard format/2 predicate).

Compilation flags:
static, synchronized

Template:
format_to_atom(Format,Arguments,Atom)
Mode and number of proofs:
format_to_atom(@atom,+list(term),-atom) - one

---

format_to_chars/3

Writes a list of arguments to a list of characters using the given format (specified as in the de facto standard format/2 predicate). Shorthand for format_to_chars(Format,Arguments,Chars,[]).

Compilation flags:
static

Template:
format_to_chars(Format,Arguments,Chars)
Mode and number of proofs:

format_to_chars(@term,+list(term),-list(character)) - one

---

format_to_chars/4

Writes a term to a list of characters with the given tail using the given format (specified as in the de facto standard format/2 predicate).

Compilation flags:
static, synchronized

Template:
format_to_chars(Format,Arguments,Chars,Tail)
Mode and number of proofs:
format_to_chars(@term,+list(term),-list(character),@term) - one

---

format_to_codes/3

Writes a list of arguments to a list of character codes using the given format (specified as in the de facto standard format/2 predicate). Shorthand for format_to_codes(Format,Arguments,Codes,[]).

Compilation flags:
static

Template:
format_to_codes(Format,Arguments,Codes)
Mode and number of proofs:
format_to_codes(@term,+list(term),-list(character_code)) - one

---

format_to_codes/4

Writes a list of arguments to a list of character codes with the given tail using the given format (specified as in the de facto standard format/2 predicate).

Compilation flags:
    static, synchronized

Template:
    format_to_codes(Format,Arguments,Codes,Tail)
Mode and number of proofs:
    format_to_codes(@term,+list(term),-list(character_code),@term) - one

---

with_output_to/2

Calls a goal deterministically with output to the given format: atom(Atom), chars(Chars), chars(Chars,Tail), codes(Codes), or codes(Codes,Tail).

Compilation flags:
    static, synchronized

Template:
    with_output_to(Output,Goal)
Meta-predicate template:
    with_output_to(*,0)
Mode and number of proofs:
    with_output_to(+compound,+callable) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.217 thurstone_mosteller_ranker

object

1.217.1 thurstone_mosteller_ranker

Thurstone-Mosteller Case V pairwise preference ranker. Learns one real-valued latent utility per item from a dataset object implementing the pairwise_ranking_dataset_protocol protocol by fitting continuity-corrected empirical paired-comparison probabilities with a deterministic weighted least-squares normal model, and returns a self-describing ranker term with diagnostics that can be used for ranking and export.

Availability:
logtalk_load(thurstone_mosteller_ranker(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public ranking_dataset_common
public score_ranker_model_common
Uses:
avltree
list
numberlist

Remarks:
(none)

Inherited public predicates:

check_option/1 check_options/1 check_ranker/1 default_option/1 default_options/1
diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 grouped_dataset_groups/2
grouped_dataset_items/2 grouped_dataset_items/3 grouped_dataset_summary/2 learn/2
learn/3 option/2 option/3 pairwise_dataset_connected_components/2
pairwise_dataset_declared_items/2 pairwise_dataset_items/2 pairwise_dataset_matchups/2
pairwise_dataset_preferences/2 pairwise_dataset_summary/2 pairwise_dataset_win_totals/2
pairwise_measurement_dataset_connected_components/2
pairwise_measurement_dataset_declared_items/2 pairwise_measurement_dataset_items/2
pairwise_measurement_dataset_measurements/2 pairwise_measurement_dataset_summary/2
print_ranker/1 rank/3 ranker_options/2 scores/2
temporal_pairwise_dataset_connected_components/2
temporal_pairwise_dataset_declared_items/2 temporal_pairwise_dataset_games/2
temporal_pairwise_dataset_games/3 temporal_pairwise_dataset_items/2
temporal_pairwise_dataset_periods/2 temporal_pairwise_dataset_summary/2 valid_option/1
valid_options/1 valid_ranker/1 validate_grouped_dataset/1 validate_grouped_dataset/2
validate_pairwise_dataset/1 validate_pairwise_dataset/2
validate_pairwise_measurement_dataset/1 validate_pairwise_measurement_dataset/2
validate_temporal_pairwise_dataset/1 validate_temporal_pairwise_dataset/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

pairwise_ranking_dataset_protocol, ranker_protocol, bradley_terry_ranker, regularized_bradley_terry_ranker

1.218 time_scales

object

1.218.1 time_scales

Time scale conversions for UTC, TAI, TT, UT1, TDB, GPS, GST, TCG, and TCB using bundled and optional override data.

Availability:
logtalk_load(time_scales(loader))

Author: Paulo Moura
Version: 1:0:1
Date: 2026-04-07

Compilation flags:
static, context_switching_calls

Implements:
public time_scales_protocol
Uses:
date
time_scales_data

Remarks:

- Supported UTC range: From 1972-01-01T00:00:00Z onwards.
- Time representation: Instants are represented as instant(Scale,Seconds,fraction(Numerator, Denominator)) with normalized fractional part in the [0,1[interval.

Inherited public predicates:
check_conversion/3 check_convert/4 check_instant/1 check_offset/3 clear_dut1_override/0 clear_leap_seconds_override/0 convert/4 dut1_entries/1 dut1_source/1 instant_to_utc_date_time/2 leap_second_date/2 leap_seconds_entries/1 leap_seconds_source/1 load_dut1_override/1 load_leap_seconds_override/1 offset/3 save_dut1_entries/1 save_leap_seconds_entries/1 supported_range/2 utc_date_time_to_instant/2 valid_conversion/3 valid_instant/1 valid_scale/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

time_scales_protocol, date

object

1.218.2 time_scales_data

Bundled and override data plus constants for UTC/TAI/TT/UT1/TDB/TCG/TCB conversions.

Availability:
logtalk_load(time_scales(loader))

Author: Paulo Moura
Version: 0:2:0
Date: 2026-02-26

Compilation flags:
static, context_switching_calls

Uses:
logtalk

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - load_leap_seconds_override/1
  - clear_leap_seconds_override/0
  - leap_seconds_source/1
  - leap_seconds_entries/1
  - save_leap_seconds_entries/1
  - load_dut1_override/1
  - clear_dut1_override/0
  - dut1_source/1
  - dut1_entries/1
  - save_dut1_entries/1
  - leap_offset_at_utc_unix/2
  - leap_effective_date/2
  - tt_minus_tai/2
  - dut1_offset_at_utc_unix/3
  - tdb_minus_tt_approx/3
  - tcg_minus_tt_approx/3
  - tcb_minus_tdb_approx/3
  - tai_minus_utc_for_tai_unix/2
- Protected predicates
- Private predicates
- Operators

Public predicates

load_leap_seconds_override/1

Loads leap-second override data from a file containing leap(UnixSeconds,OffsetSeconds). terms.

Compilation flags:
    static

Template:
    load_leap_seconds_override(File)
Mode and number of proofs:
    load_leap_seconds_override(+atom) - one

Remarks:

- Term format: Each term must be of the form leap(UnixSeconds,OffsetSeconds)..
- Term ordering: Terms must be sorted by increasing UnixSeconds.
- Value constraints: UnixSeconds must be an integer greater than or equal to 63072000 and OffsetSeconds must be a non-decreasing integer sequence.

---

clear_leap_seconds_override/0

Clears leap-second override data and reverts to bundled data.

Compilation flags:
    static

Mode and number of proofs:
    clear_leap_seconds_override - one

---

leap_seconds_source/1

Returns the active leap-seconds data source as bundled or override.

Compilation flags:
    static

Template:
    leap_seconds_source(Source)
Mode and number of proofs:
    leap_seconds_source(-atom) - one

---

leap_seconds_entries/1

Returns the active leap-seconds data as an ordered list of leap(UnixSeconds,OffsetSeconds) terms.

Compilation flags:
    static

Template:
    leap_seconds_entries(Entries)
Mode and number of proofs:
    leap_seconds_entries(-list) - one

---

save_leap_seconds_entries/1

Saves the active leap-seconds data to a file using leap(UnixSeconds,OffsetSeconds). terms.

Compilation flags:
    static

Template:
    save_leap_seconds_entries(File)
Mode and number of proofs:
    save_leap_seconds_entries(+atom) - one

---

load_dut1_override/1

Loads DUT1 override data from a file containing dut1(UnixSeconds,Numerator,Denominator). terms.

Compilation flags:
    static

Template:
    load_dut1_override(File)
Mode and number of proofs:
    load_dut1_override(+atom) - one

Remarks:

- Term format: Each term must be of the form dut1(UnixSeconds,Numerator,Denominator)..
- Term ordering: Terms must be sorted by increasing UnixSeconds.
- Value constraints: UnixSeconds and Numerator must be integers; Denominator must be a positive integer; UnixSeconds must be greater than or equal to 63072000.

---

clear_dut1_override/0

Clears DUT1 override data and reverts to bundled data.

Compilation flags:
    static

Mode and number of proofs:
    clear_dut1_override - one

---

dut1_source/1

Returns the active DUT1 data source as bundled or override.

Compilation flags:
    static

Template:

dut1_source(Source)
Mode and number of proofs:
dut1_source(-atom) - one

---

dut1_entries/1

Returns the active DUT1 data as an ordered list of dut1(UnixSeconds,Numerator,Denominator) terms.

Compilation flags:
static

Template:
dut1_entries(Entries)
Mode and number of proofs:
dut1_entries(-list) - one

---

save_dut1_entries/1

Saves the active DUT1 data to a file using dut1(UnixSeconds,Numerator,Denominator). terms.

Compilation flags:
static

Template:
save_dut1_entries(File)
Mode and number of proofs:
save_dut1_entries(+atom) - one

---

leap_offset_at_utc_unix/2

Returns the TAI-UTC offset in SI seconds for a given UTC Unix epoch second within the supported range.

Compilation flags:
    static

Template:
    leap_offset_at_utc_unix(UnixSeconds,OffsetSeconds)
Mode and number of proofs:
    leap_offset_at_utc_unix(+integer,-integer) - zero_or_one

---

leap_effective_date/2

Enumerates UTC effective dates for TAI-UTC offset changes and their resulting offset in SI seconds.

Compilation flags:
    static

Template:
    leap_effective_date(UTCDateTime,OffsetSeconds)
Mode and number of proofs:
    leap_effective_date(?compound,?integer) - zero_or_more

---

tt_minus_tai/2

Returns the constant TT minus TAI offset as a rational value Numerator/Denominator seconds.

Compilation flags:
    static

Template:
    tt_minus_tai(Numerator,Denominator)
Mode and number of proofs:
    tt_minus_tai(-integer,-integer) - one

---

dut1_offset_at_utc_unix/3

Returns DUT1 (UT1-UTC) at a UTC Unix epoch second as a rational value Numerator/Denominator.

Compilation flags:
    static

Template:
    dut1_offset_at_utc_unix(UnixSeconds,Numerator,Denominator)
Mode and number of proofs:
    dut1_offset_at_utc_unix(+integer,-integer,-integer) - zero_or_one

---

tdb_minus_tt_approx/3

Returns an approximate TDB-TT offset in seconds for a TT instant represented by integer seconds and normalized fraction.

Compilation flags:
    static

Template:
    tdb_minus_tt_approx(TTSeconds,Fraction,OffsetSeconds)
Mode and number of proofs:
    tdb_minus_tt_approx(+integer,+compound,-float) - one

---

tcg_minus_tt_approx/3

Returns an approximate TCG-TT offset in seconds for a TT instant represented by integer seconds and normalized fraction.

Compilation flags:
    static

Template:
    tcg_minus_tt_approx(TTSeconds,Fraction,OffsetSeconds)
Mode and number of proofs:
    tcg_minus_tt_approx(+integer,+compound,-float) - one

---

tcb_minus_tdb_approx/3

Returns an approximate TCB-TDB offset in seconds for a TDB instant represented by integer seconds and normalized fraction.

Compilation flags:
    static

Template:
    tcb_minus_tdb_approx(TDBSeconds,Fraction,OffsetSeconds)
Mode and number of proofs:
    tcb_minus_tdb_approx(+integer,+compound,-float) - one

---

tai_minus_utc_for_tai_unix/2

Returns the TAI-UTC offset in SI seconds for a given TAI instant represented as Unix-like integer seconds.

Compilation flags:
    static

Template:
    tai_minus_utc_for_tai_unix(TAISeconds,OffsetSeconds)
Mode and number of proofs:
    tai_minus_utc_for_tai_unix(+integer,-integer) - zero_or_one

---

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.218.3 time_scales_protocol

Time scales conversion protocol for UTC, TAI, TT, UT1, TDB, GPS, GST, TCG, and TCB.

Availability:
logtalk_load(time_scales(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-25

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - valid_scale/1
  - supported_range/2
  - leap_second_date/2
  - load_leap_seconds_override/1
  - clear_leap_seconds_override/0

    - leap_seconds_source/1
    - leap_seconds_entries/1
    - save_leap_seconds_entries/1
    - load_dut1_override/1
    - clear_dut1_override/0
    - dut1_source/1
    - dut1_entries/1
    - save_dut1_entries/1
    - utc_date_time_to_instant/2
    - valid_instant/1
    - check_instant/1
    - valid_conversion/3
    - check_conversion/3
    - instant_to_utc_date_time/2
    - convert/4
    - check_convert/4
    - offset/3
    - check_offset/3
- Protected predicates
- Private predicates
- Operators

Public predicates

valid_scale/1

True when the argument is a supported time scale (utc, tai, tt, ut1, tdb, gps, gst, tcg, or tcb).

Compilation flags:
    static

Template:
    valid_scale(Scale)
Mode and number of proofs:
    valid_scale(@atom) - zero_or_one

supported_range/2

Returns the supported UTC datetime range as date_time(Year,Month,Day,Hour,Minute,Second) terms.

Compilation flags:
    static

Template:
    supported_range(StartUTC,EndUTC)
Mode and number of proofs:
    supported_range(-compound,-compound) - one

---

leap_second_date/2

Enumerates UTC effective dates for TAI-UTC step changes and their resulting offset in SI seconds.

Compilation flags:
    static

Template:
    leap_second_date(UTCDateTime,OffsetSeconds)
Mode and number of proofs:
    leap_second_date(?compound,?integer) - zero_or_more

---

load_leap_seconds_override/1

Loads leap-second override data from a user-provided file containing leap(UnixSeconds,OffsetSeconds). terms.

Compilation flags:
    static

Template:
    load_leap_seconds_override(File)
Mode and number of proofs:
    load_leap_seconds_override(+atom) - one

---

clear_leap_seconds_override/0

Clears any previously loaded leap-second override data and reverts to bundled data.

Compilation flags:
    static

Mode and number of proofs:
    clear_leap_seconds_override - one

---

leap_seconds_source/1

Returns the active leap-seconds data source as either bundled or override.

Compilation flags:
    static

Template:
    leap_seconds_source(Source)
Mode and number of proofs:
    leap_seconds_source(-atom) - one

---

leap_seconds_entries/1

Returns the active leap-seconds table as an ordered list of leap(UnixSeconds,OffsetSeconds) terms.

Compilation flags:
    static

Template:
    leap_seconds_entries(Entries)
Mode and number of proofs:
    leap_seconds_entries(-list) - one

---

save_leap_seconds_entries/1

Saves the active leap-seconds table to a file as leap(UnixSeconds,OffsetSeconds). terms.

Compilation flags:
    static

Template:
    save_leap_seconds_entries(File)
Mode and number of proofs:
    save_leap_seconds_entries(+atom) - one

---

load_dut1_override/1

Loads DUT1 override data from a user-provided file containing dut1(UnixSeconds,Numerator,Denominator). terms.

Compilation flags:
    static

Template:
    load_dut1_override(File)
Mode and number of proofs:
    load_dut1_override(+atom) - one

---

clear_dut1_override/0

Clears any previously loaded DUT1 override data and reverts to bundled data.

Compilation flags:
    static

Mode and number of proofs:
    clear_dut1_override - one

---

dut1_source/1

Returns the active DUT1 data source as either bundled or override.

Compilation flags:
    static

Template:
    dut1_source(Source)
Mode and number of proofs:
    dut1_source(-atom) - one

---

dut1_entries/1

Returns the active DUT1 table as an ordered list of dut1(UnixSeconds,Numerator,Denominator) terms.

Compilation flags:
    static

Template:
    dut1_entries(Entries)
Mode and number of proofs:
    dut1_entries(-list) - one

---

save_dut1_entries/1

Saves the active DUT1 table to a file as dut1(UnixSeconds,Numerator,Denominator). terms.

Compilation flags:
    static

Template:
    save_dut1_entries(File)
Mode and number of proofs:
    save_dut1_entries(+atom) - one

---

utc_date_time_to_instant/2

Converts a UTC datetime term to an instant represented as instant(utc,Seconds,fraction(Numerator, Denominator)) where Seconds are Unix epoch seconds and the fractional component is normalized.

Compilation flags:
    static

Template:
    utc_date_time_to_instant(UTCDateTime,Instant)
Mode and number of proofs:
    utc_date_time_to_instant(+compound,-compound) - zero_or_one

---

valid_instant/1

True when the argument is a valid supported instant term instant(Scale,Seconds,fraction(Numerator, Denominator)).

Compilation flags:
    static

Template:
    valid_instant(Instant)
Mode and number of proofs:
    valid_instant(+compound) - zero_or_one

---

check_instant/1

Checks an instant term and throws an error instead of failing on invalid input.

Compilation flags:
    static

Template:
    check_instant(Instant)
Mode and number of proofs:
    check_instant(+compound) - one_or_error

Exceptions:
Instant is a variable:
instantiation_error
Instant is not a valid instant term:
domain_error(instant,Instant)
Scale in Instant is neither a variable nor an atom:
type_error(atom,Scale)
Scale in Instant is not supported:
domain_error(time_scale,Scale)
Seconds in Instant is neither a variable nor an integer:
type_error(integer,Seconds)
Seconds in Instant is before supported UTC epoch:
domain_error(utc_unix_seconds,Seconds)
Fraction term in Instant is not of the form fraction(Numerator,Denominator):
domain_error(fraction,Fraction)
Numerator in Instant is neither a variable nor an integer:
type_error(integer,Numerator)
Denominator in Instant is neither a variable nor an integer:
type_error(integer,Denominator)
Denominator in Instant is not positive:
domain_error(positive_denominator,Denominator)
Normalized fraction in Instant is not in the [0,1[interval:
domain_error(normalized_fraction,Fraction)

---

valid_conversion/3

True when an instant is valid for FromScale and conversion to ToScale is admissible.

Compilation flags:
static

Template:
valid_conversion(Instant,FromScale,ToScale)
Mode and number of proofs:
valid_conversion(+compound,+atom,+atom) - zero_or_one

---

check_conversion/3

Checks a conversion request and throws an error instead of failing on invalid input.

Compilation flags:
    static

Template:
    check_conversion(Instant,FromScale,ToScale)
Mode and number of proofs:
    check_conversion(+compound,+atom,+atom) - one_or_error

Exceptions:
    Instant, FromScale, or ToScale is a variable:
        instantiation_error
    FromScale is neither a variable nor an atom:
        type_error(atom,FromScale)
    ToScale is neither a variable nor an atom:
        type_error(atom,ToScale)
    FromScale is not supported:
        domain_error(time_scale,FromScale)
    ToScale is not supported:
        domain_error(time_scale,ToScale)
    Instant is not a valid instant term:
        domain_error(instant,Instant)
    Scale in Instant is neither a variable nor an atom:
        type_error(atom,Scale)
    Scale in Instant is not supported:
        domain_error(time_scale,Scale)
    Seconds in Instant is neither a variable nor an integer:
        type_error(integer,Seconds)
    Seconds in Instant is before supported UTC epoch:
        domain_error(utc_unix_seconds,Seconds)
    Fraction term in Instant is not of the form fraction(Numerator,Denominator):
        domain_error(fraction,Fraction)
    Numerator in Instant is neither a variable nor an integer:
        type_error(integer,Numerator)
    Denominator in Instant is neither a variable nor an integer:
        type_error(integer,Denominator)
    Denominator in Instant is not positive:
        domain_error(positive_denominator,Denominator)
    Normalized fraction in Instant is not in the [0,1[interval:
        domain_error(normalized_fraction,Fraction)
    Instant scale does not match FromScale:
        domain_error(scale_mismatch,instant_scale(InstantScale,FromScale))

---

instant_to_utc_date_time/2

Converts an UTC instant represented as instant(utc,Seconds,fraction(Numerator,Denominator)) to a UTC datetime term. Requires zero fractional part.

Compilation flags:
    static

Template:
    instant_to_utc_date_time(Instant,UTCDateTime)
Mode and number of proofs:
    instant_to_utc_date_time(+compound,-compound) - zero_or_one

---

convert/4

Converts an instant(Scale,Seconds,fraction(Numerator,Denominator)) from FromScale to ToScale.

Compilation flags:
    static

Template:
    convert(Instant,FromScale,ToScale,ConvertedInstant)
Mode and number of proofs:
    convert(+compound,+atom,+atom,-compound) - zero_or_one

---

check_convert/4

Converts an instant from FromScale to ToScale and throws an error instead of failing on invalid input.

Compilation flags:
    static

Template:
    check_convert(Instant,FromScale,ToScale,ConvertedInstant)

Mode and number of proofs:
check_convert(+compound,+atom,+atom,-compound) - one_or_error

Exceptions:
Instant, FromScale, or ToScale is a variable:
instantiation_error
FromScale is neither a variable nor an atom:
type_error(atom,FromScale)
ToScale is neither a variable nor an atom:
type_error(atom,ToScale)
FromScale is not supported:
domain_error(time_scale,FromScale)
ToScale is not supported:
domain_error(time_scale,ToScale)
Instant is not a valid instant term:
domain_error(instant,Instant)
Scale in Instant is neither a variable nor an atom:
type_error(atom,Scale)
Scale in Instant is not supported:
domain_error(time_scale,Scale)
Seconds in Instant is neither a variable nor an integer:
type_error(integer,Seconds)
Seconds in Instant is before supported UTC epoch:
domain_error(utc_unix_seconds,Seconds)
Fraction term in Instant is not of the form fraction(Numerator,Denominator):
domain_error(fraction,Fraction)
Numerator in Instant is neither a variable nor an integer:
type_error(integer,Numerator)
Denominator in Instant is neither a variable nor an integer:
type_error(integer,Denominator)
Denominator in Instant is not positive:
domain_error(positive_denominator,Denominator)
Normalized fraction in Instant is not in the [0,1[interval:
domain_error(normalized_fraction,Fraction)
Instant scale does not match FromScale:
domain_error(scale_mismatch,instant_scale(InstantScale,FromScale))

offset/3

Returns the offset required to convert an instant to a target scale as rational(Numerator,Denominator).

Compilation flags:
    static

Template:
    offset(Instant,ToScale,Offset)
Mode and number of proofs:
    offset(+compound,+atom,-compound) - zero_or_one

---

check_offset/3

Returns the conversion offset and throws an error instead of failing on invalid input.

Compilation flags:
    static

Template:
    check_offset(Instant,ToScale,Offset)
Mode and number of proofs:
    check_offset(+compound,+atom,-compound) - one_or_error

Exceptions:
    Instant or ToScale is a variable:
        instantiation_error
    ToScale is neither a variable nor an atom:
        type_error(atom,ToScale)
    ToScale is not supported:
        domain_error(time_scale,ToScale)
    Instant is not a valid instant term:
        domain_error(instant,Instant)
    Scale in Instant is neither a variable nor an atom:
        type_error(atom,Scale)
    Scale in Instant is not supported:
        domain_error(time_scale,Scale)
    Seconds in Instant is neither a variable nor an integer:
        type_error(integer,Seconds)
    Seconds in Instant is before supported UTC epoch:
        domain_error(utc_unix_seconds,Seconds)

Fraction term in Instant is not of the form fraction(Numerator,Denominator):
    domain_error(fraction,Fraction)
Numerator in Instant is neither a variable nor an integer:
    type_error(integer,Numerator)
Denominator in Instant is neither a variable nor an integer:
    type_error(integer,Denominator)
Denominator in Instant is not positive:
    domain_error(positive_denominator,Denominator)
Normalized fraction in Instant is not in the [0,1[interval:
    domain_error(normalized_fraction,Fraction)

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

time_scales, date

1.219 timeout

object

1.219.1 timeout

Predicates for calling goal with a time limit.

Availability:
    logtalk_load(timeout(loader))

Author: Paulo Moura
Version: 0:10:0
Date: 2022-06-15

Compilation flags:
static, context_switching_calls

Dependencies:
(none)

Remarks:

- Supported backend Prolog systems: B-Prolog, ECLiPSe, SICStus Prolog, SWI-Prolog, Trealla Prolog, XSB, XVM, and YAP.

Inherited public predicates:
(none)

- Public predicates
  - call_with_timeout/2
  - call_with_timeout/3
- Protected predicates
- Private predicates
- Operators

Public predicates

call_with_timeout/2

Calls a goal deterministically with the given time limit (expressed in seconds). Note that the goal may fail or throw an error before exhausting the time limit.

Compilation flags:
static

Template:
call_with_timeout(Goal,Timeout)
Meta-predicate template:
call_with_timeout(0,*)
Mode and number of proofs:
call_with_timeout(+callable,+positive_number) - zero_or_one

Exceptions:

Goal does not complete in the allowed time:
timeout(Goal)

---

call_with_timeout/3

Calls a goal deterministically with the given time limit (expressed in seconds) returning a reified result: true, fail, timeout, or error(Error).

Compilation flags:
static

Template:
call_with_timeout(Goal,Timeout,Result)
Meta-predicate template:
call_with_timeout(0,*,*)
Mode and number of proofs:
call_with_timeout(+callable,+positive_number,--atom) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.220 tle_orbits

object

1.220.1 tle_orbits

Parser for Two-Line Element sets with an approximate portable propagator featuring automatic branching between near-earth and deep-space variants, a legacy two-body fallback model, and ground-track sampling.

Availability:
logtalk_load(tle_orbits(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-11

Compilation flags:
static, context_switching_calls

Implements:
public tle_orbits_protocol
Uses:
crs_projections
date
iso8601
list
reader
time

Remarks:
(none)

Inherited public predicates:
ground_track/5 ground_track/6 parse/2 parse_lines/4 propagate/3 propagate/4 propagate/5 propagate_state/4 propagate_state/5

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.220.2 tle_orbits_protocol

Protocol for parsing Two-Line Element sets and performing approximate portable orbit propagation with near-earth and deep-space variants plus ground-track sampling.

Availability:
logtalk_load(tle_orbits(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-11

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - parse/2
  - parse_lines/4
  - propagate/3
  - propagate/4
  - propagate/5
  - propagate_state/4
  - propagate_state/5
  - ground_track/5
  - ground_track/6
- Protected predicates
- Private predicates
- Operators

Public predicates

parse/2

Parses one or more TLE records from a source specification into canonical tle(...) terms. Supported source specifications are atom(Atom), chars(List), codes(List), stream(Stream), and file(Path).

Compilation flags:
    static

Template:
    parse(Source,TLEs)
Mode and number of proofs:
    parse(++compound,--list(compound)) - one_or_error

parse_lines/4

Parses a single TLE record given an optional name atom or the atom none plus the two raw TLE lines.

Compilation flags:
    static

Template:
    parse_lines(Name,Line1,Line2,TLE)
Mode and number of proofs:
    parse_lines(++nonvar,++atom,++atom,--compound) - one_or_error

---

propagate/3

Propagates a parsed TLE using the default approximate model and returns a geographic(Latitude,Longitude, Height) coordinate. Supported time specifications are date_time(Year,Month,Day,Hours,Minutes,Seconds), julian_date(JulianDate), and offset_seconds(SecondsSinceEpoch).

Compilation flags:
    static

Template:
    propagate(TLE,Time,Coordinate)
Mode and number of proofs:
    propagate(++compound,++nonvar,--compound) - one_or_error

---

propagate/4

Propagates a parsed TLE using the default approximate model and returns coordinates in the requested frame. Supported frames are eci returning eci(X,Y,Z), ecef returning ecef(X,Y,Z), and wgs84_3d returning geographic(Latitude,Longitude,Height).

Compilation flags:
    static

Template:
    propagate(TLE,Time,Frame,Coordinate)

Mode and number of proofs:
propagate(++compound,++nonvar,++atom,--compound) - one_or_error

---

propagate/5

Propagates a parsed TLE using the requested propagation model and returns coordinates in the requested frame. Supported models are approximate for automatic near-earth versus deep-space dispatch, approximate_near_earth for the low-period approximate branch with J2 secular and short-period corrections plus low-order B* drag handling, approximate_deep_space for the dedicated deep-space approximate branch with resonance-aware long-period corrections, and two_body for the legacy Keplerian approximation.

Compilation flags:
static

Template:
propagate(TLE,Time,Frame,Model,Coordinate)
Mode and number of proofs:
propagate(++compound,++nonvar,++atom,++atom,--compound) - one_or_error

---

propagate_state/4

Propagates a parsed TLE using the default approximate model and returns state(Position,Velocity) in the requested frame. Velocity is derived directly from the propagated orbital elements in ECI and analytically transformed to the requested frame. Supported frames are eci returning state(eci(X,Y,Z), eci(VX,VY,VZ)), ecef returning state(ecef(X,Y,Z), ecef(VX,VY,VZ)), and wgs84_3d returning state(geographic(Latitude, Longitude,Height), enu(East,North,Up)).

Compilation flags:
static

Template:
propagate_state(TLE,Time,Frame,State)
Mode and number of proofs:
propagate_state(++compound,++nonvar,++atom,--compound) - one_or_error

---

propagate_state/5

Propagates a parsed TLE using the requested propagation model and returns state(Position,Velocity) in the requested frame. Velocity is derived directly from the propagated orbital elements in ECI and analytically transformed to the requested frame. Supported models are approximate, approximate_near_earth, approximate_deep_space, and two_body.

Compilation flags:
    static

Template:
    propagate_state(TLE,Time,Frame,Model,State)
Mode and number of proofs:
    propagate_state(++compound,++nonvar,++atom,++atom,--compound) - one_or_error

---

ground_track/5

Samples the propagated sub-satellite ground track between two UTC date_time/6 instants separated by a positive step size in seconds using the default approximate model, returning sample(DateTime, geographic(Latitude,Longitude,Height)) terms.

Compilation flags:
    static

Template:
    ground_track(TLE,StartDateTime,EndDateTime,StepSeconds,Samples)
Mode and number of proofs:
    ground_track(++compound,++compound,++compound,++number,--list(compound)) -
    one_or_error

---

ground_track/6

Samples the propagated sub-satellite ground track between two UTC date_time/6 instants separated by a positive step size in seconds using the requested propagation model. Supported models are approximate, approximate_near_earth, approximate_deep_space, and two_body.

Compilation flags:
    static

Template:
ground_track(TLE,StartDateTime,EndDateTime,StepSeconds,Model,Samples)
Mode and number of proofs:
ground_track(++compound,++compound,++compound,++number,++atom,--list(compound)) - one_or_error

---

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

tle_orbits, crs_projections_protocol

1.221 toml

object

1.221.1 toml

TOML parser and generator. Uses compound terms for parsed TOML tables, dashes for parsed TOML pairs, and atoms for parsed TOML string values.

Availability:
logtalk_load(toml(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-09

Compilation flags:
static, context_switching_calls

Extends:
public toml(compound,dash,atom)

Remarks:
(none)

Inherited public predicates:
generate/2 parse/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.221.2 toml(StringRepresentation)

- StringRepresentation - Text representation to be used when decoding TOML string values. Possible values are atom (default), chars, and codes.

TOML parser and generator. Uses compound terms for parsed TOML tables and dashes for parsed TOML pairs.

Availability:
logtalk_load(toml(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-09

Compilation flags:
static, context_switching_calls

Extends:
public toml(compound,dash,StringRepresentation)

Remarks:
(none)

Inherited public predicates:
generate/2 parse/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.221.3 toml(ObjectRepresentation,PairRepresentation,StringRepresentation)

- ObjectRepresentation - Object representation to be used when decoding TOML tables. Possible values are compound (default) and curly.
- PairRepresentation - Pair representation to be used when decoding TOML tables. Possible values are dash (default), equal, and colon.
- StringRepresentation - Text representation to be used when decoding TOML basic and literal string values. Possible values are atom (default), chars, and codes.

TOML parser and generator.

Availability:
logtalk_load(toml(loader))

Author: Paulo Moura
Version: 1:1:1
Date: 2026-04-22

Compilation flags:
static, context_switching_calls

Implements:
public toml_protocol
Uses:
list
reader

Remarks:
(none)

Inherited public predicates:
generate/2 parse/2

- Public predicates
- Protected predicates

- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.221.4 toml_protocol

TOML parser and generator protocol.

Availability:
logtalk_load(toml(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-09

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - parse/2
  - generate/2
- Protected predicates
- Private predicates
- Operators

Public predicates

parse/2

Parses TOML content read from the given source (file(Path), stream(Stream), codes(Codes), chars(Chars), or atom(Atom)) into a ground term representing the parsed TOML document.

Compilation flags:
static

Template:
parse(Source,TOML)
Mode and number of proofs:
parse(++compound,--ground) - zero_or_one

Exceptions:
Source is a variable:
instantiation_error
Source is neither a variable nor a valid source:
domain_error(toml_source,Source)

generate/2

Generates TOML output using the representation specified in the first argument (file(Path), stream(Stream), codes(Codes), chars(Chars), or atom(Atom)) from the ground TOML term in the second argument.

Compilation flags:
    static

Template:
    generate(Sink,TOML)
Mode and number of proofs:
    generate(++compound,+ground) - one_or_error

Exceptions:
    Sink is a variable:
        instantiation_error
    TOML is a variable:
        instantiation_error
    TOML is not a valid TOML term:
        domain_error(toml_term,TOML)
    Sink cannot be generated:
        domain_error(toml_sink,Sink)

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.222 tool_diagnostics

category

1.222.1 tool_diagnostics_common

Default definitions for diagnostics targets, rule enumeration, and summary/breakdown helpers.

Availability:
logtalk_load(tool_diagnostics(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-01

Compilation flags:
static

Implements:
public tool_diagnostics_protocol
Uses:
list

Remarks:
(none)

Inherited public predicates:
diagnostic/2 diagnostic/3 diagnostic_rule/5 diagnostic_rules/1 diagnostic_target/1 diagnostics/2 diagnostics/3 diagnostics_preflight/2 diagnostics_preflight/3 diagnostics_summary/2 diagnostics_summary/3 diagnostics_tool/5

- Public predicates
- Protected predicates
  - context_summaries/2
  - diagnostics_breakdown/2
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

context_summaries/2

Returns a list of context summary terms for a list of diagnostics.

Compilation flags:
    static

Template:
    context_summaries(Diagnostics,ContextSummaries)
Mode and number of proofs:
    context_summaries(+list(compound),-list(compound)) - one

---

diagnostics_breakdown/2

Returns a breakdown term with rule, severity, and confidence counts for a list of diagnostics.

Compilation flags:
    static

Template:
    diagnostics_breakdown(Diagnostics,Breakdown)
Mode and number of proofs:
    diagnostics_breakdown(+list(compound),-compound) - one

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.222.2 tool_diagnostics_protocol

Common machine-readable diagnostics protocol for developer tools.

Availability:
logtalk_load(tool_diagnostics(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-03-31

Compilation flags:
static

Dependencies:
(none)

Remarks:

- Targets and options: Implementing objects should enumerate supported targets using diagnostic_target/1 and predicates taking an Options argument must accept explanations(Boolean).
- Tool metadata predicate: Tool metadata is exposed using diagnostics_tool/5.
- Rule descriptor predicate: Rule descriptors are exposed using diagnostic_rule/5 and collected in stable order using diagnostic_rules/1.
- Diagnostic and preflight terms: Findings and analysis prerequisites are represented using diagnostic/8 and preflight_issue/7 terms.
- Summary term: Summaries are represented using diagnostics_summary/5 terms with supporting breakdown and count terms.

Inherited public predicates:
(none)

- Public predicates
  - diagnostics_tool/5

  - diagnostic_target/1
  - diagnostic_rule/5
  - diagnostic_rules/1
  - diagnostic/3
  - diagnostic/2
  - diagnostics/3
  - diagnostics/2
  - diagnostics_summary/3
  - diagnostics_summary/2
  - diagnostics_preflight/3
  - diagnostics_preflight/2
- Protected predicates
- Private predicates
- Operators

Public predicates

diagnostics_tool/5

Returns tool metadata.

Compilation flags:
    static

Template:
    diagnostics_tool(Id,Name,Version,InformationURI,Properties)
Mode and number of proofs:
    diagnostics_tool(?atom,?atom,?atom,?atom,?list(compound)) - zero_or_one

diagnostic_target/1

Enumerates supported diagnostics target patterns.

Compilation flags:
    static

Template:
    diagnostic_target(Target)
Mode and number of proofs:
    diagnostic_target(?nonvar) - zero_or_more

---

diagnostic_rule/5

Enumerates diagnostic rule descriptors.

Compilation flags:
    static

Template:
    diagnostic_rule(RuleId,ShortDescription,FullDescription,DefaultSeverity,Properties)
Mode and number of proofs:
    diagnostic_rule(?atom,?atom,?atom,?atom,?list(compound)) - zero_or_more

---

diagnostic_rules/1

Returns all supported diagnostic rule descriptors in a stable order.

Compilation flags:
    static

Template:
    diagnostic_rules(Rules)
Mode and number of proofs:
    diagnostic_rules(-list(compound)) - one

---

diagnostic/3

Enumerates, by backtracking, diagnostics for a target using the given options. Diagnostics are returned using terms of the form diagnostic(RuleId, Severity, Confidence, Message, Context, File, Lines, Properties). All implementations must accept the common option explanations(Boolean).

Compilation flags:
    static

Template:
    diagnostic(Target,Diagnostic,Options)
Mode and number of proofs:
    diagnostic(+nonvar,-compound,+list(compound)) - zero_or_more

---

diagnostic/2

Enumerates, by backtracking, diagnostics for a target using default options. Diagnostics are returned using terms of the form diagnostic(RuleId, Severity, Confidence, Message, Context, File, Lines, Properties).

Compilation flags:
    static

Template:
    diagnostic(Target,Diagnostic)
Mode and number of proofs:
    diagnostic(+nonvar,-compound) - zero_or_more

---

diagnostics/3

Returns an ordered set of diagnostics for a target using the given options. Diagnostics are returned using terms of the form diagnostic(RuleId, Severity, Confidence, Message, Context, File, Lines, Properties). All implementations must accept the common option explanations(Boolean).

Compilation flags:
    static

Template:

diagnostics(Target,Diagnostics,Options)
Mode and number of proofs:
diagnostics(+nonvar,-list(compound),+list(compound)) - one

---

diagnostics/2

Returns an ordered set of diagnostics for a target using default options. Diagnostics are returned using terms of the form diagnostic(RuleId, Severity, Confidence, Message, Context, File, Lines, Properties).

Compilation flags:
static

Template:
diagnostics(Target,Diagnostics)
Mode and number of proofs:
diagnostics(+nonvar,-list(compound)) - one

---

diagnostics_summary/3

Returns a machine-readable summary for a target using the given options. The summary counts diagnostics only and does not include preflight issues. All implementations must accept the common option explanations(Boolean).

Compilation flags:
static

Template:
diagnostics_summary(Target,Summary,Options)
Mode and number of proofs:
diagnostics_summary(+nonvar,-compound,+list(compound)) - one

---

diagnostics_summary/2

Returns a machine-readable summary for a target using default options. The summary counts diagnostics only and does not include preflight issues.

Compilation flags:
    static

Template:
    diagnostics_summary(Target,Summary)
Mode and number of proofs:
    diagnostics_summary(+nonvar,-compound) - one

---

diagnostics_preflight/3

Returns an ordered set of machine-readable preflight issues for a target using the given options. Preflight issues are returned using terms of the form preflight_issue(Id, Severity, Message, Context, File, Lines, Properties). All implementations must accept the common option explanations(Boolean).

Compilation flags:
    static

Template:
    diagnostics_preflight(Target,Issues,Options)
Mode and number of proofs:
    diagnostics_preflight(+nonvar,-list(compound),+list(compound)) - one

---

diagnostics_preflight/2

Returns an ordered set of machine-readable preflight issues for a target using default options. Preflight issues are returned using terms of the form preflight_issue(Id, Severity, Message, Context, File, Lines, Properties).

Compilation flags:
    static

Template:
    diagnostics_preflight(Target,Issues)

Mode and number of proofs:
    diagnostics_preflight(+nonvar,-list(compound)) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.223 toon

object

1.223.1 toon

TOON parser and generator. Uses curly terms for parsed TOON objects, dashes for parsed TOON pairs, and atoms for parsed TOON strings.

Availability:
    logtalk_load(toon(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2025-12-07

Compilation flags:
    static, context_switching_calls

Extends:
    public toon(curly,dash,atom)

Remarks:
    (none)

Inherited public predicates:
generate/2 parse/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.223.2 toon(StringRepresentation)

- StringRepresentation - Text representation to be used when decoding TOON strings. Possible values are atom (default), chars, and codes.

TOON parser and generator. Uses curly terms for parsed TOON objects and dashes for parsed TOON pairs.

Availability:
logtalk_load(toon(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2025-12-07

Compilation flags:
static, context_switching_calls

Extends:
public toon(curly,dash,StringRepresentation)

Remarks:
(none)

Inherited public predicates:
generate/2 parse/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.223.3 toon(ObjectRepresentation,PairRepresentation,StringRepresentation)

- ObjectRepresentation - Object representation to be used when decoding TOON objects. Possible values are curly (default) and list.
- PairRepresentation - Pair representation to be used when decoding TOON objects. Possible values are dash (default), equal, and colon.
- StringRepresentation - Text representation to be used when decoding TOON strings. Possible values are atom (default), chars, and codes.

TOON (Token-Oriented Object Notation) parser and generator. TOON is a compact, human-readable, line-oriented format that encodes the JSON data model.

Availability:
logtalk_load(toon(loader))

Author: Paulo Moura
Version: 0:2:0
Date: 2026-04-17

Compilation flags:
static, context_switching_calls

Implements:
public toon_protocol
Uses:
list
meta
reader

Remarks:
(none)

Inherited public predicates:
generate/2 parse/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.223.4 toon_protocol

TOON (Token-Oriented Object Notation) parser and generator protocol.

Availability:
logtalk_load(toon(loader))

Author: Paulo Moura
Version: 0:1:0
Date: 2025-12-15

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - parse/2
  - generate/2

- Protected predicates
- Private predicates
- Operators

Public predicates

parse/2

Parses the TOON contents read from the given source (codes(List), stream(Stream), file(Path), chars(List), or atom(Atom)) into a term. Throws an error if the TOON contents cannot be parsed.

Compilation flags:
    static

Template:
    parse(Source,Term)
Mode and number of proofs:
    parse(++compound,--term) - one_or_error

Exceptions:
    Source is a variable:
        instantiation_error
    Source is neither a variable nor a valid source:
        domain_error(toon_source,Source)

---

generate/2

Generates the content using the representation specified in the first argument (codes(List), stream(Stream), file(Path), chars(List), or atom(Atom)) for the term in the second argument. Throws an error if this term cannot be processed.

Compilation flags:
    static

Template:
    generate(Sink,Term)
Mode and number of proofs:
    generate(+compound,++term) - one_or_error

Exceptions:
Sink is a variable:
instantiation_error
Sink is neither a variable nor a valid sink:
domain_error(toon_sink,Sink)

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.224 toychr

object

1.224.1 toychrdb

Simple CHR interpreter/debugger based on the refined operational semantics of CHRs.

Availability:
logtalk_load(toychr(loader))

Author: Gregory J. Duck; adapted to Logtalk by Paulo Moura.
Version: 0:7:2
Date: 2026-04-06

Copyright: Copright 2004 Gregory J. Duck; Copyright 2019-2024 Paulo Moura
License: GPL-2.0-or-later

Compilation flags:
static, context_switching_calls

Implements:
public expanding
Uses:
list
user

Remarks:
(none)

Inherited public predicates:
goal_expansion/2 term_expansion/2

- Public predicates
  - chr_is/2
  - chr_trace/0
  - chr_notrace/0
  - chr_spy/1
  - chr_nospy/0
  - chr_no_spy/1
  - chr_option/2
- Protected predicates
  - current_prog/1
  - chr_option_print_trace/0
  - chr_option_trace_interactive/0
  - chr_option_optimization_level/1
  - chr_option_show_stack/0
  - chr_option_show_store/0
  - chr_option_show_history/0
  - chr_option_show_id/0
  - chr_option_allow_deep_guards/0
  - chr_next_state/1
  - chr_spy_point/1
- Private predicates
  - chr_rule_/1
- Operators

Public predicates

chr_is/2

Compilation flags:
    static

---

chr_trace/0

Compilation flags:
    static

---

chr_notrace/0

Compilation flags:
    static

---

chr_spy/1

Compilation flags:
    static

---

chr_nospy/0

Compilation flags:
    static

---

chr_no_spy/1

Compilation flags:
    static

---

chr_option/2

Compilation flags:
    static

Protected predicates

current_prog/1

Compilation flags:
    static

---

chr_option_print_trace/0

Compilation flags:
    dynamic

---

chr_option_trace_interactive/0

Compilation flags:
    dynamic

---

chr_option_optimization_level/1

Compilation flags:
    dynamic

---

chr_option_show_stack/0

Compilation flags:
    dynamic

---

chr_option_show_store/0

Compilation flags:
    dynamic

---

chr_option_show_history/0

Compilation flags:
    dynamic

---

chr_option_show_id/0

Compilation flags:
    dynamic

---

chr_option_allow_deep_guards/0

Compilation flags:
dynamic

---

chr_next_state/1

Compilation flags:
dynamic

---

chr_spy_point/1

Compilation flags:
dynamic

Private predicates

chr_rule_/1

Compilation flags:
dynamic

Operators

(none)

1.225 truncated_svd_projection

object

1.225.1 truncated_svd_projection

Truncated singular value decomposition reducer for continuous datasets using a portable two-sided power-iteration solver over the data matrix.

Availability:
logtalk_load(truncated_svd_projection(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-07

Compilation flags:
static, context_switching_calls

Imports:
public dimension_reducer_common
Uses:
format
linear_algebra
list
numberlist
pairs
population
type

Remarks:
(none)

Inherited public predicates:
check_dimension_reducer/1 check_option/1 check_options/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 dimension_reducer_options/2 export_to_clauses/4 export_to_file/4 learn/2 option/2 option/3 print_dimension_reducer/1 transform/3 valid_dimension_reducer/1 valid_option/1 valid_options/1

- Public predicates
  - learn/3
- Protected predicates
- Private predicates
- Operators

Public predicates

learn/3

Learns a truncated singular value decomposition reducer from the given dataset object using the specified options.

Compilation flags:
    static

Template:
    learn(Dataset,DimensionReducer,Options)
Mode and number of proofs:
    learn(+object_identifier,-compound,+list(compound)) - one

---

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

pca_projection, random_projection

1.226 tsv

object

1.226.1 tsv

TSV files reading and writing predicates using the option Header-keep.

Availability:
logtalk_load(tsv(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2023-11-15

Compilation flags:
static, context_switching_calls

Extends:
public tsv(keep,false)

Remarks:
(none)

Inherited public predicates:
read_file/2 read_file/3 read_file_by_line/2 read_file_by_line/3 read_stream/2 read_stream/3 read_stream_by_line/2 read_stream_by_line/3 write_file/3 write_stream/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.226.2 tsv(Header)

- Header - Header handling option with possible values skip and keep (default).

Backward-compatible parametric object equivalent to using tsv(_Header_, false).

Availability:
logtalk_load(tsv(loader))

Author: Paulo Moura
Version: 1:0:1
Date: 2026-02-27

Compilation flags:
static, context_switching_calls

Extends:
public tsv(Header,false)

Remarks:
(none)

Inherited public predicates:
read_file/2 read_file/3 read_file_by_line/2 read_file_by_line/3 read_stream/2
read_stream/3 read_stream_by_line/2 read_stream_by_line/3 write_file/3 write_stream/3

- Public predicates
- Protected predicates

- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.226.3 tsv(Header,Comments)

- Header - Header handling option with possible values skip and keep (default).
- Comments - Comment handling option with possible values true and false (default). When true, lines starting with # are skipped when reading TSV files and streams.

TSV file and stream reading and writing predicates.

Availability:
    logtalk_load(tsv(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-02-25

Compilation flags:
    static, context_switching_calls

Implements:
    public tsv_protocol
Uses:
    list

logtalk
reader
type

Remarks:
(none)

Inherited public predicates:
read_file/2 read_file/3 read_file_by_line/2 read_file_by_line/3 read_stream/2
read_stream/3 read_stream_by_line/2 read_stream_by_line/3 write_file/3 write_stream/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.226.4 tsv_protocol

TSV file and stream reading and writing protocol.

Availability:
logtalk_load(tsv(loader))

Author: Paulo Moura
Version: 1:0:1
Date: 2025-05-07

Compilation flags:
static

Dependencies:
(none)

Remarks:

- Type-checking: Some of the predicate file and stream argument type-checking exceptions depend on the Prolog backend compliance with standards.

Inherited public predicates:
(none)

- Public predicates
  - read_file/3
  - read_stream/3
  - read_file/2
  - read_stream/2
  - read_file_by_line/3
  - read_stream_by_line/3
  - read_file_by_line/2
  - read_stream_by_line/2
  - write_file/3
  - write_stream/3
- Protected predicates

- Private predicates
- Operators

Public predicates

read_file/3

Reads a TSV file saving the data as clauses for the specified object predicate. Fails if the file cannot be parsed.

Compilation flags:
    static

Template:
    read_file(File,Object,Predicate)
Mode and number of proofs:
    read_file(+atom,+object_identifier,+predicate_indicator) - zero_or_one

Exceptions:
    File is a variable:
        instantiation_error
    File is neither a variable nor an atom:
        type_error(atom,File)
    File is an atom but not an existing file:
        existence_error(file,File)
    File is an existing file but cannot be opened for reading:
        permission_error(open,source_sink,File)
    Object is a variable:
        instantiation_error
    Object is neither a variable nor an object identifier:
        type_error(object_identifier,Object)
    Object is a valid object identifier but not an existing object:
        existence_error(object,Object)
    Predicate is a variable:
        instantiation_error
    Predicate is neither a variable nor a predicate indicator:
        type_error(predicate_indicator,Predicate)
    Predicate is a valid predicate indicator but not an existing public predicate:
        existence_error(predicate,Predicate)

read_stream/3

Reads a TSV stream saving the data as clauses for the specified object predicate. Fails if the stream cannot be parsed.

Compilation flags:
    static

Template:
    read_stream(Stream,Object,Predicate)
Mode and number of proofs:
    read_stream(+stream_or_alias,+object_identifier,+predicate_indicator) - zero_or_one

Exceptions:
    Stream is a variable:
        instantiation_error
    Stream is neither a variable nor a stream-term or alias:
        domain_error(stream_or_alias,Stream)
    Stream is not an open stream:
        existence_error(stream,Stream)
    Stream is an output stream:
        permission_error(input,stream,Stream)
    Stream is a binary stream:
        permission_error(input,binary_stream,Stream)
    Object is a variable:
        instantiation_error
    Object is neither a variable nor an object identifier:
        type_error(object_identifier,Object)
    Object is a valid object identifier but not an existing object:
        existence_error(object,Object)
    Predicate is a variable:
        instantiation_error
    Predicate is neither a variable nor a predicate indicator:
        type_error(predicate_indicator,Predicate)
    Predicate is a valid predicate indicator but not an existing public predicate:
        existence_error(predicate,Predicate)

read_file/2

Reads a TSV file returning the data as a list of rows, each row a list of fields. Fails if the file cannot be parsed.

Compilation flags:
    static

Template:
    read_file(File,Rows)
Mode and number of proofs:
    read_file(+atom,-list(list)) - zero_or_one

Exceptions:
    File is a variable:
        instantiation_error
    File is neither a variable nor an atom:
        type_error(atom,File)
    File is an atom but not an existing file:
        existence_error(file,File)
    File is an existing file but cannot be opened for reading:
        permission_error(open,source_sink,File)

---

read_stream/2

Reads a TSV stream returning the data as a list of rows, each row a list of fields. Fails if the stream cannot be parsed.

Compilation flags:
    static

Template:
    read_stream(Stream,Rows)
Mode and number of proofs:
    read_stream(+stream_or_alias,-list(list)) - zero_or_one

Exceptions:
    Stream is a variable:
        instantiation_error
    Stream is neither a variable nor a stream-term or alias:

domain_error(stream_or_alias,Stream)
Stream is not an open stream:
existence_error(stream,Stream)
Stream is an output stream:
permission_error(input,stream,Stream)
Stream is a binary stream:
permission_error(input,binary_stream,Stream)

---

read_file_by_line/3

Reads a TSV file saving the data as clauses for the specified object predicate. The file is read line by line. Fails if the file cannot be parsed.

Compilation flags:
static

Template:
read_file_by_line(File,Object,Predicate)
Mode and number of proofs:
read_file_by_line(+atom,+object_identifier,+predicate_indicator) - zero_or_one

Exceptions:
File is a variable:
instantiation_error
File is neither a variable nor an atom:
type_error(atom,File)
File is an atom but not an existing file:
existence_error(file,File)
File is an existing file but cannot be opened for reading:
permission_error(open,source_sink,File)
Object is a variable:
instantiation_error
Object is neither a variable nor an object identifier:
type_error(object_identifier,Object)
Object is a valid object identifier but not an existing object:
existence_error(object,Object)
Predicate is a variable:
instantiation_error
Predicate is neither a variable nor a predicate indicator:
type_error(predicate_indicator,Predicate)
Predicate is a valid predicate indicator but not an existing public predicate:
existence_error(predicate,Predicate)

read_stream_by_line/3

Reads a TSV stream saving the data as clauses for the specified object predicate. The stream is read line by line. Fails if the stream cannot be parsed.

Compilation flags:
    static

Template:
    read_stream_by_line(Stream,Object,Predicate)
Mode and number of proofs:
    read_stream_by_line(+stream_or_alias,+object_identifier,+predicate_indicator) - zero_or_one

Exceptions:
    Stream is a variable:
        instantiation_error
    Stream is neither a variable nor a stream-term or alias:
        domain_error(stream_or_alias,Stream)
    Stream is not an open stream:
        existence_error(stream,Stream)
    Stream is an output stream:
        permission_error(input,stream,Stream)
    Stream is a binary stream:
        permission_error(input,binary_stream,Stream)
    Object is a variable:
        instantiation_error
    Object is neither a variable nor an object identifier:
        type_error(object_identifier,Object)
    Object is a valid object identifier but not an existing object:
        existence_error(object,Object)
    Predicate is a variable:
        instantiation_error
    Predicate is neither a variable nor a predicate indicator:
        type_error(predicate_indicator,Predicate)
    Predicate is a valid predicate indicator but not an existing public predicate:
        existence_error(predicate,Predicate)

read_file_by_line/2

Reads a TSV file returning the data as a list of rows, each row a list of fields. The file is read line by line. Fails if the file cannot be parsed.

Compilation flags:
    static

Template:
    read_file_by_line(File,Rows)
Mode and number of proofs:
    read_file_by_line(+atom,-list(list)) - zero_or_one

Exceptions:
    File is a variable:
        instantiation_error
    File is neither a variable nor an atom:
        type_error(atom,File)
    File is an atom but not an existing file:
        existence_error(file,File)
    File is an existing file but cannot be opened for reading:
        permission_error(open,source_sink,File)

---

read_stream_by_line/2

Reads a TSV stream returning the data as a list of rows, each row a list of fields. The stream is read line by line. Fails if the stream cannot be parsed.

Compilation flags:
    static

Template:
    read_stream_by_line(Stream,Rows)
Mode and number of proofs:
    read_stream_by_line(+stream_or_alias,-list(list)) - zero_or_one

Exceptions:
    Stream is a variable:
        instantiation_error
    Stream is neither a variable nor a stream-term or alias:

domain_error(stream_or_alias,Stream)
Stream is not an open stream:
existence_error(stream,Stream)
Stream is an output stream:
permission_error(input,stream,Stream)
Stream is a binary stream:
permission_error(input,binary_stream,Stream)

---

write_file/3

Writes a TSV file with the data represented by the solutions to the specified object predicate.

Compilation flags:
static

Template:
write_file(File,Object,Predicate)
Mode and number of proofs:
write_file(+atom,+object_identifier,+predicate_indicator) - one

Exceptions:
File is a variable:
instantiation_error
File is neither a variable nor an atom:
type_error(atom,File)
File is an atom but cannot be opened for writing:
permission_error(open,source_sink,File)
Object is a variable:
instantiation_error
Object is neither a variable nor an object identifier:
type_error(object_identifier,Object)
Object is a valid object identifier but not an existing object:
existence_error(object,Object)
Predicate is a variable:
instantiation_error
Predicate is neither a variable nor a predicate indicator:
type_error(predicate_indicator,Predicate)
Predicate is a valid predicate indicator but not an existing public predicate:
existence_error(predicate,Predicate)

---

write_stream/3

Writes a TSV stream with the data represented by the solutions to the specified object predicate.

Compilation flags:
    static

Template:
    write_stream(Stream,Object,Predicate)
Mode and number of proofs:
    write_stream(+stream_or_alias,+object_identifier,+predicate_indicator) - one

Exceptions:
    Stream is a variable:
        instantiation_error
    Stream is neither a variable nor a stream-term or alias:
        domain_error(stream_or_alias,Stream)
    Stream is not an open stream:
        existence_error(stream,Stream)
    Stream is an input stream:
        permission_error(output,stream,Stream)
    Stream is a binary stream:
        permission_error(output,binary_stream,Stream)
    Object is a variable:
        instantiation_error
    Object is neither a variable nor an object identifier:
        type_error(object_identifier,Object)
    Object is a valid object identifier but not an existing object:
        existence_error(object,Object)
    Predicate is a variable:
        instantiation_error
    Predicate is neither a variable nor a predicate indicator:
        type_error(predicate_indicator,Predicate)
    Predicate is a valid predicate indicator but not an existing public predicate:
        existence_error(predicate,Predicate)

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.227 tutor

object

1.227.1 tutor

This object adds explanations and suggestions to selected compiler and developer tool warning and error messages.

Availability:
    logtalk_load(tutor(loader))

Author: Paulo Moura
Version: 0:86:0
Date: 2026-03-28

Compilation flags:
    static, context_switching_calls

Imports:
    public tutor_explanations
Provides:
    logtalk::message_hook/4
Uses:
    list
    logtalk

Remarks:

- Usage: Simply load this object at startup using the goal logtalk_load(tutor(loader)).

Inherited public predicates:
explain//1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.227.2 tutor_explanations

This category provides explanations and suggestions to selected compiler and developer tool warning and error messages.

Availability:
logtalk_load(tutor(loader))

Author: Paulo Moura
Version: 0:87:0
Date: 2026-03-30

Compilation flags:
static

Uses:
    list

Remarks:
    (none)

Inherited public predicates:
    (none)

- Public predicates
  - explain//1
- Protected predicates
- Private predicates
- Operators

Public predicates

explain//1

Generates an explanation for a message.

Compilation flags:
    static

Template:
    explain(Message)
Mode and number of proofs:
    explain(@callable) - zero_or_one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.228 types

object

1.228.1 atom

Atom data type predicates.

Availability:
logtalk_load(types(loader))

Author: Paulo Moura
Version: 1:9:0
Date: 2023-04-12

Compilation flags:
static, context_switching_calls

Extends:
public atomic
Uses:
user

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 check/1 depth/2 ground/1 new/1 numbervars/1 numbervars/3 occurs/2 singletons/2 subsumes/2 subterm/2 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
  - replace_sub_atom/4
  - split/3
- Protected predicates
- Private predicates
- Operators

Public predicates

replace_sub_atom/4

Replaces all occurrences of Old by New in Input returning Output. Returns Input if Old is the empty atom. Fails when Output does not unify with the resulting atom.

Compilation flags:
    static

Template:
    replace_sub_atom(Old,New,Input,Output)
Mode and number of proofs:
    replace_sub_atom(+atom,+atom,+atom,?atom) - zero_or_one

---

split/3

Splits an atom at a given delimiter into a list of sub-atoms.

Compilation flags:
    static

Template:
    split(Atom,Delimiter,SubAtoms)
Mode and number of proofs:
    split(+atom,+atom,-list(atom)) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.228.2 atomic

Atomic data type predicates.

Availability:
logtalk_load(types(loader))

Author: Paulo Moura
Version: 1:3:0
Date: 2018-07-11

Compilation flags:
static, context_switching_calls

Extends:
public term

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 check/1 depth/2 ground/1 new/1 numbervars/1 numbervars/3 occurs/2 singletons/2 subsumes/2 subterm/2 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
- Protected predicates

- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.228.3 callable

Callable term type predicates.

Availability:
logtalk_load(types(loader))

Author: Paulo Moura
Version: 1:4:0
Date: 2018-07-11

Compilation flags:
static, context_switching_calls

Extends:
public term

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 check/1 depth/2 ground/1 new/1 numbervars/1 numbervars/3 occurs/2 singletons/2 subsumes/2 subterm/2 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.228.4 character

Character predicates (most of them assume an ASCII representation).

Availability:
logtalk_load(types(loader))

Author: Paulo Moura
Version: 1:9:0
Date: 2019-06-29

Compilation flags:
static, context_switching_calls

Implements:
public characterp
Extends:
public atom

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 check/1 depth/2 ground/1 is_alpha/1 is_alphanumeric/1 is_ascii/1 is_bin_digit/1 is_control/1 is_dec_digit/1 is_end_of_line/1 is_hex_digit/1 is_layout/1 is_letter/1 is_lower_case/1 is_newline/1 is_octal_digit/1 is_period/1 is_punctuation/1 is_quote/1 is_upper_case/1 is_vowel/1 is_white_space/1 lower_upper/2 new/1 numbervars/1 numbervars/3 occurs/2 parenthesis/2 replace_sub_atom/4 singletons/2 split/3 subsumes/2 subterm/2 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.228.5 characterp

Character protocol.

Availability:
logtalk_load(types(loader))

Author: Paulo Moura
Version: 1:3:0
Date: 2019-06-29

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - is_ascii/1
  - is_alphanumeric/1
  - is_alpha/1
  - is_letter/1
  - is_bin_digit/1
  - is_octal_digit/1
  - is_dec_digit/1
  - is_hex_digit/1
  - is_lower_case/1

    - is_upper_case/1
    - is_vowel/1
    - is_white_space/1
    - is_layout/1
    - is_quote/1
    - is_punctuation/1
    - is_period/1
    - is_control/1
    - is_newline/1
    - is_end_of_line/1
    - parenthesis/2
    - lower_upper/2
- Protected predicates
- Private predicates
- Operators

Public predicates

is_ascii/1

True if the argument is an ASCII character.

Compilation flags:
    static

Template:
    is_ascii(Char)
Mode and number of proofs:
    is_ascii(+char) - zero_or_one

is_alphanumeric/1

True if the argument is an alphanumeric character.

Compilation flags:
    static

Template:
    is_alphanumeric(Char)
Mode and number of proofs:
    is_alphanumeric(+char) - zero_or_one

---

is_alpha/1

True if the argument is a letter or an underscore.

Compilation flags:
    static

Template:
    is_alpha(Char)
Mode and number of proofs:
    is_alpha(+char) - zero_or_one

---

is_letter/1

True if the argument is a letter.

Compilation flags:
    static

Template:
    is_letter(Char)
Mode and number of proofs:
    is_letter(+char) - zero_or_one

---

is_bin_digit/1

True if the argument is a binary digit.

Compilation flags:
    static

Template:
    is_bin_digit(Char)
Mode and number of proofs:
    is_bin_digit(+char) - zero_or_one

---

is_octal_digit/1

True if the argument is an octal digit.

Compilation flags:
    static

Template:
    is_octal_digit(Char)
Mode and number of proofs:
    is_octal_digit(+char) - zero_or_one

---

is_dec_digit/1

True if the argument is a decimal digit.

Compilation flags:
    static

Template:
    is_dec_digit(Char)
Mode and number of proofs:
    is_dec_digit(+char) - zero_or_one

---

is_hex_digit/1

True if the argument is an hexadecimal digit.

Compilation flags:
    static

Template:
    is_hex_digit(Char)
Mode and number of proofs:
    is_hex_digit(+char) - zero_or_one

---

is_lower_case/1

True if the argument is a lower case letter.

Compilation flags:
    static

Template:
    is_lower_case(Char)
Mode and number of proofs:
    is_lower_case(+char) - zero_or_one

---

is_upper_case/1

True if the argument is a upper case letter.

Compilation flags:
    static

Template:
    is_upper_case(Char)
Mode and number of proofs:
    is_upper_case(+char) - zero_or_one

---

is_vowel/1

True if the argument is a vowel.

Compilation flags:
    static

Template:
    is_vowel(Char)
Mode and number of proofs:
    is_vowel(+char) - zero_or_one

---

is_white_space/1

True if the argument is a white space character (a space or a tab) inside a line of characters.

Compilation flags:
    static

Template:
    is_white_space(Char)
Mode and number of proofs:
    is_white_space(+char) - zero_or_one

---

is_layout/1

True if the argument is a layout character.

Compilation flags:
    static

Template:
    is_layout(Char)
Mode and number of proofs:
    is_layout(+char) - zero_or_one

---

is_quote/1

True if the argument is a quote character.

Compilation flags:
    static

Template:
    is_quote(Char)
Mode and number of proofs:
    is_quote(+char) - zero_or_one

---

is_punctuation/1

True if the argument is a sentence punctuation character.

Compilation flags:
    static

Template:
    is_punctuation(Char)
Mode and number of proofs:
    is_punctuation(+char) - zero_or_one

---

is_period/1

True if the argument is a character that ends a sentence.

Compilation flags:
    static

Template:
    is_period(Char)
Mode and number of proofs:
    is_period(+char) - zero_or_one

---

is_control/1

True if the argument is an ASCII control character.

Compilation flags:
    static

Template:
    is_control(Char)
Mode and number of proofs:
    is_control(+char) - zero_or_one

---

is_newline/1

True if the argument is the ASCII newline character.

Compilation flags:
    static

Template:
    is_newline(Char)
Mode and number of proofs:
    is_newline(+char) - zero_or_one

---

is_end_of_line/1

True if the argument is the ASCII end-of-line character (either a carriage return or a line feed).

Compilation flags:
    static

Template:
    is_end_of_line(Char)
Mode and number of proofs:
    is_end_of_line(+char) - zero_or_one

---

parenthesis/2

Recognizes and converts between open and close parenthesis.

Compilation flags:
    static

Template:
    parenthesis(Char1,Char2)
Mode and number of proofs:
    parenthesis(?char,?char) - zero_or_more
    parenthesis(+char,?char) - zero_or_one
    parenthesis(?char,+char) - zero_or_one

---

lower_upper/2

Recognizes and converts between lower and upper case letters.

Compilation flags:
    static

Template:
    lower_upper(Char1,Char2)
Mode and number of proofs:
    lower_upper(?char,?char) - zero_or_more
    lower_upper(+char,?char) - zero_or_one
    lower_upper(?char,+char) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

character

protocol

1.228.6 comparingp

Comparing protocol using overloading of standard operators.

Availability:
logtalk_load(types(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2000-07-24

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - (<)/2
  - (=<)/2

  - (>)/2
  - (>=)/2
  - (=:=)/2
  - (=\=)/2
- Protected predicates
- Private predicates
- Operators

Public predicates

(<)/2

True if Term1 is less than Term2.

Compilation flags:
    static

Template:
    Term1<Term2
Mode and number of proofs:
    +term< +term - zero_or_one

(=<)/2

True if Term1 is less or equal than Term2.

Compilation flags:
    static

Template:
    Term1=<Term2
Mode and number of proofs:
    +term=< +term - zero_or_one

(>)/2

True if Term1 is greater than Term2.

Compilation flags:
    static

Template:
    Term1>Term2
Mode and number of proofs:
    +term> +term - zero_or_one

(>=)/2

True if Term1 is equal or grater than Term2.

Compilation flags:
    static

Template:
    Term1>=Term2
Mode and number of proofs:
    +term>= +term - zero_or_one

(=:=)/2

True if Term1 is equal to Term2.

Compilation flags:
    static

Template:
    Term1=:=Term2
Mode and number of proofs:
    +term=:= +term - zero_or_one

(=\=)/2

True if Term1 is not equal to Term2.

Compilation flags:
    static

Template:
    Term1=\=Term2
Mode and number of proofs:
    +term=\= +term - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

object

1.228.7 compound

Compound data type.

Availability:
    logtalk_load(types(loader))

Author: Paulo Moura
Version: 1:3:0
Date: 2018-07-11

Compilation flags:
    static, context_switching_calls

Extends:

public term

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 check/1 depth/2 ground/1 new/1 numbervars/1 numbervars/3 occurs/2 singletons/2 subsumes/2 subterm/2 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.228.8 difflist

Difference list predicates.

Availability:
logtalk_load(types(loader))

Author: Paulo Moura

Version: 2:1:0
Date: 2026-02-03

Compilation flags:
static, context_switching_calls

Implements:
public listp
Extends:
public compound
Uses:
list

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 append/2 append/3 check/1 delete/3 delete_matches/3 depth/2 drop/3 empty/1 flatten/2 ground/1 hamming_distance/3 keysort/2 last/2 length/2 max/2 member/2 memberchk/2 min/2 msort/2 msort/3 new/1 nextto/3 nth0/3 nth0/4 nth1/3 nth1/4 numbervars/1 numbervars/3 occurrences/2 occurrences/3 occurs/2 partition/5 permutation/2 prefix/2 prefix/3 proper_prefix/2 proper_prefix/3 proper_suffix/2 proper_suffix/3 remove_duplicates/2 reverse/2 same_length/2 same_length/3 select/3 select/4 selectchk/3 selectchk/4 sequential_occurrences/2 sequential_occurrences/3 singletons/2 sort/2 sort/3 sort/4 split/4 sublist/2 subsequence/3 subsequence/4 substitute/4 subsumes/2 subterm/2 subtract/3 suffix/2 suffix/3 take/3 take/4 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
  - add/3
  - as_list/2
- Protected predicates
- Private predicates
- Operators

Public predicates

add/3

Adds a term to the end of a difference list.

Compilation flags:
    static

Template:
    add(Term,DiffList,NewDiffList)
Mode and number of proofs:
    add(@term,+difference_list,-difference_list) - one

---

as_list/2

Returns a list with the elements of the difference list.

Compilation flags:
    static

Template:
    as_list(DiffList,List)
Mode and number of proofs:
    as_list(@difference_list,-list) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

list, list(Type), numberlist, varlist

object

1.228.9 float

Floating point numbers data type predicates.

Availability:
logtalk_load(types(loader))

Author: Paulo Moura
Version: 1:7:0
Date: 2025-02-25

Compilation flags:
static, context_switching_calls

Extends:
public number

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 =~= / 2 (>)/2 (>=)/2 approximately_equal/2 approximately_equal/3 check/1 depth/2 essentially_equal/3 ground/1 new/1 numbervars/1 numbervars/3 occurs/2 singletons/2 subsumes/2 subterm/2 tolerance_equal/4 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
  - between/4
  - sequence/4
  - sequence/5

- Protected predicates
- Private predicates
- Operators

Public predicates

between/4

Enumerates by backtracking a sequence of N equally spaced floats in the interval [Lower,Upper]. Assumes N > 0 and Lower =< Upper; fails otherwise.

Compilation flags:
    static

Template:
    between(Lower,Upper,N,Float)
Mode and number of proofs:
    between(+float,+float,+positive_integer,-float) - zero_or_more

---

sequence/4

Generates a list with the sequence of N equally spaced floats in the interval [Lower,Upper]. Assumes N > 0 and Lower =< Upper; fails otherwise.

Compilation flags:
    static

Template:
    sequence(Lower,Upper,N,List)
Mode and number of proofs:
    sequence(+float,+float,+positive_integer,-list(float)) - zero_or_one

---

sequence/5

Generates a list with the sequence of Step spaced floats in the interval [Lower,Upper]. Also returns the length of the list. Assumes Lower =< Upper; fails otherwise.

Compilation flags:
static

Template:
sequence(Lower,Upper,Step,List,Length)
Mode and number of proofs:
sequence(+float,+float,+float,-list(float),-positive_integer) - zero_or_one

---

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.228.10 integer

Integer data type predicates.

Availability:
logtalk_load(types(loader))

Author: Paulo Moura
Version: 1:56:0
Date: 2025-05-26

Compilation flags:
static, context_switching_calls

Extends:
public number

Remarks:

- Portability notes: This object will use the backend Prolog system between/3, plus/3, and succ/2 built-in predicates when available.

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 =~= / 2 (>)/2 (>=)/2 approximately_equal/2 approximately_equal/3 check/1 depth/2 essentially_equal/3 ground/1 new/1 numbervars/1 numbervars/3 occurs/2 singletons/2 subsumes/2 subterm/2 tolerance_equal/4 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
  - between/3
  - plus/3
  - succ/2
  - sequence/3
  - sequence/4
  - power_sequence/4
- Protected predicates
- Private predicates
- Operators

Public predicates

between/3

Returns integers in the interval defined by the two first arguments.

Compilation flags:
static

Template:
between(Lower,Upper,Integer)
Mode and number of proofs:
between(+integer,+integer,+integer) - zero_or_one
between(+integer,+integer,-integer) - zero_or_more

---

plus/3

Reversible integer sum. At least two of the arguments must be instantiated to integers.

Compilation flags:
    static

Template:
    plus(I,J,Sum)
Mode and number of proofs:
    plus(+integer,+integer,?integer) - zero_or_one
    plus(+integer,?integer,+integer) - zero_or_one
    plus(?integer,+integer,+integer) - zero_or_one

---

succ/2

Successor of a natural number. At least one of the arguments must be instantiated to a natural number.

Compilation flags:
    static

Template:
    succ(I,J)
Mode and number of proofs:
    succ(+integer,?integer) - zero_or_one
    succ(?integer,+integer) - zero_or_one

---

sequence/3

Generates a list with the sequence of all integers in the interval [Lower,Upper]. Assumes Lower =< Upper and fails otherwise.

Compilation flags:
    static

Template:
sequence(Lower,Upper,List)
Mode and number of proofs:
sequence(+integer,+integer,-list(integer)) - zero_or_one

---

sequence/4

Generates a list with the sequence of integers in the interval [Lower,Upper] by Step. Assumes Lower =< Upper, Step >= 1 and fails otherwise.

Compilation flags:
static

Template:
sequence(Lower,Upper,Step,List)
Mode and number of proofs:
sequence(+integer,+integer,+integer,-list(integer)) - zero_or_one

---

power_sequence/4

Generates a list of exponentiation results given [Lower,Upper] sequence of exponents and Base. Assumes Lower =< Upper and Base > 1 and fails otherwise.

Compilation flags:
static

Template:
power_sequence(Lower,Upper,Base,List)
Mode and number of proofs:
power_sequence(+integer,+integer,+integer,-list(integer)) - zero_or_one

---

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.228.11 list

List predicates.

Availability:
logtalk_load(types(loader))

Author: Paulo Moura
Version: 4:4:0
Date: 2026-02-03

Compilation flags:
static, context_switching_calls

Implements:
public listp
Extends:
public compound

Remarks:

- Portability notes: This object will use the backend Prolog system msort/2 and sort/4 built-in predicates when available.

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 append/2 append/3 check/1 delete/3 delete_matches/3 depth/2 drop/3 empty/1 flatten/2 ground/1 hamming_distance/3 keysort/2 last/2 length/2 max/2 member/2 memberchk/2 min/2 msort/2 msort/3 new/1 nextto/3 nth0/3 nth0/4 nth1/3 nth1/4 numbervars/1 numbervars/3 occurrences/2 occurrences/3 occurs/2 partition/5 permutation/2 prefix/2 prefix/3 proper_prefix/2 proper_prefix/3

proper_suffix/2 proper_suffix/3 remove_duplicates/2 reverse/2 same_length/2 same_length/3 select/3 select/4 selectchk/3 selectchk/4 sequential_occurrences/2 sequential_occurrences/3 singletons/2 sort/2 sort/3 sort/4 split/4 sublist/2 subsequence/3 subsequence/4 substitute/4 subsumes/2 subterm/2 subtract/3 suffix/2 suffix/3 take/3 take/4 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
  - as_difflist/2
- Protected predicates
- Private predicates
- Operators

Public predicates

as_difflist/2

Converts a list to a difference list.

Compilation flags:
static

Template:
as_difflist(List,Diffist)
Mode and number of proofs:
as_difflist(+list,-difference_list) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

list(Type), numberlist, varlist, difflist

object

1.228.12 list(Type)

List predicates with elements constrained to a single type.

Availability:
logtalk_load(types(loader))

Author: Paulo Moura
Version: 1:22:0
Date: 2018-07-11

Compilation flags:
static, context_switching_calls

Extends:
public list

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 append/2 append/3 as_difflist/2 check/1 delete/3 delete_matches/3 depth/2 drop/3 empty/1 flatten/2 ground/1 hamming_distance/3 keysort/2 last/2 length/2 max/2 member/2 memberchk/2 min/2 msort/2 msort/3 new/1 nextto/3 nth0/3 nth0/4 nth1/3 nth1/4 numbervars/1 numbervars/3 occurrences/2 occurrences/3 occurs/2 partition/5 permutation/2 prefix/2 prefix/3 proper_prefix/2 proper_prefix/3 proper_suffix/2 proper_suffix/3 remove_duplicates/2 reverse/2 same_length/2 same_length/3 select/3 select/4 selectchk/3 selectchk/4 sequential_occurrences/2 sequential_occurrences/3 singletons/2 sort/2 sort/3 sort/4 split/4 sublist/2 subsequence/3 subsequence/4 substitute/4 subsumes/2 subterm/2 subtract/3 suffix/2 suffix/3 take/3 take/4 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

list, numberlist, varlist, difflist

protocol

1.228.13 listp

List protocol.

Availability:
logtalk_load(types(loader))

Author: Paulo Moura
Version: 1:19:0
Date: 2026-02-03

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - append/2
  - append/3
  - delete/3
  - delete_matches/3
  - empty/1
  - flatten/2
  - hamming_distance/3
  - keysort/2
  - last/2
  - length/2
  - max/2
  - member/2
  - memberchk/2
  - min/2
  - msort/2
  - msort/3
  - nextto/3
  - nth0/3
  - nth0/4
  - nth1/3
  - nth1/4
  - sequential_occurrences/2
  - sequential_occurrences/3
  - occurrences/2
  - occurrences/3

    - partition/5
    - permutation/2
    - prefix/2
    - prefix/3
    - proper_prefix/2
    - proper_prefix/3
    - remove_duplicates/2
    - reverse/2
    - same_length/2
    - same_length/3
    - select/3
    - selectchk/3
    - select/4
    - selectchk/4
    - sort/2
    - sort/3
    - sort/4
    - split/4
    - sublist/2
    - subsequence/3
    - subsequence/4
    - substitute/4
    - subtract/3
    - suffix/2
    - suffix/3
    - proper_suffix/2
    - proper_suffix/3
    - take/3
    - take/4
    - drop/3
- Protected predicates
- Private predicates
- Operators

Public predicates

append/2

Appends all lists in a list of lists.

Compilation flags:
    static

Template:
    append(Lists,Concatenation)
Mode and number of proofs:
    append(+list(list),?list) - zero_or_one

---

append/3

Appends two lists.

Compilation flags:
    static

Template:
    append(List1,List2,List)
Mode and number of proofs:
    append(?list,?list,?list) - zero_or_more

---

delete/3

Deletes from a list all occurrences of an element returning the list of remaining elements. Uses ==/2 for element comparison.

Compilation flags:
    static

Template:
    delete(List,Element,Remaining)
Mode and number of proofs:

delete(@list,@term,?list) - one

---

delete_matches/3

Deletes all matching elements from a list, returning the list of remaining elements. Uses =/2 for element comparison.

Compilation flags:
static

Template:
delete_matches(List,Element,Remaining)
Mode and number of proofs:
delete_matches(@list,@term,?list) - one

---

empty/1

True if the argument is an empty list.

Compilation flags:
static

Template:
empty(List)
Mode and number of proofs:
empty(@list) - zero_or_one

---

flatten/2

Flattens a list of lists into a list.

Compilation flags:
static

Template:
    flatten(List,Flatted)
Mode and number of proofs:
    flatten(+list,-list) - one

---

hamming_distance/3

Calculates the Hamming distance between two lists (using equality to compare list elements). Fails if the two lists are not of the same length.

Compilation flags:
    static

Template:
    hamming_distance(List1,List2,Distance)
Mode and number of proofs:
    hamming_distance(+list,+list,-integer) - zero_or_one

---

keysort/2

Sorts a list of key-value pairs in ascending order.

Compilation flags:
    static

Template:
    keysort(List,Sorted)
Mode and number of proofs:
    keysort(+list(pair),-list(pair)) - one

---

last/2

List last element (if it exists).

Compilation flags:
    static

Template:
    last(List,Last)
Mode and number of proofs:
    last(?list,?term) - zero_or_more

---

length/2

List length.

Compilation flags:
    static

Template:
    length(List,Length)
Mode and number of proofs:
    length(?list,?integer) - zero_or_more

---

max/2

Determines the list maximum value using standard order. Fails if the list is empty.

Compilation flags:
    static

Template:
    max(List,Maximum)
Mode and number of proofs:
    max(+list,-term) - zero_or_one

---

member/2

Element is a list member.

Compilation flags:
    static

Template:
    member(Element,List)
Mode and number of proofs:
    member(?term,?list) - zero_or_more

---

memberchk/2

Checks if a term is a member of a list.

Compilation flags:
    static

Template:
    memberchk(Element,List)
Mode and number of proofs:
    memberchk(?term,?list) - zero_or_one

---

min/2

Determines the minimum value in a list using standard order. Fails if the list is empty.

Compilation flags:
    static

Template:
    min(List,Minimum)
Mode and number of proofs:
    min(+list,-term) - zero_or_one

---

msort/2

Sorts a list in ascending order (duplicated elements are not removed).

Compilation flags:
    static

Template:
    msort(List,Sorted)
Mode and number of proofs:
    msort(+list,-list) - one

---

msort/3

Sorts a list using a user-specified comparison predicate modeled on the standard compare/3 predicate (duplicated elements are not removed).

Compilation flags:
    static

Template:
    msort(Closure,List,Sorted)
Meta-predicate template:
    msort(3,*,*)
Mode and number of proofs:
    msort(+callable,+list,-list) - one

---

nextto/3

X and Y are consecutive elements in List.

Compilation flags:
    static

Template:
    nextto(X,Y,List)
Mode and number of proofs:

nextto(?term,?term,?list) - zero_or_more

---

nth0/3

Nth element of a list (counting from zero).

Compilation flags:
static

Template:
nth0(Nth,List,Element)
Mode and number of proofs:
nth0(?integer,?list,?term) - zero_or_more

---

nth0/4

Nth element of a list (counting from zero). Rest is a list of all the other elements. Can be used to either select the nth element of List or to insert an element before the nth element in Rest.

Compilation flags:
static

Template:
nth0(Nth,List,Element,Rest)
Mode and number of proofs:
nth0(?integer,?list,?term,?list) - zero_or_more

---

nth1/3

Nth element of a list (counting from one).

Compilation flags:
static

Template:
nth1(Nth,List,Element)
Mode and number of proofs:
nth1(?integer,?list,?term) - zero_or_more

---

nth1/4

Nth element of a list (counting from one). Rest is a list of all the other elements. Can be used to either select the nth element of List or to insert an element before the nth element in Rest.

Compilation flags:
static

Template:
nth1(Nth,List,Element,Rest)
Mode and number of proofs:
nth1(?integer,?list,?term,?list) - zero_or_more

---

sequential_occurrences/2

Counts the number of sequential occurrences of each List element, unifying Occurrences with a list of Element-Count pairs. Uses term equality for element comparison.

Compilation flags:
static

Template:
sequential_occurrences(List,Occurrences)
Mode and number of proofs:
sequential_occurrences(@list,-list(pair(term,positive_integer))) - one

---

sequential_occurrences/3

Counts the number of sequential occurrences of each List element, unifying Occurrences with a list of Element-Count pairs. Uses Closure for element comparison.

Compilation flags:
    static

Template:
    sequential_occurrences(List,Closure,Occurrences)
Mode and number of proofs:
    sequential_occurrences(@list,@callable,-list(pair(term,positive_integer))) - one

---

occurrences/2

Counts the number of occurrences of each List element, unifying Occurrences with a sorted list of Element-Count pairs. Uses term equality for element comparison.

Compilation flags:
    static

Template:
    occurrences(List,Occurrences)
Mode and number of proofs:
    occurrences(@list,-list(pair(term,positive_integer))) - one

---

occurrences/3

Counts the number of occurrences of each List element, unifying Occurrences with a sorted list of Element-Count pairs. Uses Closure for element comparison.

Compilation flags:
    static

Template:
    occurrences(List,Closure,Occurrences)
Meta-predicate template:

occurrences(*,2,*)
Mode and number of proofs:
occurrences(@list,@callable,-list(pair(term,positive_integer))) - one

---

partition/5

Partitions a list in lists with values less, equal, and greater than a given value (using standard order).

Compilation flags:
static

Template:
partition(List,Value,Less,Equal,Greater)
Mode and number of proofs:
partition(+list,+number,-list,-list,-list) - one

---

permutation/2

The two lists are a permutation of the same list.

Compilation flags:
static

Template:
permutation(List,Permutation)
Mode and number of proofs:
permutation(?list,?list) - zero_or_more

---

prefix/2

Prefix is a prefix of List.

Compilation flags:
    static

Template:
    prefix(Prefix,List)
Mode and number of proofs:
    prefix(?list,+list) - zero_or_more

---

prefix/3

Prefix is a prefix of length Length of List.

Compilation flags:
    static

Template:
    prefix(Prefix,Length,List)
Mode and number of proofs:
    prefix(?list,+integer,+list) - zero_or_one
    prefix(?list,-integer,+list) - zero_or_more

---

proper_prefix/2

Prefix is a proper prefix of List.

Compilation flags:
    static

Template:
    proper_prefix(Prefix,List)
Mode and number of proofs:
    proper_prefix(?list,+list) - zero_or_more

---

proper_prefix/3

Prefix is a proper prefix of length Length of List.

Compilation flags:
    static

Template:
    proper_prefix(Prefix,Length,List)
Mode and number of proofs:
    proper_prefix(?list,+integer,+list) - zero_or_one
    proper_prefix(?list,-integer,+list) - zero_or_more

---

remove_duplicates/2

Removes duplicated list elements using equality (==/2) for comparison and keeping the left-most element when repeated.

Compilation flags:
    static

Template:
    remove_duplicates(List,Set)
Mode and number of proofs:
    remove_duplicates(+list,-list) - one

---

reverse/2

Reverses a list.

Compilation flags:
    static

Template:

reverse(List,Reversed)
Mode and number of proofs:
reverse(+list,?list) - zero_or_one
reverse(?list,+list) - zero_or_one
reverse(-list,-list) - one_or_more

---

same_length/2

The two lists have the same length.

Compilation flags:
static

Template:
same_length(List1,List2)
Mode and number of proofs:
same_length(+list,?list) - zero_or_one
same_length(?list,+list) - zero_or_one
same_length(-list,-list) - one_or_more

---

same_length/3

The two lists have the same length.

Compilation flags:
static

Template:
same_length(List1,List2,Length)
Mode and number of proofs:
same_length(+list,?list,?integer) - zero_or_one
same_length(?list,+list,?integer) - zero_or_one
same_length(-list,-list,-integer) - one_or_more

---

select/3

Selects an element from a list, returning the list of remaining elements.

Compilation flags:
    static

Template:
    select(Element,List,Remaining)
Mode and number of proofs:
    select(?term,?list,?list) - zero_or_more

---

selectchk/3

Checks that an element can be selected from a list, returning the list of remaining elements.

Compilation flags:
    static

Template:
    selectchk(Element,List,Remaining)
Mode and number of proofs:
    selectchk(?term,?list,?list) - zero_or_one

---

select/4

Selects an element from a list, replacing it by a new element and returning the resulting list.

Compilation flags:
    static

Template:
    select(Old,OldList,New,NewList)
Mode and number of proofs:
    select(?term,?list,?term,?list) - zero_or_more

---

selectchk/4

Checks that an element from a list can be replaced by a new element, returning the resulting list.

Compilation flags:
    static

Template:
    selectchk(Old,OldList,New,NewList)
Mode and number of proofs:
    selectchk(?term,?list,?term,?list) - zero_or_one

---

sort/2

Sorts a list in ascending order (duplicated elements are removed).

Compilation flags:
    static

Template:
    sort(List,Sorted)
Mode and number of proofs:
    sort(+list,-list) - one

---

sort/3

Sorts a list using a user-specified comparison predicate modeled on the standard compare/3 predicate (duplicated elements are removed).

Compilation flags:
    static

Template:
    sort(Closure,List,Sorted)
Meta-predicate template:
    sort(3,*,*)
Mode and number of proofs:

sort(+callable,+list,-list) - one

---

sort/4

Sorts a list using the given key and order. Uses the standard term comparison operators for the order. The key selects the argument in each element in the list to use for comparisons. A key value of zero uses the whole element for comparisons.

Compilation flags:
static

Template:
sort(Key,Order,List,Sorted)
Mode and number of proofs:
sort(+non_negative_integer,+atom,+list,-list) - one

Remarks:

- Removing duplicates: Use one of the @< or @> orders.
- Keeping duplicates: Use one of the @=< or @>= orders.
- Sorting in ascending order: Use one of the @< or @=< orders.
- Sorting in descending order: Use one of the @> or @>= orders.

---

split/4

Splits a list into sublists of a given length. Also returns a list with the remaining elements. Fails if the length is zero or negative.

Compilation flags:
static

Template:
split(List,Length,Sublists,Remaining)
Mode and number of proofs:
split(+list,+integer,-list(list),-list) - zero_or_one

---

sublist/2

The first list is a sublist of the second.

Compilation flags:
    static

Template:
    sublist(Sublist,List)
Mode and number of proofs:
    sublist(?list,+list) - zero_or_more

---

subsequence/3

List is an interleaving of Subsequence and Remaining. Element order is preserved.

Compilation flags:
    static

Template:
    subsequence(List,Subsequence,Remaining)
Mode and number of proofs:
    subsequence(?list,?list,?list) - zero_or_more

---

subsequence/4

Generates subsequences of a given length from a list. Also returns the remaining elements. Element order is preserved.

Compilation flags:
    static

Template:
    subsequence(List,Length,Subsequence,Remaining)
Mode and number of proofs:
    subsequence(+list,+integer,?list,?list) - zero_or_more

---

substitute/4

Substitutes all occurrences of Old in List by New, returning NewList. Uses term equality for element comparison.

Compilation flags:
    static

Template:
    substitute(Old,List,New,NewList)
Mode and number of proofs:
    substitute(@term,@list,@term,-list) - one

---

subtract/3

Removes all elements in the second list from the first list, returning the list of remaining elements.

Compilation flags:
    static

Template:
    subtract(List,Elements,Remaining)
Mode and number of proofs:
    subtract(+list,+list,-list) - one

---

suffix/2

Suffix is a suffix of List.

Compilation flags:
    static

Template:
    suffix(Suffix,List)

Mode and number of proofs:
suffix(?list,+list) - zero_or_more

---

suffix/3

Suffix is a suffix of length Length of List.

Compilation flags:
static

Template:
suffix(Suffix,Length,List)
Mode and number of proofs:
suffix(?list,+integer,+list) - zero_or_one
suffix(?list,-integer,+list) - zero_or_more

---

proper_suffix/2

Suffix is a proper suffix of List.

Compilation flags:
static

Template:
proper_suffix(Suffix,List)
Mode and number of proofs:
proper_suffix(?list,+list) - zero_or_more

---

proper_suffix/3

Suffix is a proper suffix of length Length of List.

Compilation flags:
    static

Template:
    proper_suffix(Suffix,Length,List)
Mode and number of proofs:
    proper_suffix(?list,+integer,+list) - zero_or_one
    proper_suffix(?list,-integer,+list) - zero_or_more

---

take/3

Takes the first N elements of a list. Fails if the list have fewer than N elements.

Compilation flags:
    static

Template:
    take(N,List,Elements)
Mode and number of proofs:
    take(+integer,+list,-list) - zero_or_one

---

take/4

Takes the first N elements of a list. Fails if the list have fewer than N elements.

Compilation flags:
    static

Template:
    take(N,List,Elements,Remaining)
Mode and number of proofs:
    take(+integer,+list,-list,-list) - zero_or_one

drop/3

Drops the first N elements of a list. Fails if the list have fewer than N elements.

Compilation flags:
    static

Template:
    drop(N,List,Remaining)
Mode and number of proofs:
    drop(+integer,+list,-list) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

list, list(Type), numberlistp, varlistp

object

1.228.14 natural

Natural numbers data type predicates.

Availability:
    logtalk_load(types(loader))

Author: Paulo Moura

Version: 2:3:0
Date: 2026-05-13

Compilation flags:
static, context_switching_calls

Extends:
public integer
Uses:
list
numberlist

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 =~= / 2 (>)/2 (>=)/2 approximately_equal/2 approximately_equal/3 between/3 check/1 depth/2 essentially_equal/3 ground/1 new/1 numbervars/1 numbervars/3 occurs/2 plus/3 power_sequence/4 sequence/3 sequence/4 singletons/2 subsumes/2 subterm/2 succ/2 tolerance_equal/4 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
  - factorial/2
  - fibonacci/2
  - lucas/2
  - falling_factorial/3
  - rising_factorial/3
  - catalan/2
  - subfactorial/2
  - bell/2
  - stirling_first/3
  - stirling_second/3
  - partition_number/2
  - binomial/3
  - multinomial/2
- Protected predicates
- Private predicates

- Operators

Public predicates

factorial/2

Computes the factorial of a non-negative integer.

Compilation flags:
    static

Template:
    factorial(N,Factorial)
Mode and number of proofs:
    factorial(+non_negative_integer,-non_negative_integer) - one

---

fibonacci/2

Computes the Fibonacci number of a non-negative integer.

Compilation flags:
    static

Template:
    fibonacci(N,Fibonacci)
Mode and number of proofs:
    fibonacci(+non_negative_integer,-non_negative_integer) - one

---

lucas/2

Computes the Lucas number of a non-negative integer.

Compilation flags:
    static

Template:
lucas(N,Lucas)
Mode and number of proofs:
lucas(+non_negative_integer,-non_negative_integer) - one

---

falling_factorial/3

Computes the falling factorial. N must be greater than or equal to K (fails otherwise).

Compilation flags:
static

Template:
falling_factorial(N,K,FallingFactorial)
Mode and number of proofs:
falling_factorial(+non_negative_integer,+non_negative_integer,-non_negative_integer) - zero_or_one

---

rising_factorial/3

Computes the rising factorial.

Compilation flags:
static

Template:
rising_factorial(N,K,RisingFactorial)
Mode and number of proofs:
rising_factorial(+non_negative_integer,+non_negative_integer,-non_negative_integer) - one

---

catalan/2

Computes the Catalan number of a non-negative integer.

Compilation flags:
    static

Template:
    catalan(N,Catalan)
Mode and number of proofs:
    catalan(+non_negative_integer,-non_negative_integer) - one

---

subfactorial/2

Computes the subfactorial of a non-negative integer.

Compilation flags:
    static

Template:
    subfactorial(N,Subfactorial)
Mode and number of proofs:
    subfactorial(+non_negative_integer,-non_negative_integer) - one

---

bell/2

Computes the Bell number of a non-negative integer.

Compilation flags:
    static

Template:
    bell(N,Bell)
Mode and number of proofs:
    bell(+non_negative_integer,-non_negative_integer) - one

---

stirling_first/3

Computes the unsigned Stirling number of the first kind. N must be greater than or equal to K (fails otherwise).

Compilation flags:
    static

Template:
    stirling_first(N,K,Stirling)
Mode and number of proofs:
    stirling_first(+non_negative_integer,+non_negative_integer,-non_negative_integer) - zero_or_one

---

stirling_second/3

Computes the Stirling number of the second kind. N must be greater than or equal to K (fails otherwise).

Compilation flags:
    static

Template:
    stirling_second(N,K,Stirling)
Mode and number of proofs:
    stirling_second(+non_negative_integer,+non_negative_integer,-non_negative_integer) - zero_or_one

---

partition_number/2

Computes the partition number of a non-negative integer.

Compilation flags:
    static

Template:
    partition_number(N,PartitionNumber)
Mode and number of proofs:
    partition_number(+non_negative_integer,-non_negative_integer) - one

binomial/3

Computes the binomial coefficient. N must be greater than or equal to K (fails otherwise).

Compilation flags:
    static

Template:
    binomial(N,K,Binomial)
Mode and number of proofs:
    binomial(+non_negative_integer,+non_negative_integer,-non_negative_integer) - zero_or_one

---

multinomial/2

Computes the multinomial coefficient for a list of non-negative integers.

Compilation flags:
    static

Template:
    multinomial(Integers,Multinomial)
Mode and number of proofs:
    multinomial(+list(non_negative_integer),-non_negative_integer) - one

---

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.228.15 number

Number data type predicates.

Availability:
logtalk_load(types(loader))

Author: Paulo Moura
Version: 1:15:0
Date: 2026-02-11

Compilation flags:
static, context_switching_calls

Extends:
public atomic

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 check/1 depth/2 ground/1 new/1 numbervars/1 numbervars/3 occurs/2 singletons/2 subsumes/2 subterm/2 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
  - approximately_equal/2
  - approximately_equal/3
  - essentially_equal/3
  - tolerance_equal/4

    - =~= / 2
- Protected predicates
- Private predicates
- Operators
    - op(700,xfx,=~=)

Public predicates

approximately_equal/2

Compares two numbers for approximate equality given the epsilon arithmetic constant value using the de facto standard formula abs(Number1 - Number2) =< max(abs(Number1), abs(Number2)) * epsilon. No type-checking.

Compilation flags:
    static

Template:
    approximately_equal(Number1,Number2)
Mode and number of proofs:
    approximately_equal(+number,+number) - zero_or_one

---

approximately_equal/3

Compares two numbers for approximate equality given a user-defined epsilon value using the de facto standard formula abs(Number1 - Number2) =< max(abs(Number1), abs(Number2)) * Epsilon. No type-checking.

Compilation flags:
    static

Template:
    approximately_equal(Number1,Number2,Epsilon)
Mode and number of proofs:
    approximately_equal(+number,+number,+number) - zero_or_one

Remarks:

- Epsilon range: Epsilon should be the epsilon arithmetic constant value or a small multiple of it. Only use a larger value if a greater error is expected.
- Comparison with essential equality: For the same epsilon value, approximate equality is weaker requirement than essential equality.

---

essentially_equal/3

Compares two numbers for essential equality given an epsilon value using the de facto standard formula abs(Number1 - Number2) =< min(abs(Number1), abs(Number2)) * Epsilon. No type-checking.

Compilation flags:
    static

Template:
    essentially_equal(Number1,Number2,Epsilon)
Mode and number of proofs:
    essentially_equal(+number,+number,+number) - zero_or_one

Remarks:

- Comparison with approximate equality: For the same epsilon value, essential equality is a stronger requirement than approximate equality.

---

tolerance_equal/4

Compares two numbers for close equality given relative and absolute tolerances using the de facto standard formula abs(Number1 - Number2) =< max(RelativeTolerance * max(abs(Number1), abs(Number2)), AbsoluteTolerance). No type-checking.

Compilation flags:
    static

Template:
    tolerance_equal(Number1,Number2,RelativeTolerance,AbsoluteTolerance)
Mode and number of proofs:
    tolerance_equal(+number,+number,+number,+number) - zero_or_one

---

=~= / 2

Compares two floats (or lists of floats) for approximate equality using 100*epsilon for the absolute error and, if that fails, 99.999% accuracy for the relative error. Note that these precision values may not be adequate for all cases. No type-checking.

Compilation flags:
    static

Template:
    =~=(Float1,Float2)
Mode and number of proofs:
    =~=(+number,+number) - zero_or_one
    =~=(+list(number),+list(number)) - zero_or_one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

op(700,xfx,=~=)

Scope:
    public

object

1.228.16 numberlist

List of numbers predicates.

Availability:
    logtalk_load(types(loader))

Author: Paulo Moura

Version: 1:20:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Implements:
public numberlistp
Extends:
public list
Uses:
list

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 append/2 append/3 as_difflist/2 average/2 chebyshev_distance/3 chebyshev_norm/2 check/1 delete/3 delete_matches/3 depth/2 drop/3 empty/1 euclidean_distance/3 euclidean_norm/2 flatten/2 ground/1 hamming_distance/3 keysort/2 last/2 least_common_multiple/2 length/2 linear_regression/4 manhattan_distance/3 manhattan_norm/2 max/2 median/2 member/2 memberchk/2 min/2 min_max/3 minkowski_distance/4 modes/2 msort/2 msort/3 new/1 nextto/3 normalize_range/2 normalize_range/4 normalize_scalar/2 normalize_unit/2 nth0/3 nth0/4 nth1/3 nth1/4 numbervars/1 numbervars/3 occurrences/2 occurrences/3 occurs/2 partition/5 permutation/2 prefix/2 prefix/3 product/2 proper_prefix/2 proper_prefix/3 proper_suffix/2 proper_suffix/3 remove_duplicates/2 rescale/3 reverse/2 same_length/2 same_length/3 scalar_product/3 select/3 select/4 selectchk/3 selectchk/4 sequential_occurrences/2 sequential_occurrences/3 singletons/2 softmax/2 softmax/3 sort/2 sort/3 sort/4 split/4 sublist/2 subsequence/3 subsequence/4 substitute/4 subsumes/2 subterm/2 subtract/3 suffix/2 suffix/3 sum/2 take/3 take/4 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

list, list(Type), varlist, difflist

protocol

1.228.17 numberlistp

List of numbers protocol.

Availability:
logtalk_load(types(loader))

Author: Paulo Moura
Version: 1:13:0
Date: 2026-05-06

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:

(none)

- Public predicates
  - min/2
  - max/2
  - min_max/3
  - product/2
  - sum/2
  - average/2
  - median/2
  - modes/2
  - euclidean_norm/2
  - chebyshev_norm/2
  - manhattan_norm/2
  - euclidean_distance/3
  - chebyshev_distance/3
  - manhattan_distance/3
  - minkowski_distance/4
  - scalar_product/3
  - normalize_range/2
  - normalize_range/4
  - normalize_unit/2
  - normalize_scalar/2
  - rescale/3
  - least_common_multiple/2
  - softmax/2
  - softmax/3
  - linear_regression/4
- Protected predicates
- Private predicates
- Operators

Public predicates

min/2

Determines the minimum value in a list using arithmetic order. Fails if the list is empty.

Compilation flags:
    static

Template:
    min(List,Minimum)
Mode and number of proofs:
    min(+list(number),-number) - zero_or_one

---

max/2

Determines the list maximum value using arithmetic order. Fails if the list is empty.

Compilation flags:
    static

Template:
    max(List,Maximum)
Mode and number of proofs:
    max(+list(number),-number) - zero_or_one

---

min_max/3

Determines the minimum and maximum values in a list using arithmetic order. Fails if the list is empty.

Compilation flags:
    static

Template:
    min_max(List,Minimum,Maximum)
Mode and number of proofs:
    min_max(+list(number),-number,-number) - zero_or_one

---

product/2

Calculates the product of all list numbers. Fails if the list is empty.

Compilation flags:
    static

Template:
    product(List,Product)
Mode and number of proofs:
    product(+list(number),-number) - zero_or_one

---

sum/2

Calculates the sum of all list numbers. Returns the integer zero if the list is empty.

Compilation flags:
    static

Template:
    sum(List,Sum)
Mode and number of proofs:
    sum(+list(number),-number) - one

---

average/2

Calculates the average (i.e., arithmetic mean) of a list of numbers. Fails if the list is empty.

Compilation flags:
    static

Template:
    average(List,Average)
Mode and number of proofs:

average(+list(number),-float) - zero_or_one

---

median/2

Calculates the median of a list of numbers. Fails if the list is empty.

Compilation flags:
static

Template:
median(List,Median)
Mode and number of proofs:
median(+list(number),-float) - zero_or_one

---

modes/2

Returns the list of modes of a list of numbers in ascending order. Fails if the list is empty.

Compilation flags:
static

Template:
modes(List,Modes)
Mode and number of proofs:
modes(+list(number),-list(number)) - zero_or_one

---

euclidean_norm/2

Calculates the Euclidean norm of a list of numbers. Fails if the list is empty.

Compilation flags:
static

Template:

euclidean_norm(List,Norm)
Mode and number of proofs:
euclidean_norm(+list(number),-float) - zero_or_one

---

chebyshev_norm/2

Calculates the Chebyshev norm of a list of numbers. Fails if the list is empty.

Compilation flags:
static

Template:
chebyshev_norm(List,Norm)
Mode and number of proofs:
chebyshev_norm(+list(integer),-integer) - zero_or_one
chebyshev_norm(+list(float),-float) - zero_or_one

---

manhattan_norm/2

Calculates the Manhattan norm of a list of numbers. Fails if the list is empty.

Compilation flags:
static

Template:
manhattan_norm(List,Norm)
Mode and number of proofs:
manhattan_norm(+list(integer),-integer) - zero_or_one
manhattan_norm(+list(float),-float) - zero_or_one

---

euclidean_distance/3

Calculates the Euclidean distance between two lists of numbers. Fails if the two lists are empty or not of the same length.

Compilation flags:
    static

Template:
    euclidean_distance(List1,List2,Distance)
Mode and number of proofs:
    euclidean_distance(+list(number),+list(number),-float) - zero_or_one

---

chebyshev_distance/3

Calculates the Chebyshev distance between two lists of numbers. Fails if the two lists are empty or not of the same length.

Compilation flags:
    static

Template:
    chebyshev_distance(List1,List2,Distance)
Mode and number of proofs:
    chebyshev_distance(+list(integer),+list(integer),-integer) - zero_or_one
    chebyshev_distance(+list(float),+list(float),-float) - zero_or_one

---

manhattan_distance/3

Calculates the Manhattan distance between two lists of numbers. Fails if the two lists are empty or not of the same length.

Compilation flags:
    static

Template:
    manhattan_distance(List1,List2,Distance)

Mode and number of proofs:
manhattan_distance(+list(integer),+list(integer),-integer) - zero_or_one
manhattan_distance(+list(float),+list(float),-float) - zero_or_one

---

minkowski_distance/4

Calculates the Minkowski distance between two lists of numbers given a positive power. Fails if the two lists are empty or not of the same length.

Compilation flags:
static

Template:
minkowski_distance(List1,List2,Power,Distance)
Mode and number of proofs:
minkowski_distance(+list(number),+list(number),+number,-float) - zero_or_one

---

scalar_product/3

Calculates the scalar product of two lists of numbers. Fails if the two lists are empty or not of the same length.

Compilation flags:
static

Template:
scalar_product(List1,List2,Product)
Mode and number of proofs:
scalar_product(+list(integer),+list(integer),-integer) - zero_or_one
scalar_product(+list(float),+list(float),-float) - zero_or_one

---

normalize_range/2

Normalizes a list of numbers into the [0.0,1.0] range. Caller must handle arithmetic exceptions if the input list if not normalizable.

Compilation flags:
    static

Template:
    normalize_range(List,NormalizedList)
Mode and number of proofs:
    normalize_range(+list(number),-list(float)) - one

---

normalize_range/4

Normalizes a list of numbers into the given range. Caller must handle arithmetic exceptions if the input list if not normalizable.

Compilation flags:
    static

Template:
    normalize_range(List,Minimum,Maximum,NormalizedList)
Mode and number of proofs:
    normalize_range(+list(number),+number,+number,-list(float)) - one

---

normalize_unit/2

Normalizes a list of numbers returning its unit vector (i.e., a list with Euclidean norm equal to one). Caller must handle arithmetic exceptions if the input list if not normalizable.

Compilation flags:
    static

Template:
    normalize_unit(List,NormalizedList)
Mode and number of proofs:

normalize_unit(+list(number),-list(float)) - one

---

normalize_scalar/2

Normalizes a list of numbers such that the sum of all numbers is equal to one. Caller must handle arithmetic exceptions if the input list if not normalizable.

Compilation flags:
static

Template:
normalize_scalar(List,NormalizedList)
Mode and number of proofs:
normalize_scalar(+list(number),-list(float)) - one

---

rescale/3

Rescales all numbers in a list by the given factor.

Compilation flags:
static

Template:
rescale(List,Factor,RescaledList)
Mode and number of proofs:
rescale(+list(integer),+integer,-list(integer)) - one
rescale(+list(number),+float,-list(float)) - one

---

least_common_multiple/2

Computes the least common multiple of a list of two or more positive integers. Fails if the list is empty or contains a single element. Fails also if any of the elements is zero. May require backend support for unbound integer arithmetic.

Compilation flags:
    static

Template:
    least_common_multiple(Integers,LeastCommonMultiple)
Mode and number of proofs:
    least_common_multiple(+list(positive_integer),-positive_integer) - zero_or_one

---

softmax/2

Computes the softmax of a list of floats, returning a probability distribution. The implementation is numerically stable.

Compilation flags:
    static

Template:
    softmax(Floats,Softmax)
Mode and number of proofs:
    softmax(+list(float),-list(float)) - one

---

softmax/3

Computes the softmax of a list of floats with the given temperature, returning a probability distribution. The implementation is numerically stable.

Compilation flags:
    static

Template:
    softmax(Floats,Temperature,Softmax)

Mode and number of proofs:
softmax(+list(float),+positive_float,-list(float)) - one

Remarks:

- Temperature > 1.0: Makes the distribution more uniform.
- Temperature < 1.0: Makes the distribution more concentrated on the largest values.
- Temperature = 1.0: Standard softmax behavior.

---

linear_regression/4

Computes the simple linear regression for the given lists of X and Y coordinates, returning the slope and the intercept. Fails if the lists have less than two elements, are not of the same length, or if all X values are the same.

Compilation flags:
static

Template:
linear_regression(Xs,Ys,Slope,Intercept)
Mode and number of proofs:
linear_regression(+list(number),+list(number),-float,-float) - zero_or_one

---

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

numberlist, listp, varlistp

object

1.228.18 pairs

Useful predicates over lists of pairs (key-value terms).

Availability:
logtalk_load(types(loader))

Author: Paulo Moura
Version: 2:1:1
Date: 2023-11-21

Compilation flags:
static, context_switching_calls

Dependencies:
(none)

Remarks:

- Usage: This object can be loaded independently of other entities in the types library by using the goal logtalk_load(types(pairs)).

Inherited public predicates:
(none)

- Public predicates
  - keys_values/3
  - keys/2
  - key/2
  - values/2
  - value/3
  - transpose/2
  - group_sorted_by_key/2
  - group_consecutive_by_key/2
  - group_by_key/2
  - map/3
- Protected predicates

- Private predicates
- Operators

Public predicates

keys_values/3

Converts between a list of pairs and lists of keys and values. When converting to pairs, this predicate fails if the list of keys and the list of values have different lengths.

Compilation flags:
    static

Template:
    keys_values(Pairs,Keys,Values)
Mode and number of proofs:
    keys_values(+list(pair),-list,-list) - one
    keys_values(-list(pair),+list,+list) - zero_or_one

---

keys/2

Returns a list of keys from a list of pairs.

Compilation flags:
    static

Template:
    keys(Pairs,Keys)
Mode and number of proofs:
    keys(+list(pair),-list) - one

---

key/2

Enumerates by backtracking all keys from a list of pairs.

Compilation flags:
    static

Template:
    key(Pairs,Key)
Mode and number of proofs:
    key(+list(pair),-term) - zero_or_more

---

values/2

Returns a list of values from a list of pairs.

Compilation flags:
    static

Template:
    values(Pairs,Values)
Mode and number of proofs:
    values(+list(pair),-list) - one

---

value/3

Returns a value addressed by the given path (a key or a list of keys in the case of nested list of pairs). Fails if path does not exist.

Compilation flags:
    static

Template:
    value(Pairs,Path,Value)
Mode and number of proofs:
    value(+list(pair),+term,-term) - zero_or_one
    value(+list(pair),+list,-term) - zero_or_one

---

transpose/2

Transposes a list of pairs by swapping each pair key and value. The relative order of the list elements is kept.

Compilation flags:
    static

Template:
    transpose(Pairs,TransposedPairs)
Mode and number of proofs:
    transpose(+list(pair),-list(pair)) - one

---

group_sorted_by_key/2

Groups pairs by key by sorting them and then constructing new pairs by grouping all values for a given key in a list. Keys are compared using equality. Relative order of values per key is kept. Resulting list of pairs is sorted by key.

Compilation flags:
    static

Template:
    group_sorted_by_key(Pairs,Groups)
Mode and number of proofs:
    group_sorted_by_key(+list(pair),-list(pair)) - one

---

group_consecutive_by_key/2

Groups pairs by constructing new pairs by grouping all values for consecutive key in a list. Keys are compared using equality. The relative order of the values for the same key is kept.

Compilation flags:
    static

Template:
group_consecutive_by_key(Pairs,Groups)
Mode and number of proofs:
group_consecutive_by_key(+list(pair),-list(pair)) - one

---

group_by_key/2

Same as the group_sorted_by_key/2 predicate. Deprecated.

Compilation flags:
static

Template:
group_by_key(Pairs,Groups)
Mode and number of proofs:
group_by_key(+list(pair),-list(pair)) - one

---

map/3

Maps a list into pairs using a closure that applies to each list element to compute its key.

Compilation flags:
static

Template:
map(Closure,List,Pairs)
Meta-predicate template:
map(2,*,*)
Mode and number of proofs:
map(@callable,+list,-list(pair)) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

object

1.228.19 term

Term utility predicates.

Availability:
logtalk_load(types(loader))

Author: Paulo Moura
Version: 1:11:0
Date: 2022-05-13

Compilation flags:
static, context_switching_calls

Implements:
public termp
Aliases:
termp variables/2 as vars/2

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 check/1 depth/2 ground/1 new/1 numbervars/1 numbervars/3 occurs/2 singletons/2 subsumes/2 subterm/2 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.228.20 termp

Term utility predicates protocol.

Availability:
logtalk_load(types(loader))

Author: Paulo Moura
Version: 1:35:0
Date: 2022-05-13

Compilation flags:
static

Extends:
public comparingp

Remarks:

(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2

- Public predicates
  - depth/2
  - ground/1
  - new/1
  - occurs/2
  - subsumes/2
  - subterm/2
  - valid/1
  - check/1
  - variant/2
  - variables/2
  - singletons/2
  - numbervars/3
  - numbervars/1
  - varnumbers/3
  - varnumbers/2
- Protected predicates
- Private predicates
- Operators

Public predicates

depth/2

True if the depth of Term is Depth. The depth of atomic terms is zero; the depth of a compound term is one plus the maximum depth of its sub-terms.

Compilation flags:
static

Template:
depth(Term,Depth)

Mode and number of proofs:
depth(@term,?integer) - zero_or_one

---

ground/1

True if the argument is ground. Deprecated. Use the ground/1 standard predicate instead.

Compilation flags:
static

Template:
ground(Term)
Mode and number of proofs:
ground(@term) - zero_or_one

---

new/1

Creates a new term instance (if meaningful).

Compilation flags:
static

Template:
new(Term)
Mode and number of proofs:
new(-nonvar) - zero_or_one

---

occurs/2

True if the variable occurs in the term.

Compilation flags:
static

Template:
occurs(Variable,Term)
Mode and number of proofs:
occurs(@var,@term) - zero_or_one

---

subsumes/2

The first term subsumes the second term. Deprecated. Use the subsumes_term/2 standard predicate instead.

Compilation flags:
static

Template:
subsumes(General,Specific)
Mode and number of proofs:
subsumes(@term,@term) - zero_or_one

---

subterm/2

The first term is a subterm of the second term.

Compilation flags:
static

Template:
subterm(Subterm,Term)
Mode and number of proofs:
subterm(?term,+term) - zero_or_more

---

valid/1

Term is valid.

Compilation flags:
    static

Template:
    valid(Term)
Mode and number of proofs:
    valid(@nonvar) - zero_or_one

---

check/1

Checks if a term is valid. Throws an exception if the term is not valid.

Compilation flags:
    static

Template:
    check(Term)
Mode and number of proofs:
    check(@nonvar) - one

---

variant/2

Each term is a variant of the other (i.e., they are structurally equivalent).

Compilation flags:
    static

Template:
    variant(Term1,Term2)
Mode and number of proofs:
    variant(@term,@term) - zero_or_one

---

variables/2

Returns a list of all term variables (ordered as found when doing a depth-first, left-to-right traversal of Term). Deprecated. Use the standard term_variables/2 predicate instead.

Compilation flags:
    static

Template:
    variables(Term,List)
Mode and number of proofs:
    variables(@term,-list) - one

---

singletons/2

Returns a list of all term singleton variables (ordered as found when doing a depth-first, left-to-right traversal of Term).

Compilation flags:
    static

Template:
    singletons(Term,Singletons)
Mode and number of proofs:
    singletons(@term,-list) - one

---

numbervars/3

Grounds a term by replacing all variables with '$VAR'(N) terms with N starting at From. The Next argument is unified with the next value for N after binding all variables.

Compilation flags:
    static

Template:
    numbervars(Term,From,Next)
Mode and number of proofs:

numbervars(?term,+integer,?integer) - zero_or_one

---

numbervars/1

Grounds a term by replacing all variables with '$VAR'(N) terms with N starting at 0.

Compilation flags:
static

Template:
numbervars(Term)
Mode and number of proofs:
numbervars(?term) - zero_or_one

---

varnumbers/3

Replaces all '$VAR'(N) sub-terms in a term with fresh variables for all values of N grater or equal to From. Variables in Term are shared with Copy.

Compilation flags:
static

Template:
varnumbers(Term,From,Copy)
Mode and number of proofs:
varnumbers(@term,+integer,?term) - zero_or_one

---

varnumbers/2

Replaces all '$VAR'(N) sub-terms in a term with fresh variables for all values of N grater or equal to 0. Variables in Term are shared with Copy.

Compilation flags:
static

Template:
varnumbers(Term,Copy)
Mode and number of proofs:
varnumbers(@term,?term) - zero_or_one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

term

object

1.228.21 type

Type checking predicates. User extensible. New types can be defined by adding clauses for the type/1 and check/2 multifile predicates.

Availability:
logtalk_load(types(loader))

Author: Paulo Moura
Version: 2:8:0
Date: 2026-02-28

Compilation flags:
static, context_switching_calls, complements(restrict)

Uses:
list

Remarks:

- Logtalk specific types: entity, object, protocol, category, entity_identifier, object_identifier, protocol_identifier, category_identifier, event, predicate.
- Prolog module related types (when the backend compiler supports modules): module, module_identifier, qualified_callable.
- Prolog base types: term, var, nonvar, atomic, atom, number, integer, float, compound, callable, ground.
- Atom derived types: non_quoted_atom, non_empty_atom, boolean, character, in_character, char, operator_specifier, hex_char.
- Atom derived parametric types: atom(CharSet), atom(CharSet,Length), non_empty_atom(CharSet), character(CharSet), in_character(CharSet), char(CharSet).
- Number derived types: positive_number, negative_number, non_positive_number, non_negative_number.
- Float derived types: positive_float, negative_float, non_positive_float, non_negative_float, probability.
- Integer derived types: positive_integer, negative_integer, non_positive_integer, non_negative_integer, byte, in_byte, character_code, in_character_code, code, operator_priority, hex_code.
- Integer derived parametric types: character_code(CharSet), in_character_code(CharSet), code(CharSet).
- List types (compound derived types): list, non_empty_list, partial_list, list_or_partial_list, list(Type), list(Type,Length), list(Type,Min,Max), list(Type,Length,Min,Max), non_empty_list(Type), codes, chars.
- Difference list types (compound derived types): difference_list, difference_list(Type).
- Other compound derived types: compound(Name,Types), predicate_indicator, non_terminal_indicator, predicate_or_non_terminal_indicator, clause, grammar_rule, pair, pair(KeyType,ValueType), cyclic, acyclic.
- Stream types: stream, stream_or_alias, stream(Property), stream_or_alias(Property).
- Other types: text, text(CharSet), Object::Closure, between(Type,Lower,Upper), property(Type, LambdaExpression), one_of(Type,Set), var_or(Type), ground(Type), types(Types), constrain(Type, Closure), type.
- Type predicate notes: This type is used to check for an object public predicate specified as Object::Functor/Arity.
- Type boolean notes: The two value of this type are the atoms true and false.
- Stream types notes: In the case of the stream(Property) and stream_or_alias(Property) types, Property must be a valid stream property.
- Type order notes: The three possible values of this type are the single character atoms <, =, and >.
- Type character_code notes: This type takes into account Unicode support by the backend compiler. When Unicode is supported, it distinguishes between BMP and full support. When Unicode is not supported, it assumes a byte representation for characters.
- Types text and text(CharSet) notes: These types allow text to be represented using atoms, character lists, or character code lists.

- Type Object::Closure notes: Allows calling a public object predicate for type-checking. The predicate should provide valid/2 predicate semantics and assume called with a bound argument. The Closure closure is extended with a single argument, the value to be checked.
- Type compound(Name,Types) notes: This type verifies that a compound term have the given Name and its arguments conform to Types.
- Type between(Type, Lower, Upper) notes: The type argument allows distinguishing between numbers and other types. It also allows choosing between mixed integer/float comparisons and strict float or integer comparisons. The term is type-checked before testing for interval membership.
- Type property(Type, Lambda) notes: Verifies that Term satisfies a property described using a lambda expression of the form [Parameter]>>Goal. The lambda expression is applied in the context of user. The term is type-checked before calling the goal.
- Type one_of(Type, Set) notes: For checking if a given term is an element of a set. The set is represented using a list. The term is type-checked before testing for set membership.
- Type var_or(Type) notes: Allows checking if a term is either a variable or a valid value of the given type.
- Type ground(Type) notes: Allows checking if a term is ground and a valid value of the given type.
- Type types(Types) notes: Allows checking if a term is a valid value for one of the types in a list of types.
- Type constrain(Type,Closure) notes: Allows checking if a term is a valid value for the given type and satisfies the given closure.
- Type type notes: Allows checking if a term is a valid type.
- Type qualified_callable notes: Allows checking if a term is a possibly module-qualified callable term. When the term is qualified, it also checks that the qualification modules are type correct. When the term is not qualified, its semantics are the same as the callable type.
- Design choices: The main predicates are valid/2 and check/3. These are defined using the predicate check/2. Defining clauses for check/2 instead of valid/2 gives the user full control of exception terms without requiring an additional predicate.
- Error context: The built-in execution-context method context/1 can be used to provide the calling context for errors when using the predicate check/3.
- Registering new types: New types can be registered by defining clauses for the type/1 and check/2 multifile predicates. Clauses for both predicates must have a bound first argument to avoid introducing spurious choice-points when type-checking terms.
- Meta-types: Meta-types are types that have one or more sub-type arguments. E.g. var_or(Type). The sub-types of a meta-type can be enumerated by defining a clause for the meta_type/3 multifile predicate.
- Character sets: When testing character or character code based terms (e.g., atom), it is possible to choose a character set (ascii_identifier, ascii_printable, ascii_full, hexadecimal, byte, unicode_bmp, or unicode_full) using the parameterizable types.
- Caveats: The type argument (and any type parameterization) to the predicates is not type-checked (or checked for consistency) for performance reasons.
- Unicode limitations: Currently, correct character/code type-checking is only ensured for SWI-Prolog and XVM as other backends do not provide support for querying a Unicode code point category.
- Random seed: When setting the random generator seed using the set_seed/1 predicate inherited from the arbitrary category, the seed must be a valid value for the portable random generator provided by the random library.

Inherited public predicates:
arbitrary/1 arbitrary/2 edge_case/2 get_seed/1 max_size/1 mutation/3 set_seed/1 shrink/3 shrink_sequence/3 shrinker/1

- Public predicates
  - type/1
  - meta_type/3
  - valid/2
  - check/3
  - check/2
- Protected predicates
- Private predicates
- Operators

Public predicates

type/1

Table of defined types. A new type can be registered by defining a clause for this predicate and adding a clause for the check/2 multifile predicate.

Compilation flags:
static, multifile

Template:
type(Type)
Mode and number of proofs:
type(?callable) - zero_or_more

---

meta_type/3

Table of defined meta-types. A registered type that is a meta-type can be described by defining a clause for this predicate to enumerate its sub-types and optional values in case of a single sub-type.

Compilation flags:
static, multifile

Template:
meta_type(MetaType,SubTypes,Values)
Mode and number of proofs:
meta_type(?callable,-list,-list) - zero_or_more

---

valid/2

True if the given term is of the specified type. Fails otherwise.

Compilation flags:
static

Template:
valid(Type,Term)
Mode and number of proofs:
valid(@callable,@term) - zero_or_one

---

check/3

True if the given term is of the specified type. Throws an error otherwise using the format error(Error, Context). For the possible values of Error see the check/2 predicate.

Compilation flags:
static

Template:
check(Type,Term,Context)
Mode and number of proofs:
check(@callable,@term,@term) - one_or_error

---

check/2

True if the given term is of the specified type. Throws an error otherwise. A new type can be added by defining a clause for this predicate and registering it by adding a clause for the type/1 multifile predicate.

Compilation flags:
    static, multifile

Template:
    check(Type,Term)
Meta-predicate template:
    check(::,*)
Mode and number of proofs:
    check(@callable,@term) - one_or_error

Exceptions:
    Term is not bound as required:
        instantiation_error
    Term is bound but not of the specified type:
        type_error(Type,Term)
    Term is the of the correct type but not in the specified domain:
        domain_error(Domain,Term)
    Term is the of the correct type and domain but the resource it represents does not exist:
        existence_error(Type,Term)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

arbitrary, os_types, either, maybe

object

1.228.22 varlist

List of variables predicates.

Availability:
logtalk_load(types(loader))

Author: Paulo Moura
Version: 2:0:0
Date: 2020-05-11

Compilation flags:
static, context_switching_calls

Implements:
public varlistp

Remarks:
(none)

Inherited public predicates:
append/3 check/1 delete/3 empty/1 flatten/2 last/2 length/2 memberchk/2 nextto/3 nth0/3 nth0/4 nth1/3 nth1/4 permutation/2 prefix/2 remove_duplicates/2 reverse/2 same_length/2 select/3 sublist/2 subtract/3 suffix/2 valid/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

list, list(Type), numberlist, difflist

protocol

1.228.23 varlistp

List of variables protocol.

Availability:
logtalk_load(types(loader))

Author: Paulo Moura
Version: 1:3:0
Date: 2022-09-19

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - append/3
  - delete/3
  - empty/1
  - flatten/2
  - last/2
  - length/2
  - memberchk/2
  - nextto/3
  - nth0/3
  - nth0/4
  - nth1/3
  - nth1/4
  - permutation/2
  - prefix/2
  - remove_duplicates/2
  - reverse/2
  - same_length/2
  - select/3
  - sublist/2
  - subtract/3
  - suffix/2
  - valid/1
  - check/1
- Protected predicates
- Private predicates
- Operators

Public predicates

append/3

Appends two lists.

Compilation flags:
    static

Template:
append(List1,List2,List)
Mode and number of proofs:
append(?list,?list,?list) - zero_or_more

---

delete/3

Deletes from a list all occurrences of an element returning the list of remaining elements.

Compilation flags:
static

Template:
delete(List,Element,Remaining)
Mode and number of proofs:
delete(@list,@term,?list) - one

---

empty/1

True if the argument is an empty list.

Compilation flags:
static

Template:
empty(List)
Mode and number of proofs:
empty(@list) - zero_or_one

---

flatten/2

Flattens a list of lists into a list.

Compilation flags:
    static

Template:
    flatten(List,Flatted)
Mode and number of proofs:
    flatten(@list,-list) - one

---

last/2

List last element (if it exists).

Compilation flags:
    static

Template:
    last(List,Last)
Mode and number of proofs:
    last(@list,@var) - zero_or_one

---

length/2

List length.

Compilation flags:
    static

Template:
    length(List,Length)
Mode and number of proofs:
    length(@list,?integer) - zero_or_one

---

memberchk/2

Checks if a variable is a member of a list.

Compilation flags:
    static

Template:
    memberchk(Element,List)
Mode and number of proofs:
    memberchk(@var,@list) - zero_or_one

---

nextto/3

X and Y are consecutive elements in List.

Compilation flags:
    static

Template:
    nextto(X,Y,List)
Mode and number of proofs:
    nextto(@var,@var,?list) - zero_or_more

---

nth0/3

Nth element of a list (counting from zero).

Compilation flags:
    static

Template:
    nth0(Nth,List,Element)
Mode and number of proofs:
    nth0(?integer,+list,@var) - zero_or_more

---

nth0/4

Nth element of a list (counting from zero). Rest is a list of all the other elements. Can be used to either select the nth element of List or to insert an element before the nth element in Rest.

Compilation flags:
    static

Template:
    nth0(Nth,List,Element,Rest)
Mode and number of proofs:
    nth0(?integer,+list,@var,?list) - zero_or_more

---

nth1/3

Nth element of a list (counting from one).

Compilation flags:
    static

Template:
    nth1(Nth,List,Element)
Mode and number of proofs:
    nth1(?integer,+list,@var) - zero_or_more

---

nth1/4

Nth element of a list (counting from zero). Rest is a list of all the other elements. Can be used to either select the nth element of List or to insert an element before the nth element in Rest.

Compilation flags:
    static

Template:
    nth1(Nth,List,Element,Rest)
Mode and number of proofs:
    nth1(?integer,+list,@var,?list) - zero_or_more

---

permutation/2

The two lists are a permutation of the same list.

Compilation flags:
    static

Template:
    permutation(List,Permutation)
Mode and number of proofs:
    permutation(@list,@list) - zero_or_one

---

prefix/2

Prefix is a prefix of List.

Compilation flags:
    static

Template:
    prefix(Prefix,List)
Mode and number of proofs:
    prefix(?list,@list) - zero_or_more

---

remove_duplicates/2

Removes duplicated variables and keeping the left-most variable when repeated.

Compilation flags:
    static

Template:
    remove_duplicates(List,Set)
Mode and number of proofs:

remove_duplicates(+list,-list) - one

---

reverse/2

Reverses a list.

Compilation flags:
static

Template:
reverse(List,Reversed)
Mode and number of proofs:
reverse(@list,?list) - zero_or_one
reverse(?list,@list) - zero_or_one
reverse(-list,-list) - one_or_more

---

same_length/2

The two lists have the same length.

Compilation flags:
static

Template:
same_length(List1,List2)
Mode and number of proofs:
same_length(@list,?list) - zero_or_one
same_length(?list,@list) - zero_or_one
same_length(-list,-list) - one_or_more

---

select/3

Selects an element from a list, returning the list of remaining elements.

Compilation flags:
    static

Template:
    select(Element,List,Remaining)
Mode and number of proofs:
    select(@var,?list,?list) - zero_or_more

---

sublist/2

The first list is a sublist of the second.

Compilation flags:
    static

Template:
    sublist(Sublist,List)
Mode and number of proofs:
    sublist(?list,@list) - zero_or_more

---

subtract/3

Removes all elements in the second list from the first list, returning the list of remaining elements.

Compilation flags:
    static

Template:
    subtract(List,Elements,Remaining)
Mode and number of proofs:
    subtract(@list,@list,-list) - one

---

suffix/2

Suffix is a suffix of List.

Compilation flags:
    static

Template:
    suffix(Suffix,List)
Mode and number of proofs:
    suffix(?list,@list) - zero_or_more

---

valid/1

Term is a valid list of variables.

Compilation flags:
    static

Template:
    valid(Term)
Mode and number of proofs:
    valid(@nonvar) - zero_or_one

---

check/1

Checks if a term is a valid list of variables. Throws an exception if the term is not valid.

Compilation flags:
    static

Template:
    check(Term)
Mode and number of proofs:
    check(@nonvar) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

varlist, listp, numberlistp

1.229 tzif

object

1.229.1 tzif

Loader, per-zone cache and snapshot persistence support, and zone-aware UTC lookup predicates for TZif v1/v2/v3 files.

Availability:
logtalk_load(tzif(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-04-17

Compilation flags:
static, context_switching_calls

Implements:
public tzif_protocol
Uses:
date
list
logtalk
os
reader

tzif_zone_ids

Remarks:
(none)

Inherited public predicates:
abbreviation/2 abbreviation/3 abbreviation/4 cache/1 cache_source/1 cached_tzif/1 clear_cache/0 daylight_saving_time/2 daylight_saving_time/3 daylight_saving_time/4 load/1 load/2 local_abbreviation/2 local_abbreviation/3 local_abbreviation/4 local_abbreviation_reified/2 local_abbreviation_reified/3 local_abbreviation_reified/4 local_abbreviation_with_resolution/3 local_abbreviation_with_resolution/4 local_abbreviation_with_resolution/5 local_daylight_saving_time/2 local_daylight_saving_time/3 local_daylight_saving_time/4 local_daylight_saving_time_reified/2 local_daylight_saving_time_reified/3 local_daylight_saving_time_reified/4 local_daylight_saving_time_with_resolution/3 local_daylight_saving_time_with_resolution/4 local_daylight_saving_time_with_resolution/5 local_offset/2 local_offset/3 local_offset/4 local_offset_reified/2 local_offset_reified/3 local_offset_reified/4 local_offset_with_resolution/3 local_offset_with_resolution/4 local_offset_with_resolution/5 local_time_type/2 local_time_type/3 local_time_type/4 local_time_type_reified/2 local_time_type_reified/3 local_time_type_reified/4 local_time_type_with_resolution/3 local_time_type_with_resolution/4 local_time_type_with_resolution/5 offset/2 offset/3 offset/4 save/1 save/2 time_type/2 time_type/3 time_type/4 zone/3 zones/1 zones/2

- Public predicates
- Protected predicates
- Private predicates
  - cached_tzif_/2
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

cached_tzif_/2

Table holding the currently cached TZif terms keyed by zone identifier.

Compilation flags:
dynamic

Template:
cached_tzif_(Zone,TZif)
Mode and number of proofs:
cached_tzif_(?atom,?compound) - zero_or_more

Operators

(none)

protocol

1.229.2 tzif_protocol

Protocol for loading TZif data sets, persisting loaded terms, and answering zone-aware UTC-based offset, DST, and abbreviation queries.

Availability:
logtalk_load(tzif(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-07

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - load/1
  - load/2
  - cache/1
  - save/2
  - save/1
  - clear_cache/0
  - cache_source/1
  - cached_tzif/1
  - zone/3
  - zones/2
  - zones/1
  - time_type/4
  - time_type/3
  - time_type/2
  - offset/4
  - offset/3
  - offset/2
  - daylight_saving_time/4
  - daylight_saving_time/3
  - daylight_saving_time/2
  - abbreviation/4
  - abbreviation/3
  - abbreviation/2
  - local_time_type/4
  - local_time_type/3
  - local_time_type/2
  - local_time_type_with_resolution/5
  - local_time_type_with_resolution/4
  - local_time_type_with_resolution/3
  - local_offset/4
  - local_offset/3

  - local_offset/2
  - local_offset_with_resolution/5
  - local_offset_with_resolution/4
  - local_offset_with_resolution/3
  - local_daylight_saving_time/4
  - local_daylight_saving_time/3
  - local_daylight_saving_time/2
  - local_daylight_saving_time_with_resolution/5
  - local_daylight_saving_time_with_resolution/4
  - local_daylight_saving_time_with_resolution/3
  - local_abbreviation/4
  - local_abbreviation/3
  - local_abbreviation/2
  - local_abbreviation_with_resolution/5
  - local_abbreviation_with_resolution/4
  - local_abbreviation_with_resolution/3
  - local_time_type_reified/4
  - local_time_type_reified/3
  - local_time_type_reified/2
  - local_offset_reified/4
  - local_offset_reified/3
  - local_offset_reified/2
  - local_daylight_saving_time_reified/4
  - local_daylight_saving_time_reified/3
  - local_daylight_saving_time_reified/2
  - local_abbreviation_reified/4
  - local_abbreviation_reified/3
  - local_abbreviation_reified/2
- Protected predicates
- Private predicates
- Operators

Public predicates

load/1

Loads a TZif source given as file(Path, ZoneId), files(Root, Paths), directory(Root), stream(Stream, ZoneId), bytes(Bytes, ZoneId), or snapshot(File) and caches the resulting per-zone tzif(...) terms, replacing cached entries with matching zone identifiers. For directory(Root) sources, regular files whose relative paths are not recognized zone identifiers are ignored, allowing system zoneinfo trees that contain metadata files such as leapseconds. Zone identifiers are validated against bundled IANA TZDB 2026a canonical names plus backward-compatible aliases.

Compilation flags:
    static

Template:
    load(Source)
Mode and number of proofs:
    load(+compound) - zero_or_one

---

load/2

Loads a TZif source given as file(Path, ZoneId), files(Root, Paths), directory(Root), stream(Stream, ZoneId), bytes(Bytes, ZoneId), or snapshot(File) into a list of per-zone tzif(...) compound terms without caching them. For directory(Root) sources, regular files whose relative paths are not recognized zone identifiers are ignored, allowing system zoneinfo trees that contain metadata files such as leapseconds. Zone identifiers are validated against bundled IANA TZDB 2026a canonical names plus backward-compatible aliases.

Compilation flags:
    static

Template:
    load(Source,TZifs)
Mode and number of proofs:
    load(+compound,-list(compound)) - zero_or_one

---

cache/1

Caches one or more loaded per-zone tzif(...) compound terms, replacing cached entries with matching zone identifiers.

Compilation flags:
    static

Template:
    cache(TZifs)
Mode and number of proofs:
    cache(+list(compound)) - one

---

save/2

Saves a list of per-zone tzif(...) terms to a plain Prolog snapshot file, writing one serialized term per line.

Compilation flags:
    static

Template:
    save(TZifs,File)
Mode and number of proofs:
    save(+list(compound),+atom) - one

---

save/1

Saves all cached tzif(...) terms to a plain Prolog snapshot file.

Compilation flags:
    static

Template:
    save(File)
Mode and number of proofs:
    save(+atom) - one_or_error

---

clear_cache/0

Clears all cached TZif terms.

Compilation flags:
    static

Mode and number of proofs:
    clear_cache - one

---

cache_source/1

Enumerates the source terms recorded in the cached TZif terms.

Compilation flags:
    static

Template:
    cache_source(Source)
Mode and number of proofs:
    cache_source(-compound) - zero_or_more

---

cached_tzif/1

Enumerates the cached per-zone tzif(...) terms.

Compilation flags:
    static

Template:
    cached_tzif(TZif)
Mode and number of proofs:
    cached_tzif(-compound) - zero_or_more

---

zone/3

Returns the loaded zone identifier and its nested parsed zone-data term from a per-zone tzif(...) term.

Compilation flags:
    static

Template:
    zone(TZif,Zone,ZoneData)
Mode and number of proofs:
    zone(+compound,?atom,-compound) - zero_or_one

---

zones/2

Returns the list of zone identifiers loaded in a list of per-zone tzif(...) terms.

Compilation flags:
    static

Template:
    zones(TZifs,Zones)
Mode and number of proofs:
    zones(+list(compound),-list(atom)) - one

---

zones/1

Returns the list of zone identifiers loaded in the cached tzif(...) terms.

Compilation flags:
    static

Template:
    zones(Zones)
Mode and number of proofs:
    zones(-list(atom)) - one_or_error

---

time_type/4

Returns the applicable local time type for a loaded zone and a UTC instant given either as Unix seconds or as a date_time/6 term.

Compilation flags:
    static

Template:
    time_type(TZif,Zone,UTC,TimeType)
Mode and number of proofs:
    time_type(+compound,+atom,+types([integer,compound]),-compound) - zero_or_one

---

time_type/3

Returns the applicable local time type for a zone in the cached TZif terms.

Compilation flags:
    static

Template:
    time_type(Zone,UTC,TimeType)
Mode and number of proofs:
    time_type(+atom,+types([integer,compound]),-compound) - one_or_error

---

time_type/2

Cached single-zone convenience variant of time_type/3 using the cached TZif terms; requires exactly one cached zone.

Compilation flags:
    static

Template:
    time_type(UTC,TimeType)
Mode and number of proofs:
    time_type(+types([integer,compound]),-compound) - one_or_error

---

offset/4

Returns the UTC offset in seconds for a loaded zone and a UTC instant.

Compilation flags:
    static

Template:
    offset(TZif,Zone,UTC,OffsetSeconds)
Mode and number of proofs:
    offset(+compound,+atom,+types([integer,compound]),-integer) - zero_or_one

---

offset/3

Returns the UTC offset in seconds for a zone in the cached TZif terms.

Compilation flags:
    static

Template:
    offset(Zone,UTC,OffsetSeconds)
Mode and number of proofs:
    offset(+atom,+types([integer,compound]),-integer) - one_or_error

---

offset/2

Cached single-zone convenience variant of offset/3 using the cached TZif terms; requires exactly one cached zone.

Compilation flags:
    static

Template:
    offset(UTC,OffsetSeconds)

Mode and number of proofs:
offset(+types([integer,compound]),-integer) - one_or_error

---

daylight_saving_time/4

Returns true or false according to whether daylight saving time is active for a loaded zone and a UTC instant.

Compilation flags:
static

Template:
daylight_saving_time(TZif,Zone,UTC,IsDST)
Mode and number of proofs:
daylight_saving_time(+compound,+atom,+types([integer,compound]),-atom) - zero_or_one

---

daylight_saving_time/3

Returns daylight-saving information for a zone in the cached TZif terms.

Compilation flags:
static

Template:
daylight_saving_time(Zone,UTC,IsDST)
Mode and number of proofs:
daylight_saving_time(+atom,+types([integer,compound]),-atom) - one_or_error

---

daylight_saving_time/2

Cached single-zone convenience variant of daylight_saving_time/3 using the cached TZif terms; requires exactly one cached zone.

Compilation flags:
static

Template:
daylight_saving_time(UTC,IsDST)
Mode and number of proofs:
daylight_saving_time(+types([integer,compound]),-atom) - one_or_error

---

abbreviation/4

Returns the time-zone abbreviation for a loaded zone and a UTC instant.

Compilation flags:
static

Template:
abbreviation(TZif,Zone,UTC,Abbreviation)
Mode and number of proofs:
abbreviation(+compound,+atom,+types([integer,compound]),-atom) - zero_or_one

---

abbreviation/3

Returns the time-zone abbreviation for a zone in the cached TZif terms.

Compilation flags:
static

Template:
abbreviation(Zone,UTC,Abbreviation)
Mode and number of proofs:
abbreviation(+atom,+types([integer,compound]),-atom) - one_or_error

---

abbreviation/2

Cached single-zone convenience variant of abbreviation/3 using the cached TZif terms; requires exactly one cached zone.

Compilation flags:
    static

Template:
    abbreviation(UTC,Abbreviation)
Mode and number of proofs:
    abbreviation(+types([integer,compound]),-atom) - one_or_error

---

local_time_type/4

Returns the applicable local time type for a loaded zone and a local civil time given as a date_time/6 term. This strict variant fails unless the local civil time has a unique interpretation.

Compilation flags:
    static

Template:
    local_time_type(TZif,Zone,LocalDateTime,TimeType)
Mode and number of proofs:
    local_time_type(+compound,+atom,+compound,-compound) - zero_or_one

---

local_time_type/3

Returns the applicable local time type for a zone in the cached TZif terms. This strict variant fails unless the local civil time has a unique interpretation.

Compilation flags:
    static

Template:
    local_time_type(Zone,LocalDateTime,TimeType)
Mode and number of proofs:

local_time_type(+atom,+compound,-compound) - one_or_error

---

local_time_type/2

Cached single-zone convenience variant of strict local civil-time lookup; requires exactly one cached zone and fails unless the local civil time has a unique interpretation.

Compilation flags:
static

Template:
local_time_type(LocalDateTime,TimeType)
Mode and number of proofs:
local_time_type(+compound,-compound) - one_or_error

---

local_time_type_with_resolution/5

Returns the applicable local time type for a loaded zone and a local civil time using the explicit resolution mode: strict (fail unless exactly one interpretation exists), first (prefer the earliest valid interpretation), second (prefer the latest valid interpretation), and all (enumerate all valid interpretations in chronological order).

Compilation flags:
static

Template:
local_time_type_with_resolution(TZif,Zone,LocalDateTime,ResolutionMode,TimeType)
Mode and number of proofs:
local_time_type_with_resolution(+compound,+atom,+compound,+atom,-compound) - zero_or_more

---

local_time_type_with_resolution/4

Returns the applicable local time type for a zone in the cached TZif terms using the explicit resolution mode: strict (fail unless exactly one interpretation exists), first (prefer the earliest valid interpretation), second (prefer the latest valid interpretation), and all (enumerate all valid interpretations in chronological order).

Compilation flags:
    static

Template:
    local_time_type_with_resolution(Zone,LocalDateTime,ResolutionMode,TimeType)
Mode and number of proofs:
    local_time_type_with_resolution(+atom,+compound,+atom,-compound) - zero_or_more

---

local_time_type_with_resolution/3

Cached single-zone convenience variant of local civil-time lookup using the explicit resolution mode: strict (fail unless exactly one interpretation exists), first (prefer the earliest valid interpretation), second (prefer the latest valid interpretation), and all (enumerate all valid interpretations in chronological order). Requires exactly one cached zone.

Compilation flags:
    static

Template:
    local_time_type_with_resolution(LocalDateTime,ResolutionMode,TimeType)
Mode and number of proofs:
    local_time_type_with_resolution(+compound,+atom,-compound) - zero_or_more

---

local_offset/4

Returns the UTC offset in seconds for a loaded zone and a local civil time. This strict variant fails unless the local civil time has a unique interpretation.

Compilation flags:
    static

Template:
local_offset(TZif,Zone,LocalDateTime,OffsetSeconds)
Mode and number of proofs:
local_offset(+compound,+atom,+compound,-integer) - zero_or_one

---

local_offset/3

Returns the UTC offset in seconds for a zone in the cached TZif terms. This strict variant fails unless the local civil time has a unique interpretation.

Compilation flags:
static

Template:
local_offset(Zone,LocalDateTime,OffsetSeconds)
Mode and number of proofs:
local_offset(+atom,+compound,-integer) - one_or_error

---

local_offset/2

Cached single-zone convenience variant of strict local civil-time offset lookup.

Compilation flags:
static

Template:
local_offset(LocalDateTime,OffsetSeconds)
Mode and number of proofs:
local_offset(+compound,-integer) - one_or_error

---

local_offset_with_resolution/5

Returns the UTC offset in seconds for a loaded zone and a local civil time using the explicit resolution mode: strict (fail unless exactly one interpretation exists), first (prefer the earliest valid interpretation), second (prefer the latest valid interpretation), and all (enumerate all valid interpretations in chronological order).

Compilation flags:
    static

Template:
    local_offset_with_resolution(TZif,Zone,LocalDateTime,ResolutionMode,OffsetSeconds)
Mode and number of proofs:
    local_offset_with_resolution(+compound,+atom,+compound,+atom,-integer) - zero_or_more

---

local_offset_with_resolution/4

Returns the UTC offset in seconds for a zone in the cached TZif terms using the explicit resolution mode: strict (fail unless exactly one interpretation exists), first (prefer the earliest valid interpretation), second (prefer the latest valid interpretation), and all (enumerate all valid interpretations in chronological order).

Compilation flags:
    static

Template:
    local_offset_with_resolution(Zone,LocalDateTime,ResolutionMode,OffsetSeconds)
Mode and number of proofs:
    local_offset_with_resolution(+atom,+compound,+atom,-integer) - zero_or_more

---

local_offset_with_resolution/3

Cached single-zone convenience variant of local civil-time offset lookup using the explicit resolution mode; requires exactly one cached zone.

Compilation flags:
    static

Template:

local_offset_with_resolution(LocalDateTime,ResolutionMode,OffsetSeconds)
Mode and number of proofs:
local_offset_with_resolution(+compound,+atom,-integer) - zero_or_more

---

local_daylight_saving_time/4

Returns daylight-saving information for a loaded zone and a local civil time. This strict variant fails unless the local civil time has a unique interpretation.

Compilation flags:
static

Template:
local_daylight_saving_time(TZif,Zone,LocalDateTime,IsDST)
Mode and number of proofs:
local_daylight_saving_time(+compound,+atom,+compound,-atom) - zero_or_one

---

local_daylight_saving_time/3

Returns daylight-saving information for a zone in the cached TZif terms. This strict variant fails unless the local civil time has a unique interpretation.

Compilation flags:
static

Template:
local_daylight_saving_time(Zone,LocalDateTime,IsDST)
Mode and number of proofs:
local_daylight_saving_time(+atom,+compound,-atom) - one_or_error

---

local_daylight_saving_time/2

Cached single-zone convenience variant of strict local daylight-saving lookup.

Compilation flags:
static

Template:
local_daylight_saving_time(LocalDateTime,IsDST)
Mode and number of proofs:
local_daylight_saving_time(+compound,-atom) - one_or_error

---

local_daylight_saving_time_with_resolution/5

Returns daylight-saving information for a loaded zone and a local civil time using the explicit resolution mode: strict (fail unless exactly one interpretation exists), first (prefer the earliest valid interpretation), second (prefer the latest valid interpretation), and all (enumerate all valid interpretations in chronological order).

Compilation flags:
static

Template:
local_daylight_saving_time_with_resolution(TZif,Zone,LocalDateTime,ResolutionMode,IsDST)
Mode and number of proofs:
local_daylight_saving_time_with_resolution(+compound,+atom,+compound,+atom,-atom) - zero_or_more

---

local_daylight_saving_time_with_resolution/4

Returns daylight-saving information for a zone in the cached TZif terms using the explicit resolution mode: strict (fail unless exactly one interpretation exists), first (prefer the earliest valid interpretation), second (prefer the latest valid interpretation), and all (enumerate all valid interpretations in chronological order).

Compilation flags:
static

Template:

local_daylight_saving_time_with_resolution(Zone,LocalDateTime,ResolutionMode,IsDST)
Mode and number of proofs:
local_daylight_saving_time_with_resolution(+atom,+compound,+atom,-atom) - zero_or_more

---

local_daylight_saving_time_with_resolution/3

Cached single-zone convenience variant of local civil-time daylight-saving lookup using the explicit resolution mode; requires exactly one cached zone.

Compilation flags:
static

Template:
local_daylight_saving_time_with_resolution(LocalDateTime,ResolutionMode,IsDST)
Mode and number of proofs:
local_daylight_saving_time_with_resolution(+compound,+atom,-atom) - zero_or_more

---

local_abbreviation/4

Returns the time-zone abbreviation for a loaded zone and a local civil time. This strict variant fails unless the local civil time has a unique interpretation.

Compilation flags:
static

Template:
local_abbreviation(TZif,Zone,LocalDateTime,Abbreviation)
Mode and number of proofs:
local_abbreviation(+compound,+atom,+compound,-atom) - zero_or_one

---

local_abbreviation/3

Returns the time-zone abbreviation for a zone in the cached TZif terms. This strict variant fails unless the local civil time has a unique interpretation.

Compilation flags:
    static

Template:
    local_abbreviation(Zone,LocalDateTime,Abbreviation)
Mode and number of proofs:
    local_abbreviation(+atom,+compound,-atom) - one_or_error

---

local_abbreviation/2

Cached single-zone convenience variant of strict local abbreviation lookup.

Compilation flags:
    static

Template:
    local_abbreviation(LocalDateTime,Abbreviation)
Mode and number of proofs:
    local_abbreviation(+compound,-atom) - one_or_error

---

local_abbreviation_with_resolution/5

Returns the time-zone abbreviation for a loaded zone and a local civil time using the explicit resolution mode: strict (fail unless exactly one interpretation exists), first (prefer the earliest valid interpretation), second (prefer the latest valid interpretation), and all (enumerate all valid interpretations in chronological order).

Compilation flags:
    static

Template:
    local_abbreviation_with_resolution(TZif,Zone,LocalDateTime,ResolutionMode,Abbreviation)

Mode and number of proofs:
local_abbreviation_with_resolution(+compound,+atom,+compound,+atom,-atom) - zero_or_more

---

local_abbreviation_with_resolution/4

Returns the time-zone abbreviation for a zone in the cached TZif terms using the explicit resolution mode: strict (fail unless exactly one interpretation exists), first (prefer the earliest valid interpretation), second (prefer the latest valid interpretation), and all (enumerate all valid interpretations in chronological order).

Compilation flags:
static

Template:
local_abbreviation_with_resolution(Zone,LocalDateTime,ResolutionMode,Abbreviation)
Mode and number of proofs:
local_abbreviation_with_resolution(+atom,+compound,+atom,-atom) - zero_or_more

---

local_abbreviation_with_resolution/3

Cached single-zone convenience variant of local civil-time abbreviation lookup using the explicit resolution mode; requires exactly one cached zone.

Compilation flags:
static

Template:
local_abbreviation_with_resolution(LocalDateTime,ResolutionMode,Abbreviation)
Mode and number of proofs:
local_abbreviation_with_resolution(+compound,+atom,-atom) - zero_or_more

---

local_time_type_reified/4

Returns a reified local civil-time lookup result for a loaded zone as one of unique(TimeType), ambiguous(TimeTypes), or nonexistent.

Compilation flags:
    static

Template:
    local_time_type_reified(TZif,Zone,LocalDateTime,Result)
Mode and number of proofs:
    local_time_type_reified(+compound,+atom,+compound,-compound) - one

---

local_time_type_reified/3

Returns a reified local civil-time lookup result for a zone in the cached TZif terms as one of unique(TimeType), ambiguous(TimeTypes), or nonexistent.

Compilation flags:
    static

Template:
    local_time_type_reified(Zone,LocalDateTime,Result)
Mode and number of proofs:
    local_time_type_reified(+atom,+compound,-compound) - one_or_error

---

local_time_type_reified/2

Cached single-zone convenience variant of reified local civil-time lookup; returns unique(TimeType), ambiguous(TimeTypes), or nonexistent.

Compilation flags:
    static

Template:
    local_time_type_reified(LocalDateTime,Result)
Mode and number of proofs:

local_time_type_reified(+compound,-compound) - one_or_error

---

local_offset_reified/4

Returns a reified local offset lookup result for a loaded zone as one of unique(OffsetSeconds), ambiguous(OffsetSecondsList), or nonexistent.

Compilation flags:
static

Template:
local_offset_reified(TZif,Zone,LocalDateTime,Result)
Mode and number of proofs:
local_offset_reified(+compound,+atom,+compound,-compound) - one

---

local_offset_reified/3

Returns a reified local offset lookup result for a zone in the cached TZif terms as one of unique(OffsetSeconds), ambiguous(OffsetSecondsList), or nonexistent.

Compilation flags:
static

Template:
local_offset_reified(Zone,LocalDateTime,Result)
Mode and number of proofs:
local_offset_reified(+atom,+compound,-compound) - one_or_error

---

local_offset_reified/2

Cached single-zone convenience variant of reified local offset lookup; returns unique(OffsetSeconds), ambiguous(OffsetSecondsList), or nonexistent.

Compilation flags:
    static

Template:
    local_offset_reified(LocalDateTime,Result)
Mode and number of proofs:
    local_offset_reified(+compound,-compound) - one_or_error

---

local_daylight_saving_time_reified/4

Returns a reified local daylight-saving lookup result for a loaded zone as one of unique(IsDST), ambiguous(IsDSTList), or nonexistent.

Compilation flags:
    static

Template:
    local_daylight_saving_time_reified(TZif,Zone,LocalDateTime,Result)
Mode and number of proofs:
    local_daylight_saving_time_reified(+compound,+atom,+compound,-compound) - one

---

local_daylight_saving_time_reified/3

Returns a reified local daylight-saving lookup result for a zone in the cached TZif terms as one of unique(IsDST), ambiguous(IsDSTList), or nonexistent.

Compilation flags:
    static

Template:
    local_daylight_saving_time_reified(Zone,LocalDateTime,Result)
Mode and number of proofs:

local_daylight_saving_time_reified(+atom,+compound,-compound) - one_or_error

---

local_daylight_saving_time_reified/2

Cached single-zone convenience variant of reified local daylight-saving lookup; returns unique(IsDST), ambiguous(IsDSTList), or nonexistent.

Compilation flags:
static

Template:
local_daylight_saving_time_reified(LocalDateTime,Result)
Mode and number of proofs:
local_daylight_saving_time_reified(+compound,-compound) - one_or_error

---

local_abbreviation_reified/4

Returns a reified local abbreviation lookup result for a loaded zone as one of unique(Abbreviation), ambiguous(Abbreviations), or nonexistent.

Compilation flags:
static

Template:
local_abbreviation_reified(TZif,Zone,LocalDateTime,Result)
Mode and number of proofs:
local_abbreviation_reified(+compound,+atom,+compound,-compound) - one

---

local_abbreviation_reified/3

Returns a reified local abbreviation lookup result for a zone in the cached TZif terms as one of unique(Abbreviation), ambiguous(Abbreviations), or nonexistent.

Compilation flags:
    static

Template:
    local_abbreviation_reified(Zone,LocalDateTime,Result)
Mode and number of proofs:
    local_abbreviation_reified(+atom,+compound,-compound) - one_or_error

---

local_abbreviation_reified/2

Cached single-zone convenience variant of reified local abbreviation lookup; returns unique(Abbreviation), ambiguous(Abbreviations), or nonexistent.

Compilation flags:
    static

Template:
    local_abbreviation_reified(LocalDateTime,Result)
Mode and number of proofs:
    local_abbreviation_reified(+compound,-compound) - one_or_error

---

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

object

1.229.3 tzif_zone_ids

Bundled canonical TZDB zone ids and backward-compatible aliases derived from IANA TZDB 2026a.

Availability:
logtalk_load(tzif(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-04-07

Compilation flags:
static, context_switching_calls

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - tzdb_version/1
  - known_zone_id/1
  - zone_id_kind/2
- Protected predicates
- Private predicates
- Operators

Public predicates

tzdb_version/1

Bundled IANA TZDB release used to derive the zone-id table.

Compilation flags:
    static

Template:
    tzdb_version(Version)
Mode and number of proofs:
    tzdb_version(-atom) - one

---

known_zone_id/1

True when the argument is a bundled canonical TZDB zone id or backward-compatible alias.

Compilation flags:
    static

Template:
    known_zone_id(ZoneId)
Mode and number of proofs:
    known_zone_id(+atom) - zero_or_one

---

zone_id_kind/2

Classifies a bundled zone id as canonical or as a backward-compatible alias targeting another zone id.

Compilation flags:
    static

Template:
    zone_id_kind(ZoneId,Kind)
Mode and number of proofs:
    zone_id_kind(+atom,-compound) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.230 ulid

object

1.230.1 ulid

Universally Unique Lexicographically Sortable Identifier (ULID) generator using an atom representation.

Availability:
logtalk_load(ulid(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-02-26

Compilation flags:
static, context_switching_calls

Extends:
public ulid(atom)

Remarks:
(none)

Inherited public predicates:
generate/1 generate/2 generate/8 timestamp/2 timestamp/8

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

ulid(Representation), ulid_types, cuid2, ids, ksuid, nanoid, snowflakeid, uuid

object

1.230.2 ulid(Representation)

- Representation - Text representation for the ULID. Possible values are atom, chars, and codes.

Universally Unique Lexicographically Sortable Identifier (ULID) generator.

Availability:
logtalk_load(ulid(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-02-26

Compilation flags:

static, context_switching_calls

Implements:
public ulid_protocol
Uses:
fast_random
iso8601
list
os

Remarks:
(none)

Inherited public predicates:
generate/1 generate/2 generate/8 timestamp/2 timestamp/8

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

ulid, ulid_types, cuid2(Representation,Size,Alphabet), ids(Representation,Bytes), ksuid(Representation,Alphabet), nanoid(Representation,Size,Alphabet), snowflakeid(Representation,EpochMilliseconds,Tir uuid(Representation)

protocol

1.230.3 ulid_protocol

Universally Unique Lexicographically Sortable Identifier (ULID) generator protocol.

Availability:
logtalk_load(ulid(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2023-05-17

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - generate/1
  - generate/2
  - generate/8

  - timestamp/2
  - timestamp/8
- Protected predicates
- Private predicates
- Operators

Public predicates

generate/1

Generates a new ULID.

Compilation flags:
    static

Template:
    generate(ULID)
Mode and number of proofs:
    generate(--ulid) - one

---

generate/2

Generates a new ULID from a timestamp (number of milliseconds since the Unix epoch: 00:00:00 UTC on January 1, 1970).

Compilation flags:
    static

Template:
    generate(Milliseconds,ULID)
Mode and number of proofs:
    generate(+integer,--ulid) - one

---

generate/8

Generates a new ULID from a timestamp discrete components.

Compilation flags:
    static

Template:
    generate(Year,Month,Day,Hours,Minutes,Seconds,Milliseconds,ULID)
Mode and number of proofs:
    generate(+integer,+integer,+integer,+integer,+integer,+integer,+integer,--ulid) - one

---

timestamp/2

Returns the given ULID timestamp (number of milliseconds since the Unix epoch: 00:00:00 UTC on January 1, 1970).

Compilation flags:
    static

Template:
    timestamp(ULID,Milliseconds)
Mode and number of proofs:
    timestamp(++ulid,-integer) - one

---

timestamp/8

Decodes a ULID into its timestamp discrete components.

Compilation flags:
    static

Template:
    timestamp(ULID,Year,Month,Day,Hours,Minutes,Seconds,Milliseconds)
Mode and number of proofs:
    timestamp(++ulid,-integer,-integer,-integer,-integer,-integer,-integer,-integer) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

category

1.230.4 ulid_types

ULID type definition.

Availability:
logtalk_load(ulid(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2023-05-19

Compilation flags:
static

Provides:
type::type/1
type::check/2
Uses:
list
type

Remarks:

- Provided types: This category adds a ulid(Representation) type for type-checking when using the ulid library object. Valid representation values are atom, chars, and codes.

Inherited public predicates:

(none)

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

ulid(Representation), ulid

1.231 union_find

object

1.231.1 union_find

Union find data structure implementation.

Availability:
logtalk_load(union_find(loader))

Author: José Antonio Riaza Valverde; adapted to Logtalk by Paulo Moura
Version: 1:0:0

Date: 2022-02-18

Compilation flags:
static, context_switching_calls

Implements:
public union_find_protocol
Extends:
public compound
Uses:
avltree

Remarks:
(none)

Inherited public predicates:
(<)/2 (=:=)/2 (=<)/2 (=\=)/2 (>)/2 (>=)/2 check/1 depth/2 disjoint_sets/2 find/4 find/5 ground/1 make_set/3 new/1 new/2 numbervars/1 numbervars/3 occurs/2 singletons/2 subsumes/2 subterm/2 union/4 union_all/3 valid/1 variables/2 variant/2 varnumbers/2 varnumbers/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.231.2 union_find_protocol

Union-find data structure protocol.

Availability:
logtalk_load(union_find(loader))

Author: José Antonio Riaza Valverde; adapted to Logtalk by Paulo Moura
Version: 1:0:0
Date: 2022-02-17

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - new/2
  - make_set/3
  - union/4
  - union_all/3
  - find/4

    - find/5
    - disjoint_sets/2
- Protected predicates
- Private predicates
- Operators

Public predicates

new/2

Creates a new union-find data structure with a list of elements as keys.

Compilation flags:
    static

Template:
    new(Elements,UnionFind)
Mode and number of proofs:
    new(+list(element),?union_find) - zero_or_one

---

make_set/3

Makes a new set by creating a new element with a unique key Element, a rank of 0, and a parent pointer to itself. The parent pointer to itself indicates that the element is the representative member of its own set.

Compilation flags:
    static

Template:
    make_set(UnionFind,Element,NewUnionFind)
Mode and number of proofs:
    make_set(+union_find,+element,?union_find) - zero_or_one

---

union/4

Merges the two trees, if distinct, that contain the given elements. The trees are joined by attaching the shorter tree (by rank) to the root of the taller tree. Fails if any of the elements is not found.

Compilation flags:
    static

Template:
    union(UnionFind,Element1,Element2,NewUnionFind)
Mode and number of proofs:
    union(+union_find,+element,+element,?union_find) - zero_or_one

---

union_all/3

Merges the distinct trees for all the given elements returning the resulting union-find data structure. Fails if any of the elements is not found.

Compilation flags:
    static

Template:
    union_all(UnionFind,Elements,NewUnionFind)
Mode and number of proofs:
    union_all(+union_find,+list(element),?union_find) - zero_or_one

---

find/4

Finds the root element of a set by following the chain of parent pointers from the given element. Root is the representative member of the set to which the element belongs, and may be element itself. Fails if the element is not found.

Compilation flags:
    static

Template:
    find(UnionFind,Element,Root,NewUnionFind)

Mode and number of proofs:
find(+union_find,+element,?element,?union_find) - zero_or_one

Remarks:

- Path compression: The structure of the tree containing the element is flattened by making every node point to the root whenever this predicate is used on it.

---

find/5

Same as the find/4 predicate, but returning also the rank of the root. Fails if the element is not found.

Compilation flags:
static

Template:
find(UnionFind,Element,Root,Rank,UnionFindOut)
Mode and number of proofs:
find(+union_find,+element,?element,?rank,?union_find) - zero_or_one

Remarks:

- Path compression: The structure of the tree containing the element is flattened by making every node point to the root whenever this predicate is used on it.

---

disjoint_sets/2

Returns the list of disjoint sets in the given union-find data structure.

Compilation flags:
static

Template:
disjoint_sets(UnionFind,Sets)
Mode and number of proofs:
disjoint_sets(+union_find,?sets) - zero_or_one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

union_find

1.232 url

object

1.232.1 url(Representation)

- Representation - URL and is components representation. Valid values are atom, codes, and chars.

URL validating, parsing, and normalizing predicates following RFC3986 nomenclature.

Availability:
logtalk_load(url(loader))

Author: Paulo Moura
Version: 1:2:0
Date: 2026-04-17

Compilation flags:
static, context_switching_calls

Uses:
list

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - valid/1
  - parse/2
  - generate/2
  - normalize/2
  - file_path_components/2
- Protected predicates
- Private predicates
  - downcase_text/2
- Operators

Public predicates

valid/1

True iff the argument is a valid URL, including optional query and fragment components.

Compilation flags:
static

Template:
valid(URL)
Mode and number of proofs:
valid(++text) - zero_or_one

---

parse/2

Parses a URL into a list of its components: [scheme(Scheme), authority(Authority), path(Path), query(Query), fragment(Fragment)]. Fails if the URL is invalid and cannot be parsed.

Compilation flags:
static

Template:
parse(URL,Components)
Mode and number of proofs:
parse(++text,-list(compound)) - zero_or_one

---

generate/2

Generates a normalized URL from a list of its components: [scheme(Scheme), authority(Authority), path(Path), query(Query), fragment(Fragment)] for standard URLs, or scheme-specific components for mailto, news, tel, and urn URLs. Fails if the components are invalid.

Compilation flags:
static

Template:
generate(Components,URL)
Mode and number of proofs:
generate(++list(compound),-text) - zero_or_one

---

normalize/2

Normalizes a URL by standardizing its components. Normalization includes converting the scheme to lowercase, percent-encoding characters that require escaping, ensuring proper path separators, and handling relative paths.

Compilation flags:
static

Template:
normalize(URL,NormalizedURL)
Mode and number of proofs:
normalize(++text,-text) - one

---

file_path_components/2

Converts a file-system path into file URL components represented as [authority(Authority), path(Path)]. Windows drive-letter and UNC paths are normalized to RFC3986-compatible file URL components.

Compilation flags:
    static

Template:
    file_path_components(FilePath,Components)
Mode and number of proofs:
    file_path_components(++text,-list(compound)) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

downcase_text/2

Converts text to lowercase (ASCII only). Only uppercase letters A-Z are converted to lowercase.

Compilation flags:
    static

Template:
    downcase_text(Text,LowerText)
Mode and number of proofs:
    downcase_text(+text,-text) - one

Operators

(none)

1.233 uuid

object

1.233.1 uuid

Universally unique identifier (UUID) generator using an atom representation.

Availability:
logtalk_load(uuid(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-26

Compilation flags:
static, context_switching_calls

Extends:
public uuid(atom)

Remarks:
(none)

Inherited public predicates:
random_node/1 uuid_max/1 uuid_nil/1 uuid_null/1 uuid_v1/2 uuid_v3/3 uuid_v4/1 uuid_v5/3 uuid_v7/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

uuid(Representation), cuid2, ksuid, ids, nanoid, snowflakeid, ulid

object

1.233.2 uuid(Representation)

- Representation - Text representation for the UUID. Possible values are atom, chars, and codes.

Universally unique identifier (UUID) generator.

Availability:
logtalk_load(uuid(loader))

Author: Paulo Moura
Version: 0:9:0
Date: 2026-04-08

Compilation flags:
static, context_switching_calls

Implements:
public uuid_protocol
Uses:
fast_random
iso8601
list
os

Remarks:
(none)

Inherited public predicates:
random_node/1 uuid_max/1 uuid_nil/1 uuid_null/1 uuid_v1/2 uuid_v3/3 uuid_v4/1 uuid_v5/3 uuid_v7/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

uuid, cuid2(Representation,Size,Alphabet), ksuid(Representation,Alphabet), ids(Representation,Bytes), nanoid(Representation,Size,Alphabet), snowflakeid(Representation,EpochMilliseconds,TimeUnitMilliseconds,TimestampBi ulid(Representation)

protocol

1.233.3 uuid_protocol

Universally unique identifier (UUID) generator protocol.

Availability:
logtalk_load(uuid(loader))

Author: Paulo Moura
Version: 0:6:0
Date: 2026-04-08

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - uuid_v1/2
  - uuid_v3/3
  - uuid_v4/1
  - uuid_v5/3
  - uuid_v7/1
  - uuid_null/1
  - uuid_nil/1
  - uuid_max/1
  - random_node/1
- Protected predicates
- Private predicates
- Operators

Public predicates

uuid_v1/2

Returns a version 1 UUID for the given MAC address (a list of six bytes). The MAC address can be replaced by a random 6 bytes node identifier as per RFC 4122 when the MAC address is not available or should not be disclosed.

Compilation flags:
    static

Template:
    uuid_v1(MAC,UUID)
Mode and number of proofs:
    uuid_v1(+list(byte),--text) - one

---

uuid_v3/3

Returns a version 3 UUID for the given namespace UUID and name. Namespace UUIDs and names can be represented as atoms, lists of characters, or lists of character codes. Name character codes must be bytes.

Compilation flags:
    static

Template:
    uuid_v3(Namespace,Name,UUID)
Mode and number of proofs:
    uuid_v3(+text,+text,--text) - one

---

uuid_v4/1

Returns a version 4 UUID.

Compilation flags:
    static

Template:

uuid_v4(UUID)
Mode and number of proofs:
uuid_v4(--text) - one

---

uuid_v5/3

Returns a version 5 UUID for the given namespace UUID and name. Namespace UUIDs and names can be represented as atoms, lists of characters, or lists of character codes. Name character codes must be bytes.

Compilation flags:
static

Template:
uuid_v5(Namespace,Name,UUID)
Mode and number of proofs:
uuid_v5(+text,+text,--text) - one

---

uuid_v7/1

Returns a version 7 UUID.

Compilation flags:
static

Template:
uuid_v7(UUID)
Mode and number of proofs:
uuid_v7(--text) - one

---

uuid_null/1

Returns the null UUID. Deprecated. Use uuid_nil/1 instead.

Compilation flags:
    static

Template:
    uuid_null(UUID)
Mode and number of proofs:
    uuid_null(--text) - one

---

uuid_nil/1

Returns the Nil UUID.

Compilation flags:
    static

Template:
    uuid_nil(UUID)
Mode and number of proofs:
    uuid_nil(--text) - one

---

uuid_max/1

Returns the Max UUID.

Compilation flags:
    static

Template:
    uuid_max(UUID)
Mode and number of proofs:
    uuid_max(--text) - one

---

random_node/1

Generates a list with six random bytes that can be used in alternative to a MAC address when generating version 1 UUIDs.

Compilation flags:
    static

Template:
    random_node(Node)
Mode and number of proofs:
    random_node(--list(byte)) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.234 validations

object

1.234.1 validated

Types and predicates for type-checking and handling lists of validation terms. Inspired by Scala Cats and Kotlin Arrow.

Availability:
    logtalk_load(validations(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-02-22

Compilation flags:
static, context_switching_calls

Provides:
type::type/1
type::check/2
arbitrary::arbitrary/1
arbitrary::arbitrary/2
arbitrary::shrinker/1
arbitrary::shrink/3
arbitrary::edge_case/2

Uses:
list
random
type

Remarks:

- Type-checking support: Defines a validated(ValueType, ErrorType) type for checking validation terms where the value and error terms must be of the given types.
- QuickCheck support: Defines clauses for the type::arbitrary/1-2, arbitrary::shrinker/1, arbitrary::shrink/3, and arbitrary::edge_case/2 predicates to allow generating random values for the validated(ValueType, ErrorType) type.

Inherited public predicates:
(none)

- Public predicates
  - valids/2
  - invalids/2
  - partition/3
  - map/3
  - map/4
  - sequence/2
  - traverse/3
- Protected predicates
- Private predicates
- Operators

Public predicates

valids/2

Returns the values stored in validation terms that hold valid values.

Compilation flags:
    static

Template:
    valids(Validations,Values)
Mode and number of proofs:
    valids(+list(validation),-list) - one

---

invalids/2

Returns a flattened list with all errors stored in invalid validation terms.

Compilation flags:
    static

Template:
    invalids(Validations,Errors)
Mode and number of proofs:
    invalids(+list(validation),-list) - one

---

partition/3

Retrieves and partitions valid values and flattened accumulated errors from validation terms.

Compilation flags:
    static

Template:
    partition(Validations,Values,Errors)
Mode and number of proofs:
    partition(+list(validation),-list,-list) - one

---

map/3

Applies a closure to each list element to generate validation terms and returns a pair Values-Errors accumulating all valid values and all errors in one pass.

Compilation flags:
    static

Template:
    map(Closure,Terms,ValuesErrors)
Meta-predicate template:
    map(2,*,*)
Mode and number of proofs:
    map(+callable,+list,--compound) - one

---

map/4

Applies a closure to each list element to generate validation terms and returns valid values and accumulated errors in one pass.

Compilation flags:
    static

Template:
    map(Closure,Terms,Values,Errors)
Meta-predicate template:
    map(2,*,*,*)
Mode and number of proofs:
    map(+callable,+list,-list,-list) - one

---

sequence/2

Sequences a list of validation terms into a single validation term, accumulating all errors.

Compilation flags:
    static

Template:
    sequence(Validations,Validation)
Mode and number of proofs:
    sequence(+list(validation),--nonvar) - one

---

traverse/3

Applies a closure to each list element to generate validation terms and then sequences them, accumulating all errors.

Compilation flags:
    static

Template:
    traverse(Closure,Terms,Validation)
Meta-predicate template:
    traverse(2,*,*)
Mode and number of proofs:
    traverse(+callable,+list,--nonvar) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

validation, validation(Validation), type, arbitrary

object

1.234.2 validation

Constructors for validation terms. A validation term is either valid(Value) or invalid(Errors) where Errors is a list of errors. Validation terms allow applicative-style error accumulation.

Availability:
logtalk_load(validations(loader))

Author: Paulo Moura
Version: 1:0:1
Date: 2026-04-22

Compilation flags:
static, context_switching_calls

Provides:
type::type/1
type::check/2
Uses:
type

Remarks:

- Type-checking support: Defines a validation type for use with the type library object.

Inherited public predicates:
(none)

- Public predicates
  - of_valid/2
  - of_invalid/2
  - of_invalids/2
  - from_goal/4
  - from_goal/3
  - from_goal/2
  - from_generator/4
  - from_generator/3
  - from_generator/2
  - from_optional/3
  - from_expected/2
- Protected predicates
- Private predicates
- Operators

Public predicates

of_valid/2

Constructs a validation term holding a valid value.

Compilation flags:
    static

Template:
    of_valid(Value,Validation)
Mode and number of proofs:
    of_valid(@term,--nonvar) - one

of_invalid/2

Constructs a validation term holding a single error.

Compilation flags:
    static

Template:
    of_invalid(Error,Validation)
Mode and number of proofs:
    of_invalid(@term,--nonvar) - one

---

of_invalids/2

Constructs a validation term holding a list of errors.

Compilation flags:
    static

Template:
    of_invalids(Errors,Validation)
Mode and number of proofs:
    of_invalids(@list,--nonvar) - one

---

from_goal/4

Constructs a validation term holding a value bound by calling the given goal. Otherwise returns a validation term with a single error represented by the Error argument.

Compilation flags:
    static

Template:
    from_goal(Goal,Value,Error,Validation)
Meta-predicate template:
    from_goal(0,*,*,*)
Mode and number of proofs:

from_goal(+callable,--term,@term,--nonvar) - one

---

from_goal/3

Constructs a validation term holding a value bound by calling the given goal. Otherwise returns a validation term with a single error being the goal error or the atom fail representing goal failure.

Compilation flags:
static

Template:
from_goal(Goal,Value,Validation)
Meta-predicate template:
from_goal(0,*,*)
Mode and number of proofs:
from_goal(+callable,--term,--nonvar) - one

---

from_goal/2

Constructs a validation term holding a value bound by calling the given closure. Otherwise returns a validation term with a single error being the closure error or the atom fail representing closure failure.

Compilation flags:
static

Template:
from_goal(Closure,Validation)
Meta-predicate template:
from_goal(1,*)
Mode and number of proofs:
from_goal(+callable,--nonvar) - one

---

from_generator/4

Constructs validation terms with the values generated by calling the given goal. On goal error or failure, returns a validation term with a single error represented by the Error argument.

Compilation flags:
    static

Template:
    from_generator(Goal,Value,Error,Validation)
Meta-predicate template:
    from_generator(0,*,*,*)
Mode and number of proofs:
    from_generator(+callable,--term,@term,--nonvar) - one_or_more

---

from_generator/3

Constructs validation terms with the values generated by calling the given goal. On goal error or failure, returns a validation term with, respectively, a single error being the goal error or the atom fail representing goal failure.

Compilation flags:
    static

Template:
    from_generator(Goal,Value,Validation)
Meta-predicate template:
    from_generator(0,*,*)
Mode and number of proofs:
    from_generator(+callable,--term,--nonvar) - one_or_more

---

from_generator/2

Constructs validation terms with the values generated by calling the given closure. On closure error or failure, returns a validation term with, respectively, a single error being the closure error or the atom fail representing closure failure.

Compilation flags:
    static

Template:
    from_generator(Closure,Validation)
Meta-predicate template:
    from_generator(1,*)
Mode and number of proofs:
    from_generator(+callable,--nonvar) - one_or_more

---

from_optional/3

Converts an optional term to a validation term. Returns a valid term holding the value if the optional term is not empty. Returns an invalid term with the given error otherwise.

Compilation flags:
    static

Template:
    from_optional(Optional,Error,Validation)
Mode and number of proofs:
    from_optional(+nonvar,@term,--nonvar) - one

---

from_expected/2

Converts an expected term to a validation term. Returns a valid term holding the value if the expected term holds a value. Returns an invalid term with the expected term error otherwise.

Compilation flags:
    static

Template:

from_expected(Expected,Validation)
Mode and number of proofs:
from_expected(+nonvar,--nonvar) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

validation(Validation), validated, type, arbitrary

object

1.234.3 validation(Validation)

Validation term predicates. Requires passing a validation term (constructed using the validation object predicates) as a parameter.

Availability:
logtalk_load(validations(loader))

Author: Paulo Moura
Version: 0:2:0
Date: 2026-02-21

Compilation flags:
static, context_switching_calls

Uses:
list

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - is_valid/0
  - is_invalid/0
  - if_valid/1
  - if_invalid/1
  - if_valid_or_else/2
  - valid/1
  - invalid/1
  - filter/3
  - map/2
  - flat_map/2
  - map_or_else/3
  - map_catching/2
  - map_invalid/2
  - map_both/3
  - swap/1
  - or/2
  - or_else/2
  - or_else_get/2
  - or_else_call/2
  - or_else_fail/1
  - or_else_throw/1
  - or_else_throw/2
  - zip/3
  - flatten/1
  - to_optional/1
  - to_expected/1
- Protected predicates
- Private predicates

- Operators

Public predicates

is_valid/0

True if the validation term holds a valid value.

Compilation flags:
    static

Mode and number of proofs:
    is_valid - zero_or_one

---

is_invalid/0

True if the validation term holds one or more errors.

Compilation flags:
    static

Mode and number of proofs:
    is_invalid - zero_or_one

---

if_valid/1

Applies a closure when the validation term holds a valid value using the value as argument. Succeeds otherwise.

Compilation flags:
    static

Template:
    if_valid(Closure)
Meta-predicate template:

if_valid(1)
Mode and number of proofs:
if_valid(+callable) - zero_or_more

---

if_invalid/1

Applies a closure when the validation term holds errors using the errors list as argument. Succeeds otherwise.

Compilation flags:
static

Template:
if_invalid(Closure)
Meta-predicate template:
if_invalid(1)
Mode and number of proofs:
if_invalid(+callable) - zero_or_more

---

if_valid_or_else/2

Applies either ValidClosure or InvalidClosure depending on the validation term holding a value or errors.

Compilation flags:
static

Template:
if_valid_or_else(ValidClosure,InvalidClosure)
Meta-predicate template:
if_valid_or_else(1,1)
Mode and number of proofs:
if_valid_or_else(+callable,+callable) - zero_or_more

---

valid/1

Returns the value hold by the validation term. Throws an error otherwise.

Compilation flags:
    static

Template:
    valid(Value)
Mode and number of proofs:
    valid(--term) - one_or_error

Exceptions:
    Validation term holds errors:
        existence_error(valid_value,Validation)

---

invalid/1

Returns the errors hold by the validation term. Throws an error otherwise.

Compilation flags:
    static

Template:
    invalid(Errors)
Mode and number of proofs:
    invalid(--list) - one_or_error

Exceptions:
    Validation term holds a valid value:
        existence_error(validation_errors,Validation)

---

filter/3

When the validation term holds a value and the value satisfies the closure, returns the same validation term. When the validation term holds a value that does not satisfy the closure, returns an invalid term with the given error. When the validation term holds errors, returns the same validation term.

Compilation flags:
    static

Template:
    filter(Closure,Error,NewValidation)
Meta-predicate template:
    filter(1,*,*)
Mode and number of proofs:
    filter(+callable,@term,--nonvar) - one

---

map/2

When the validation term holds a valid value and mapping a closure with the value and the new value as additional arguments succeeds, returns a new valid term. Otherwise returns the same validation term.

Compilation flags:
    static

Template:
    map(Closure,NewValidation)
Meta-predicate template:
    map(2,*)
Mode and number of proofs:
    map(+callable,--nonvar) - one

---

flat_map/2

When the validation term holds a valid value, applies a closure with the value and the new validation term as additional arguments. Returns the new validation term on success. When the validation term holds errors, short-circuits by returning the same validation term without calling the closure. This is the monadic escape hatch for dependent steps; use zip/3, sequence/2, or traverse/3 for error accumulation on independent steps.

Compilation flags:
    static

Template:
    flat_map(Closure,NewValidation)
Meta-predicate template:
    flat_map(2,*)
Mode and number of proofs:
    flat_map(+callable,--nonvar) - one

---

map_or_else/3

When the validation term holds a value and mapping a closure with the value and the new value as additional arguments is successful, returns the new value. Otherwise returns the given default value.

Compilation flags:
    static

Template:
    map_or_else(Closure,Default,Value)
Meta-predicate template:
    map_or_else(2,*,*)
Mode and number of proofs:
    map_or_else(+callable,@term,--term) - one

---

map_catching/2

When the validation term holds a value, applies a closure to it. Returns a valid term with the new value if the closure succeeds. Returns an invalid term with the error if the closure throws an error. Returns an invalid term with the atom fail as error if the closure fails. When the validation term holds errors, returns the same validation term.

Compilation flags:
    static

Template:
    map_catching(Closure,NewValidation)
Meta-predicate template:
    map_catching(2,*)
Mode and number of proofs:
    map_catching(+callable,--nonvar) - one

---

map_invalid/2

When the validation term holds errors and mapping a closure with the errors list and the new errors list as additional arguments succeeds, returns a new invalid term. Otherwise returns the same validation term.

Compilation flags:
    static

Template:
    map_invalid(Closure,NewValidation)
Meta-predicate template:
    map_invalid(2,*)
Mode and number of proofs:
    map_invalid(+callable,--nonvar) - one

---

map_both/3

When the validation term holds a value and mapping ValidClosure with the value is successful, returns a valid term with the new value. When the validation term holds errors and mapping InvalidClosure with the errors list is successful, returns an invalid term with the new errors. Otherwise returns the same validation term.

Compilation flags:
    static

Template:
    map_both(ValidClosure,InvalidClosure,NewValidation)
Meta-predicate template:
    map_both(2,2,*)
Mode and number of proofs:
    map_both(+callable,+callable,--nonvar) - one

---

swap/1

Swaps the valid and invalid terms. If the validation term holds a value, returns an invalid term with a singleton list containing that value. If the validation term holds errors, returns a valid term with the errors list.

Compilation flags:
    static

Template:
    swap(NewValidation)
Mode and number of proofs:
    swap(--nonvar) - one

---

or/2

Returns the same validation term if it holds a value. Otherwise calls closure to generate a new validation term. Fails if the validation term holds errors and calling the closure fails or throws an error.

Compilation flags:
    static

Template:
    or(NewValidation,Closure)
Meta-predicate template:
    or(*,1)
Mode and number of proofs:
    or(--term,@callable) - zero_or_one

---

or_else/2

Returns the valid value if present or the default value otherwise.

Compilation flags:
    static

Template:
    or_else(Value,Default)
Mode and number of proofs:
    or_else(--term,@term) - one

---

or_else_get/2

Returns the value hold by the validation term if valid. Otherwise applies a closure to compute the value. Throws an error when the validation term holds errors and a value cannot be computed.

Compilation flags:
    static

Template:
    or_else_get(Value,Closure)

Meta-predicate template:
    or_else_get(*,1)
Mode and number of proofs:
    or_else_get(--term,+callable) - one_or_error

Exceptions:
    Validation term holds errors and a value cannot be computed:
        existence_error(valid_value,Validation)

---

or_else_call/2

Returns the value hold by the validation term if valid. Calls a goal deterministically otherwise.

Compilation flags:
    static

Template:
    or_else_call(Value,Goal)
Meta-predicate template:
    or_else_call(*,0)
Mode and number of proofs:
    or_else_call(--term,+callable) - zero_or_one

---

or_else_fail/1

Returns the valid value if present. Fails otherwise.

Compilation flags:
    static

Template:
    or_else_fail(Value)
Mode and number of proofs:
    or_else_fail(--term) - zero_or_one

---

or_else_throw/1

Returns the value hold by the validation term if valid. Throws the errors list hold by the validation term as an exception otherwise.

Compilation flags:
    static

Template:
    or_else_throw(Value)
Mode and number of proofs:
    or_else_throw(--term) - one_or_error

---

or_else_throw/2

Returns the value hold by the validation term if valid. Throws the given error otherwise, ignoring any errors hold by the validation term.

Compilation flags:
    static

Template:
    or_else_throw(Value,Error)
Mode and number of proofs:
    or_else_throw(--term,@nonvar) - one_or_error

---

zip/3

When both this validation and the other validation hold values and applying a closure with both values and the new value as additional arguments is successful, returns a valid term with the new value. When both hold errors, returns an invalid term with all errors accumulated. Otherwise returns the first invalid term.

Compilation flags:
    static

Template:
    zip(Closure,OtherValidation,NewValidation)

Meta-predicate template:
zip(3,*,*)
Mode and number of proofs:
zip(+callable,+nonvar,--nonvar) - one

---

flatten/1

Flattens a nested validation term. When the validation term holds a value that is itself a validation term, returns the inner validation term. When the validation term holds a non-validation value, returns the same validation term. When the validation term holds errors, returns the same validation term.

Compilation flags:
static

Template:
flatten(NewValidation)
Mode and number of proofs:
flatten(--nonvar) - one

---

to_optional/1

Converts the validation term to an optional term. Returns a non-empty optional term holding the value if the validation term holds a value. Returns an empty optional term if the validation term holds errors.

Compilation flags:
static

Template:
to_optional(Optional)
Mode and number of proofs:
to_optional(--nonvar) - one

---

to_expected/1

Converts the validation term to an expected term. Returns an expected term holding the value if the validation term holds a value. Returns an expected term with the errors list as the unexpected error otherwise.

Compilation flags:
    static

Template:
    to_expected(Expected)
Mode and number of proofs:
    to_expected(--nonvar) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

validation, validated

1.235 verdi_neruda

object

1.235.1 a_star_interpreter(W)

A* interpreter for general logic programs. The parameter W is used to fine tune the behavior. W = 0 gives us a breadth-first search and W = 1 gives us a greedy best-first search. The default value for W is 0.5.

Availability:
logtalk_load(verdi_neruda(loader))

Author: Victor Lagerkvist
Version: 1:0:0
Date: 2010-06-13

Compilation flags:
static, context_switching_calls

Imports:
public best_first

Remarks:
(none)

Inherited public predicates:
prove/2 prove/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.235.2 benchmark_generators

Generates random data structures for use in benchmarks.

Availability:
logtalk_load(verdi_neruda(loader))

Author: Victor Lagerkvist
Version: 1:0:0
Date: 2010-06-13

Compilation flags:
static, context_switching_calls

Uses:
random

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - random_tree/1
- Protected predicates

- Private predicates
- Operators

Public predicates

random_tree/1

Generates a random tree.

Compilation flags:
    static

Template:
    random_tree(Tree)
Mode and number of proofs:
    random_tree(-tree) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.235.3 best_first

Best-first framework for general logic programs.

Availability:
    logtalk_load(verdi_neruda(loader))

Author: Victor Lagerkvist

Version: 1:1:1
Date: 2026-01-28

Compilation flags:
static

Implements:
public interpreterp
Uses:
counter
pairing_heap(Order)

Remarks:
(none)

Inherited public predicates:
prove/2 prove/3

- Public predicates
- Protected predicates
  - f/4
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

f/4

.

Compilation flags:
static

Template:
f(Length1,Length2,Depth,Cost)

Mode and number of proofs:
f(+float,+float,+float,-float) - zero_or_more

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.235.4 bfs_interpreter

Breadth-first interpreter for general logic programs.

Availability:
logtalk_load(verdi_neruda(loader))

Author: Victor Lagerkvist
Version: 1:0:0
Date: 2010-06-13

Compilation flags:
static, context_switching_calls

Implements:
public interpreterp
Uses:
counter
queue

Remarks:
(none)

Inherited public predicates:
prove/2 prove/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.235.5 bup_interpreter

Semi-naive bottom-up interpreter for general (stratified) logic programs. Magic transformation is realized through an expansion hook.

Availability:
logtalk_load(verdi_neruda(loader))

Author: Ulf Nilsson. Ported to Logtalk and augmented with negation by Victor Lagerkvist.
Version: 1:1:3
Date: 2023-11-30

Compilation flags:
static, context_switching_calls

Implements:
public interpreterp
Uses:
counter

list
magic
term

Remarks:
(none)

Inherited public predicates:
prove/2 prove/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.235.6 counter

Counter implemented with asserta/retract.

Availability:
logtalk_load(verdi_neruda(loader))

Author: Victor Lagerkvist

Version: 1:0:1
Date: 2022-10-08

Compilation flags:
static, context_switching_calls

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - increment/0
  - increase/1
  - set/1
  - value/1
  - reset/0
- Protected predicates
- Private predicates
  - c/1
- Operators

Public predicates

increment/0

Increment the counter by 1.

Compilation flags:
static

Mode and number of proofs:
increment - one

---

increase/1

Increments the counter by the specified amount.

Compilation flags:
    static

Template:
    increase(I)
Mode and number of proofs:
    increase(+number) - one

---

set/1

Sets the counter to the specified amount.

Compilation flags:
    static

Template:
    set(N)
Mode and number of proofs:
    set(+number) - one

---

value/1

Gets the current value of the counter.

Compilation flags:
    static

Template:
    value(N)
Mode and number of proofs:

value(?number) - one

---

reset/0

Resets the counter to zero.

Compilation flags:
    static

Mode and number of proofs:
    reset - one

Protected predicates

(none)

Private predicates

c/1

Stores the current value of the counter.

Compilation flags:
    dynamic

Template:
    c(N)
Mode and number of proofs:
    c(?number) - zero_or_one

Operators

(none)

protocol

1.235.7 databasep

Database protocol.

Availability:
logtalk_load(verdi_neruda(loader))

Author: Victor Lagerkvist
Version: 1:0:0
Date: 2010-06-13

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - rule/4
  - rule/3
  - rule/2
  - bench_goal/1
- Protected predicates
- Private predicates
- Operators

Public predicates

rule/4

Clauses for this predicate are automatically generated using term-expansion. The third argument contains the length of Body.

Compilation flags:
    static

Template:
    rule(Head,Body,Length,Tail)
Mode and number of proofs:
    rule(?callable,?callable,-,-) - zero_or_more

---

rule/3

Clauses for this predicate are automatically generated using term-expansion. The third argument denotes the tail of the Body.

Compilation flags:
    static

Template:
    rule(Head,Body,Tail)
Mode and number of proofs:
    rule(?callable,?callable,-) - zero_or_more

---

rule/2

Clauses for this predicate are automatically generated using term-expansion.

Compilation flags:
    static

Template:
    rule(Head,Body)

Mode and number of proofs:
rule(?callable,-list(callable)) - zero_or_more

---

bench_goal/1

Table of benchmark goals. They are used from shell.lgt to make benchmarking easier.

Compilation flags:
static

Template:
bench_goal(Goal)
Mode and number of proofs:
bench_goal(?callable) - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

object

1.235.8 debug_expansion(Mode)

Expands debug/1 calls. The parameter Mode can be either the atom "debug" or "production".

Availability:
logtalk_load(verdi_neruda(loader))

Author: Paulo Moura
Version: 1:0:0

Date: 2010-04-15

Compilation flags:
static, context_switching_calls

Implements:
public expanding

Remarks:
(none)

Inherited public predicates:
goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.235.9 demodb

Availability:
logtalk_load(verdi_neruda(loader))

Compilation flags:
static, context_switching_calls

Implements:
public databasep

Remarks:
(none)

Inherited public predicates:
bench_goal/1 rule/2 rule/3 rule/4

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.235.10 dfs_interpreter

Depth-first interpreter for general logic programs.

Availability:
logtalk_load(verdi_neruda(loader))

Author: Victor Lagerkvist
Version: 1:0:0
Date: 2010-06-13

Compilation flags:
static, context_switching_calls

Implements:
public interpreterp
Uses:
counter

Remarks:
(none)

Inherited public predicates:
prove/2 prove/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

category

1.235.11 flatting

Flattens conjunction of goals with the form f and g into a list [f,g].

Availability:
logtalk_load(verdi_neruda(loader))

Author: Victor Lagerkvist
Version: 1:1:0
Date: 2025-10-06

Compilation flags:
static

source: Based on source code from The Craft of Prolog, by Richard O'Keefe.

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
- Protected predicates
  - flatten_goals//1
- Private predicates
- Operators

Public predicates

(none)

Protected predicates

flatten_goals//1

Flattens a conjunction of goals.

Compilation flags:
    static

Template:
    flatten_goals(Conjunction)
Mode and number of proofs:
    flatten_goals(+callable) - one

Private predicates

(none)

Operators

(none)

object

1.235.12 heuristic_expansion(Mode)

Expands rules of the form p if f and g to rule(p, [f,g|Tail], Length, Tail).

Availability:
logtalk_load(verdi_neruda(loader))

Author: Victor Lagerkvist
Version: 1:0:2
Date: 2022-10-08

Compilation flags:
static, context_switching_calls

Implements:
public expanding
Extends:
public rule_expansion(Mode)
Uses:
list

Remarks:
(none)

Inherited public predicates:
goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.235.13 iddfs_interpreter(Increment)

Iterative deepening depth-first interpreter for general logic programs. Based on source code from The Craft of Prolog, by Richard O'Keefe. The default value for the increment is 1.

Availability:
logtalk_load(verdi_neruda(loader))

Author: Victor Lagerkvist
Version: 1:0:0
Date: 2010-06-13

Compilation flags:
static, context_switching_calls

Implements:
public interpreterp
Uses:
counter
dfs_interpreter

Remarks:
(none)

Inherited public predicates:
prove/2 prove/3

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.235.14 interpreterp

Protocol for an interpreter.

Availability:
logtalk_load(verdi_neruda(loader))

Author: Victor Lagerkvist
Version: 1:0:0
Date: 2010-06-13

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - prove/2
  - prove/3
- Protected predicates
- Private predicates
- Operators

Public predicates

prove/2

True if goal is provable in the specified database.

Compilation flags:
static

Template:
prove(Goal,DB)
Mode and number of proofs:
prove(+goal,+database) - zero_or_more

---

prove/3

True if goal is provable within the given depth-limit in the specified database.

Compilation flags:
static

Template:
prove(Goal,Limit,DB)

Mode and number of proofs:
prove(+goal,+limit,+database) - zero_or_more

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

object

1.235.15 magic

Object encapsulating magic methods.

Availability:
logtalk_load(verdi_neruda(loader))

Author: Ulf Nilsson. Ported to Logtalk and augmented with stratified negation by Victor Lagerkvist.
Version: 1:0:0
Date: 2010-06-13

Compilation flags:
static, context_switching_calls

Uses:
list

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - magicise/4
  - magic/2
- Protected predicates
- Private predicates
- Operators

Public predicates

magicise/4

Transform (Head :- Body) into a magic clause (NewHead :- NewBody).

Compilation flags:
    static

Template:
    magicise(Head,Body,NewHead,NewBody)
Mode and number of proofs:
    magicise(+term,+list,-term,-list) - zero_or_one

---

magic/2

Prefix the predicate symbol of Old with magic.

Compilation flags:
    static

Template:
    magic(Old,New)
Mode and number of proofs:
    magic(+callable,-callable) - zero_or_one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.235.16 magic_expansion(Mode)

Expands rules of the form p if f and g to the more manageable rule(p, [f,g]) and performs magic transformation of clauses.

Availability:
logtalk_load(verdi_neruda(loader))

Author: Victor Lagerkvist
Version: 1:0:2
Date: 2022-10-08

Compilation flags:
static, context_switching_calls

Implements:
public expanding
Imports:
public flatting
Extends:
public debug_expansion(Mode)
Uses:
list
magic

Remarks:
(none)

Inherited public predicates:
goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.235.17 rule_expansion(Mode)

Expands rules of the form p if f and g to the more manageable rule(p, [f,g]).

Availability:
    logtalk_load(verdi_neruda(loader))

Author: Victor Lagerkvist
Version: 1:0:2
Date: 2022-10-08

Compilation flags:
    static, context_switching_calls

Implements:
    public expanding
Imports:
    public flatting

Extends:
public debug_expansion(Mode)

Remarks:
(none)

Inherited public predicates:
goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.235.18 shell

User frontend to start the application.

Availability:
logtalk_load(verdi_neruda(loader))

Author: Paulo Moura
Version: 1:0:0

Date: 2019-03-20

Compilation flags:
static, context_switching_calls

Uses:
shell(Interpreters)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - welcome/0
  - start/0
- Protected predicates
- Private predicates
- Operators

Public predicates

welcome/0

Compilation flags:
static

---

start/0

Compilation flags:
static

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.235.19 shell(Interpreters)

Prolog shell for the interpreters.

Availability:
logtalk_load(verdi_neruda(loader))

Author: Victor Lagerkvist and Paulo Moura
Version: 1:1:3
Date: 2024-03-15

Compilation flags:
static, context_switching_calls

Uses:
counter
list
meta
pairs

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - init/0
- Protected predicates
- Private predicates
- Operators

Public predicates

init/0

Compilation flags:
    static

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

object

1.235.20 shell_expansion(Mode)

Expansion object for the shell.

Availability:
    logtalk_load(verdi_neruda(loader))

Author: Victor Lagerkvist
Version: 1:0:1
Date: 2022-10-08

Compilation flags:

static, context_switching_calls

Implements:
public expanding
Extends:
public rule_expansion(Mode)

Remarks:
(none)

Inherited public predicates:
goal_expansion/2 term_expansion/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

1.236 wkt_wkb

object

1.236.1 wkt_wkb

Well-Known Text (WKT) and Well-Known Binary (WKB) geometry parser, generator, and validator.

Availability:
logtalk_load(wkt_wkb(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-09

Compilation flags:
static, context_switching_calls

Implements:
public wkt_wkb_protocol
Uses:
ieee_754(Precision,ByteOrder,NaNRepresentation)
ieee_754_fields(Precision,ByteOrder)
list
reader

Remarks:
(none)

Inherited public predicates:
generate/2 parse/2 validate/1 validate/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

wkt_wkb_protocol, geojson, geospatial, cbor, message_pack(StringRepresentation)

protocol

1.236.2 wkt_wkb_protocol

Well-Known Text (WKT) and Well-Known Binary (WKB) geometry parser, generator, and validator protocol.

Availability:
logtalk_load(wkt_wkb(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-09

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - parse/2
  - generate/2
  - validate/1
  - validate/2
- Protected predicates
- Private predicates
- Operators

Public predicates

parse/2

Parses a geometry from the given source. Supported sources are wkt(file(Path)), wkt(stream(Stream)), wkt(atom(Atom)), wkt(chars(List)), wkt(codes(List)), wkb(file(Path)), wkb(stream(Stream)), wkb(bytes(List)), wkb(hex(atom(Atom))), wkb(hex(chars(List))), and wkb(hex(codes(List))).

Compilation flags:
static

Template:
parse(Source,Geometry)
Mode and number of proofs:
parse(++compound,--term) - one_or_error

Exceptions:
Source is a variable:
instantiation_error
Source is neither a variable nor a valid source:
domain_error(wkt_wkb_source,Source)
Source data does not represent a valid WKT or WKB geometry:
domain_error(wkt_wkb,Source)

generate/2

Generates WKT or WKB content using the representation specified in the first argument from a native geometry term. Supported sinks are wkt(file(Path)), wkt(stream(Stream)), wkt(atom(Atom)), wkt(chars(List)), wkt(codes(List)), wkb(file(Path)), wkb(stream(Stream)), wkb(bytes(List)), wkb(hex(atom(Atom))), wkb(hex(chars(List))), wkb(hex(codes(List))), and the same WKB sink wrappers wrapped as wkb(Sink, Order) where Order is little or big.

Compilation flags:
    static

Template:
    generate(Sink,Geometry)
Mode and number of proofs:
    generate(+compound,++term) - one_or_error

Exceptions:
    Sink is a variable:
        instantiation_error
    Sink is neither a variable nor a valid sink:
        domain_error(wkt_wkb_sink,Sink)
    Geometry is not a valid native geometry term:
        domain_error(wkt_wkb,Geometry)
    Geometry is valid but cannot be represented in WKT due to special ordinates:
        domain_error(wkt_wkb_representation,wkt(Geometry))

---

validate/1

True iff the argument is a valid native WKT/WKB geometry term.

Compilation flags:
    static

Template:
    validate(Geometry)
Mode and number of proofs:
    validate(@term) - zero_or_one

---

validate/2

Validates a native geometry term returning a list of reason terms whose final argument is the failing path.

Compilation flags:
    static

Template:
    validate(Geometry,Errors)
Mode and number of proofs:
    validate(@term,-list(compound)) - one_or_error

Remarks:

- invalid_geometry_term(Path): The term is not any supported geometry representation.
- invalid_options(Path): The options argument is not a list.
- unknown_option(Option, Path): An option term is not recognized.
- duplicate_option(Name, Path): The same option name occurs more than once.
- invalid_dimensions(Path): A dimensions/1 option does not use one of the supported values xy, z, m, or zm.
- invalid_position(Path): A position is not a list of numeric or supported special IEEE-754 coordinates with the expected dimensionality.
- invalid_position_array(Path): A coordinate array expected to contain positions is not a list.
- mixed_coordinate_dimension(Path): Coordinates in the same geometry do not all have the same dimensionality.
- coordinate_dimension_mismatch(Actual, Expected, Path): The coordinates dimensionality does not match the dimensions/1 option.
- insufficient_positions(MinimumLength, Path): A coordinate array contains fewer positions than required for the enclosing geometry.
- ring_not_closed(Path): A polygon ring does not repeat its first position as its last position.
- invalid_polygon(Path): A polygon coordinate structure is invalid.
- invalid_multi_polygon(Path): A multi-polygon coordinate structure is invalid.
- invalid_geometry_collection(Path): A geometry collection member is not a list of valid geometry terms.

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

wkt_wkb, geojson, geospatial

1.237 wrapper

object

1.237.1 wrapper

Adviser tool for porting and wrapping plain Prolog applications.

Availability:
logtalk_load(wrapper(loader))

Author: Paulo Moura
Version: 0:12:4
Date: 2026-01-20

Compilation flags:
static, context_switching_calls

Implements:
public expanding
Provides:
logtalk::message_hook/4
logtalk::message_prefix_stream/4
logtalk::message_tokens//2
Uses:
logtalk
os

Remarks:

- prolog_extensions(Extensions) option: List of file name extensions used to recognize Prolog source files (default is ['.pl','.pro','.prolog']).
- logtalk_extension(Extension) option: Logtalk file name extension to be used for the generated wrapper files (default is '.lgt').
- exclude_files(Files) option: List of Prolog source files names to exclude (default is []).
- exclude_directories(Directories) option: List of sub-directory names to exclude (default is []).
- include_wrapped_files(Boolean): Generate include/1 directives for the wrapped Prolog source files (default is true).

Inherited public predicates:

goal_expansion/2 term_expansion/2

- Public predicates
  - rdirectory/2
  - rdirectory/1
  - directory/2
  - directory/1
  - directories/2
  - directories/1
  - files/2
  - files/1
  - file/2
  - file/1
  - save/1
  - save/0
  - default_option/1
  - default_options/1
- Protected predicates
- Private predicates
  - merge_options/2
  - predicate_called_but_not_defined_/2
  - object_predicate_called_/3
  - module_predicate_called_/3
  - unknown_predicate_called_/2

  - missing_predicate_directive_/3
  - non_standard_predicate_call_/2
  - dynamic_directive_/3
  - multifile_directive_/3
  - add_directive_before_entity_/2
  - add_directive_/2
  - add_directive_/3
  - remove_directive_/2
  - file_being_advised_/4
- Operators

Public predicates

rdirectory/2

Advises the user on missing directives for converting all plain Prolog files in a directory and its sub-directories to Logtalk objects using the specified options.

Compilation flags:
    static

Template:
    rdirectory(Directory,Options)
Mode and number of proofs:
    rdirectory(+atom,+list(compound)) - one

---

rdirectory/1

Advises the user on missing directives for converting all plain Prolog files in a directory and its sub-directories to Logtalk objects using default options.

Compilation flags:
    static

Template:
    rdirectory(Directory)
Mode and number of proofs:

rdirectory(+atom) - one

---

directory/2

Advises the user on missing directives for converting all plain Prolog files in a directory to Logtalk objects using the specified options.

Compilation flags:
static

Template:
directory(Directory,Options)
Mode and number of proofs:
directory(+atom,+list(compound)) - one

---

directory/1

Advises the user on missing directives for converting all plain Prolog files in a directory to Logtalk objects using default options.

Compilation flags:
static

Template:
directory(Directory)
Mode and number of proofs:
directory(+atom) - one

---

directories/2

Advises the user on missing directives for converting all Prolog files in a set of directories to Logtalk objects using the specified options.

Compilation flags:
    static

Template:
    directories(Directories,Options)
Mode and number of proofs:
    directories(+list(atom),+list(compound)) - one

---

directories/1

Advises the user on missing directives for converting all Prolog files in a set of directories to Logtalk objects using default options.

Compilation flags:
    static

Template:
    directories(Directories)
Mode and number of proofs:
    directories(+list(atom)) - one

---

files/2

Advises the user on missing directives for converting a list of plain Prolog files to Logtalk objects using the specified options.

Compilation flags:
    static

Template:
    files(Files,Options)
Mode and number of proofs:

files(+list(atom),+list(compound)) - one

---

files/1

Advises the user on missing directives for converting a list of plain Prolog files to Logtalk objects using default options.

Compilation flags:
static

Template:
files(Files)
Mode and number of proofs:
files(+list(atom)) - one

---

file/2

Advises the user on missing directives for converting a plain Prolog file to Logtalk objects using the specified options.

Compilation flags:
static

Template:
file(File,Options)
Mode and number of proofs:
file(+atom,+list(compound)) - one

---

file/1

Advises the user on missing directives for converting a plain Prolog file to Logtalk objects using default options.

Compilation flags:
    static

Template:
    file(File)
Mode and number of proofs:
    file(+atom) - one

---

save/1

Saves the generated wrapper objects (plus a loader file per directory) for all advised files using the specified options. The wrapper objects are saved to the same directories that contain the wrapped Prolog files.

Compilation flags:
    static

Template:
    save(Options)
Mode and number of proofs:
    save(+list(compound)) - one

---

save/0

Saves the generated wrapper objects (plus a loader file per directory) for all advised files using default options. The wrapper objects are saved to the same directories that contain the wrapped Prolog files.

Compilation flags:
    static

Mode and number of proofs:
    save - one

---

default_option/1

Enumerates by backtracking the default options used when generating the wrapper objects.

Compilation flags:
    static

Template:
    default_option(DefaultOption)
Mode and number of proofs:
    default_option(?compound) - zero_or_more

---

default_options/1

Returns a list of the default options used when generating the wrapper objects.

Compilation flags:
    static

Template:
    default_options(DefaultOptions)
Mode and number of proofs:
    default_options(-list(compound)) - one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

merge_options/2

Merges the user options with the default options, returning the list of options used when generating the wrapper objects.

Compilation flags:
    static

Template:
merge_options(UserOptions,Options)
Mode and number of proofs:
merge_options(+list(compound),-list(compound)) - one

---

predicate_called_but_not_defined_/2

Table of called object predicates that are not locally defined.

Compilation flags:
dynamic

Template:
predicate_called_but_not_defined_(Object,Predicate)
Mode and number of proofs:
predicate_called_but_not_defined_(?atom,?predicate_indicator) - zero_or_more

---

object_predicate_called_/3

Table of called object predicates.

Compilation flags:
dynamic

Template:
object_predicate_called_(Object,Other,Predicate)
Mode and number of proofs:
object_predicate_called_(?atom,?atom,?predicate_indicator) - zero_or_more

---

module_predicate_called_/3

Table of called module predicates.

Compilation flags:
dynamic

Template:
module_predicate_called_(Object,Module,Predicate)
Mode and number of proofs:
module_predicate_called_(?atom,?atom,?predicate_indicator) - zero_or_more

---

unknown_predicate_called_/2

Table of predicates called but not defined.

Compilation flags:
dynamic

Template:
unknown_predicate_called_(Object,Predicate)
Mode and number of proofs:
unknown_predicate_called_(?atom,?predicate_indicator) - zero_or_more

---

missing_predicate_directive_/3

Table of missing predicate directives.

Compilation flags:
dynamic

Template:
missing_predicate_directive_(Object,Directive,Predicate)
Mode and number of proofs:
missing_predicate_directive_(?atom,?predicate_indicator,?predicate_indicator) - zero_or_more

---

non_standard_predicate_call_/2

Table of called non-standard predicates.

Compilation flags:
    dynamic

Template:
    non_standard_predicate_call_(Object,Predicate)
Mode and number of proofs:
    non_standard_predicate_call_(?atom,?predicate_indicator) - zero_or_more

---

dynamic_directive_/3

Table of declared dynamic predicates.

Compilation flags:
    dynamic

Template:
    dynamic_directive_(Object,Line,Predicate)
Mode and number of proofs:
    dynamic_directive_(?atom,?integer,?predicate_indicator) - zero_or_more

---

multifile_directive_/3

Table of declared multifile predicates.

Compilation flags:
    dynamic

Template:
    multifile_directive_(Object,Line,Predicate)
Mode and number of proofs:
    multifile_directive_(?atom,?integer,?predicate_indicator) - zero_or_more

---

add_directive_before_entity_/2

Table of directives to be added before the entity opening directive.

Compilation flags:
dynamic

Template:
add_directive_before_entity_(Object,Directive)
Mode and number of proofs:
add_directive_before_entity_(?atom,?predicate_indicator) - zero_or_more

---

add_directive_/2

Table of directives to be added.

Compilation flags:
dynamic

Template:
add_directive_(Object,Directive)
Mode and number of proofs:
add_directive_(?atom,?predicate_indicator) - zero_or_more

---

add_directive_/3

Table of directives to be added to complement existing directives.

Compilation flags:
dynamic

Template:
add_directive_(Object,Directive,NewDirective)
Mode and number of proofs:
add_directive_(?atom,?predicate_indicator,?predicate_indicator) - zero_or_more

---

remove_directive_/2

Table of directives to be removed.

Compilation flags:
dynamic

Template:
remove_directive_(Object,Directive)
Mode and number of proofs:
remove_directive_(?atom,?predicate_indicator) - zero_or_more

---

file_being_advised_/4

Table of files being advised are respective directories and names (basename without extension).

Compilation flags:
dynamic

Template:
file_being_advised_(File,Path,Directory,Name)
Mode and number of proofs:
file_being_advised_(?atom,?atom,?atom,?atom) - zero_or_more

Operators

(none)

1.238 xml_parser

object

1.238.1 xml

Bi-directional XML parser.

Availability:
logtalk_load(xml_parser(loader))

Author: John Fletcher; adapted to Logtalk by Paulo Moura.
Version: 3:9:0
Date: 2025-10-06

Copyright: Copyright (C) 2001-2005 Binding Time Limited, Copyright (C) 2005-2013 John Fletcher
License: This program is offered free of charge, as unsupported source code. You may use it, copy it, distribute it, modify it or sell it without restriction, but entirely at your own risk.

Compilation flags:
static, context_switching_calls

Uses:
list
term

Remarks:

- On-line documentation: https://binding-time.co.uk/index.php/Parsing_XML_with_Prolog
- Compliance: This XML parser supports a subset of XML suitable for XML Data and Worldwide Web applications. It is neither as strict nor as comprehensive as the XML 1.0 Specification mandates.
- Compliance-strictness: It is not as strict, because, while the specification must eliminate ambiguities, not all errors need to be regarded as faults, and some reasonable examples of real XML usage would have to be rejected if they were.
- Compliance-comprehensive: It is not as comprehensive, because, where the XML specification makes provision for more or less complete DTDs to be provided as part of a document, xml.pl actions the local definition of ENTITIES only. Other DTD extensions are treated as commentary.
- Bi-directional conversions: Conversions are not fully symmetrical as weaker XML is accepted than can be generated. Notably, in-bound (Codes -> Document) parsing does not require strictly well-formed XML. If Codes does not represent well-formed XML, Document is instantiated to the term malformed(<attributes>,<content>).

Inherited public predicates:
(none)

- Public predicates
  - parse/2
  - parse/3
  - subterm/2
  - pp/1
- Protected predicates
- Private predicates
  - xml_to_document/3
  - empty_map/1
  - map_member/3
  - map_store/4
  - pp_string/1
  - fault/5
  - exception/4
  - document_generation//2
  - pcdata_7bit//1
  - character_data_format/3
  - cdata_generation//1
- Operators

Public predicates

parse/2

Parses a list of character codes to/from a data structure of the form xml(<atts>,<content>).

Compilation flags:
    static

Template:
    parse(Codes,Document)
Mode and number of proofs:
    parse(+list(character_code),?nonvar) - zero_or_one
    parse(?list(character_code),+nonvar) - zero_or_one

parse/3

Parses a list of character codes to/from a data structure of the form xml(<atts>,<content>) using the given list of options.

Compilation flags:
static

Template:
parse(Options,Codes,Document)
Mode and number of proofs:
parse(++list(compound),+list(character_code),?nonvar) - zero_or_one
parse(++list(compound),?list(character_code),+nonvar) - zero_or_one

Remarks:

- extended_characters(Boolean) option: Use the extended character entities for XHTML (default true).
- format(Boolean) option: For parsing, strip layouts when no character data appears between elements (default true). For generating, indent the element content (default true).
- remove_attribute_prefixes(Boolean) option: Remove namespace prefixes from attributes when it's the same as the prefix of the parent element (default false).
- allow_ampersand(Boolean) option: Allow unescaped ampersand characters (&) to occur in PCDATA (default false).

---

subterm/2

Unifies Subterm with a sub-term of XMLTerm. Note that XMLTerm is a sub-term of itself.

Compilation flags:
static

Template:
subterm(XMLTerm,Subterm)
Mode and number of proofs:
subterm(+nonvar,?nonvar) - zero_or_one

---

pp/1

Pretty prints a XML document on the current output stream.

Compilation flags:
    static

Template:
    pp(XMLDocument)
Mode and number of proofs:
    pp(+nonvar) - zero_or_one

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

xml_to_document/3

Translates the list of character codes XML into the Prolog term Document. Options is a list of terms controlling the treatment of layout characters and character entities.

Compilation flags:
    static

Template:
    xml_to_document(Options,XML,Document)
Mode and number of proofs:
    xml_to_document(+nonvar,+nonvar,?nonvar) - zero_or_one

---

empty_map/1

True if Map is a null map.

Compilation flags:
    static

Template:
    empty_map(Map)
Mode and number of proofs:
    empty_map(?nonvar) - zero_or_one

---

map_member/3

True if Map is a ordered map structure which records the pair Key-Data. Key must be ground.

Compilation flags:
    static

Template:
    map_member(Key,Map,Data)
Mode and number of proofs:
    map_member(+nonvar,+nonvar,?nonvar) - zero_or_one

---

map_store/4

True if Map0 is an ordered map structure, Key must be ground, and Map1 is identical to Map0 except that the pair Key-Data is recorded by Map1.

Compilation flags:
    static

Template:
    map_store(Map0,Key,Data,Map1)
Mode and number of proofs:
    map_store(+nonvar,+nonvar,+nonvar,?nonvar) - zero_or_one

---

pp_string/1

Prints String onto the current output stream. If String contains only 7-bit chars it is printed in shorthand quoted format, otherwise it is written as a list.

Compilation flags:
    static

Template:
    pp_string(String)
Mode and number of proofs:
    pp_string(+nonvar) - zero_or_one

---

fault/5

Identifies SubTerm as a sub-term of Term which cannot be serialized after Indentation. Message is an atom naming the type of error; Path is a string encoding a list of SubTerm's ancestor elements in the form <tag>{(id)}* where <tag> is the element tag and <id> is the value of any attribute _named_ id.

Compilation flags:
    static

Template:
    fault(Term,Indentation,SubTerm,Path,Message)
Mode and number of proofs:
    fault(+nonvar,+nonvar,?nonvar,?nonvar,?nonvar) - zero_or_one

---

exception/4

Hook to raise an exception to be raised in respect of a fault in the XML Term Document.

Compilation flags:
    static

Template:

exception(Message,Document,Culprit,Path)
Mode and number of proofs:
exception(+atom,+nonvar,+nonvar,+nonvar) - one

---

document_generation//2

DCG generating Document as a list of character codes. Format is true|false defining whether layouts, to provide indentation, should be added between the element content of the resultant "string". Note that formatting is disabled for elements that are interspersed with pcdata/1 terms, such as XHTML's 'inline' elements. Also, Format is over-ridden, for an individual element, by an explicit 'xml:space'="preserve" attribute.

Compilation flags:
static

Template:
document_generation(Format,Document)
Mode and number of proofs:
document_generation(+nonvar,+nonvar) - zero_or_one

---

pcdata_7bit//1

Represents the ASCII character set in its simplest format, using the character entities &, ", <, and > which are common to both XML and HTML. The numeric entity ' is used in place of ' because browsers don't recognize it in HTML.

Compilation flags:
static

Template:
pcdata_7bit(Code)
Mode and number of proofs:
pcdata_7bit(?nonvar) - zero_or_one

---

character_data_format/3

Holds when Format0 and Format1 are the statuses of XML formatting before and after Codes - which may be null.

Compilation flags:
    static

Template:
    character_data_format(Codes,Format0,Format1)
Mode and number of proofs:
    character_data_format(+nonvar,+nonvar,?nonvar) - zero_or_one

---

cdata_generation//1

Holds when Format0 and Format1 are the statuses of XML formatting before and after Codes - which may be null.

Compilation flags:
    static

Template:
    cdata_generation(Codes)
Mode and number of proofs:
    cdata_generation(+list) - zero_or_one

Operators

(none)

1.239 yaml

object

1.239.1 yaml

YAML parser and generator.

Availability:
logtalk_load(yaml(loader))

Author: Paulo Moura
Version: 1:1:0
Date: 2026-04-17

Compilation flags:
static, context_switching_calls

Implements:
public yaml_protocol
Uses:
list
reader

Remarks:
(none)

Inherited public predicates:
generate/2 generate_all/2 parse/2 parse_all/2

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

protocol

1.239.2 yaml_protocol

YAML parser and generator protocol.

Availability:
logtalk_load(yaml(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-01-31

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - parse/2
  - parse_all/2
  - generate/2
  - generate_all/2
- Protected predicates
- Private predicates
- Operators

Public predicates

parse/2

Parses YAML content read from the given source (file(Path), stream(Stream), codes(Codes), chars(Chars), or atom(Atom)) into a ground term representing the parsed YAML data.

Compilation flags:
    static

Template:
    parse(Source,YAML)
Mode and number of proofs:
    parse(++compound,--ground) - one_or_error

Exceptions:
    Source is a variable:
        instantiation_error
    Source is neither a variable nor a valid source:
        domain_error(yaml_source,Source)

---

parse_all/2

Parses all YAML documents from the given source (file(Path), stream(Stream), codes(Codes), chars(Chars), or atom(Atom)) into a list of ground terms. Documents are separated by --- markers and optionally terminated by ... markers.

Compilation flags:
    static

Template:
    parse_all(Source,YAMLs)
Mode and number of proofs:
    parse_all(++compound,--list(ground)) - one_or_error

Exceptions:
    Source is a variable:
        instantiation_error
    Source is neither a variable nor a valid source:
        domain_error(yaml_source,Source)

---

generate/2

Generates YAML output using the representation specified in the first argument (file(Path), stream(Stream), codes(Codes), chars(Chars), or atom(Atom)) from the ground YAML term in the second argument.

Compilation flags:
    static

Template:
    generate(Sink,YAML)
Mode and number of proofs:
    generate(++compound,+ground) - one_or_error

Exceptions:
    Sink is a variable:
        instantiation_error
    YAML is a variable:
        instantiation_error
    YAML is not a valid YAML term:
        domain_error(yaml_term,YAML)

Sink cannot be generated:
domain_error(yaml_sink,Sink)

---

generate_all/2

Generates YAML output with multiple documents separated by --- markers using the representation specified in the first argument (file(Path), stream(Stream), codes(Codes), chars(Chars), or atom(Atom)) from the list of ground YAML terms in the second argument.

Compilation flags:
static

Template:
generate_all(Sink,YAMLs)
Mode and number of proofs:
generate_all(++compound,+list(ground)) - one_or_error

Exceptions:
Sink is a variable:
instantiation_error
YAMLs is a variable:
instantiation_error
YAMLs is not a list:
type_error(list,YAMLs)
An element of YAMLs is not a valid YAML term:
domain_error(yaml_term,Term)
Sink cannot be generated:
domain_error(yaml_sink,Sink)

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

1.240 z_score_anomaly_detector

object

1.240.1 z_score_anomaly_detector

Statistical Z-score anomaly detector for continuous datasets. Learns per-attribute population mean and standard deviation from baseline training examples selected from a dataset object implementing the anomaly_dataset_protocol protocol and returns a detector term that can be used for scoring, prediction, and export.

Availability:
logtalk_load(z_score_anomaly_detector(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2026-05-06

Compilation flags:
static, context_switching_calls

Imports:
public anomaly_detector_common
Uses:
format
list
numberlist
pairs
population
type

Remarks:
(none)

Inherited public predicates:

anomaly_detector_options/2 check_anomaly_detector/1 check_option/1 check_options/1 default_option/1 default_options/1 diagnostic/2 diagnostics/2 export_to_clauses/4 export_to_file/4 learn/2 learn/3 option/2 option/3 predict/3 predict/4 print_anomaly_detector/1 score/3 score_all/3 valid_anomaly_detector/1 valid_option/1 valid_options/1

- Public predicates
- Protected predicates
- Private predicates
- Operators

Public predicates

(no local declarations; see entity ancestors if any)

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

See also

anomaly_dataset_protocol, anomaly_detector_protocol, modified_z_score_anomaly_detector, isolation_forest_anomaly_detector, knn_distance_anomaly_detector, lof_anomaly_detector

1.241 zippers

protocol

1.241.1 zipperp

Zipper protocol.

Availability:
logtalk_load(zippers(loader))

Author: Paulo Moura
Version: 1:0:0
Date: 2019-01-20

Compilation flags:
static

Dependencies:
(none)

Remarks:
(none)

Inherited public predicates:
(none)

- Public predicates
  - zip/2
  - zip/3
  - unzip/2
  - current/2
  - next/2
  - next/3
  - previous/2
  - previous/3
  - rewind/2
  - rewind/3
  - forward/2
  - forward/3
  - apply/2

  - insert_before/3
  - insert_after/3
  - replace/3
  - delete_and_previous/2
  - delete_and_next/2
  - delete_and_unzip/2
  - delete_all_before/2
  - delete_all_before_and_unzip/2
  - delete_all_after/2
  - delete_all_after_and_unzip/2
- Protected predicates
- Private predicates
- Operators

Public predicates

zip/2

Adds a zipper to a compound term holding a sequence of elements. Fails if the sequence is empty.

Compilation flags:
    static

Template:
    zip(Sequence,Zipper)
Mode and number of proofs:
    zip(+sequence,--zipper) - zero_or_one

---

zip/3

Adds a zipper to a compound term holding a sequence of elements. Also returns the first element. Fails if the sequence is empty.

Compilation flags:
    static

Template:
zip(Sequence,Zipper,First)
Mode and number of proofs:
zip(+sequence,--zipper,--term) - zero_or_one

---

unzip/2

Removes a zipper from a sequence.

Compilation flags:
static

Template:
unzip(Zipper,Sequence)
Mode and number of proofs:
unzip(@zipper,--sequence) - one

---

current/2

Current element.

Compilation flags:
static

Template:
current(Zipper,Current)
Mode and number of proofs:
current(+zipper,?term) - zero_or_one

---

next/2

Moves to the next element. Fails if already at the last elements.

Compilation flags:
    static

Template:
    next(Zipper,NewZipper)
Mode and number of proofs:
    next(+zipper,--zipper) - zero_or_one

---

next/3

Moves to and returns the next element. Fails if already at the last elements.

Compilation flags:
    static

Template:
    next(Zipper,NewZipper,Next)
Mode and number of proofs:
    next(+zipper,--zipper,-term) - zero_or_one

---

previous/2

Moves to the previous element. Fails if already at the first elements.

Compilation flags:
    static

Template:
    previous(Zipper,NewZipper)
Mode and number of proofs:
    previous(+zipper,--zipper) - zero_or_one

---

previous/3

Moves to and returns the previous element. Fails if already at the first element.

Compilation flags:
    static

Template:
    previous(Zipper,NewZipper,Previous)
Mode and number of proofs:
    previous(+zipper,--zipper,-term) - zero_or_one

---

rewind/2

Rewinds the zipper so that the first element becomes the current element.

Compilation flags:
    static

Template:
    rewind(Zipper,NewZipper)
Mode and number of proofs:
    rewind(+zipper,--zipper) - one

---

rewind/3

Rewinds the zipper so that the first element becomes the current element. Also returns the first element.

Compilation flags:
    static

Template:
    rewind(Zipper,NewZipper,First)
Mode and number of proofs:
    rewind(+zipper,--zipper,?term) - zero_or_one

---

forward/2

Forward the zipper so that the last element becomes the current element.

Compilation flags:
    static

Template:
    forward(Zipper,NewZipper)
Mode and number of proofs:
    forward(+zipper,--zipper) - one

---

forward/3

Forward the zipper so that the last element becomes the current element. Also returns the last element.

Compilation flags:
    static

Template:
    forward(Zipper,NewZipper,Last)
Mode and number of proofs:
    forward(+zipper,--zipper,?term) - zero_or_one

---

apply/2

Applies a closure to the current element.

Compilation flags:
    static

Template:
    apply(Closure,Zipper)
Meta-predicate template:
    apply(1,*)
Mode and number of proofs:
    apply(+callable,+zipper) - zero_or_more

---

insert_before/3

Inserts an element before the current one.

Compilation flags:
    static

Template:
    insert_before(Zipper,Element,NewZipper)
Mode and number of proofs:
    insert_before(+zipper,?term,--zipper) - zero_or_one

---

insert_after/3

Inserts an element after the current one.

Compilation flags:
    static

Template:
    insert_after(Zipper,Element,NewZipper)
Mode and number of proofs:
    insert_after(+zipper,?term,--zipper) - zero_or_one

---

replace/3

Replaces the current element with a new element.

Compilation flags:
    static

Template:
    replace(Zipper,NewCurrent,NewZipper)
Mode and number of proofs:

replace(+zipper,?term,--zipper) - one

---

delete_and_previous/2

Deletes the current element and moves to the previous element. Fails if no previous element exists.

Compilation flags:
static

Template:
delete_and_previous(Zipper,NewZipper)
Mode and number of proofs:
delete_and_previous(+zipper,--zipper) - zero_or_one

---

delete_and_next/2

Deletes the current element and moves to the next element. Fails if no next element exists.

Compilation flags:
static

Template:
delete_and_next(Zipper,NewZipper)
Mode and number of proofs:
delete_and_next(+zipper,--zipper) - zero_or_one

---

delete_and_unzip/2

Deletes the current element and removes the zipper returning the resulting sequence.

Compilation flags:
static

Template:

delete_and_unzip(Zipper,Sequence)
Mode and number of proofs:
delete_and_unzip(+zipper,--sequence) - one

---

delete_all_before/2

Deletes all elements before the current element.

Compilation flags:
static

Template:
delete_all_before(Zipper,NewZipper)
Mode and number of proofs:
delete_all_before(+zipper,--zipper) - one

---

delete_all_before_and_unzip/2

Deletes all elements before the current element and removes the zipper returning the resulting sequence.

Compilation flags:
static

Template:
delete_all_before_and_unzip(Zipper,NewZipper)
Mode and number of proofs:
delete_all_before_and_unzip(+zipper,--sequence) - one

---

delete_all_after/2

Deletes all elements after the current element.

Compilation flags:
    static

Template:
    delete_all_after(Zipper,NewZipper)
Mode and number of proofs:
    delete_all_after(+zipper,--zipper) - one

---

delete_all_after_and_unzip/2

Deletes all elements after the current element and removes the zipper returning the resulting sequence.

Compilation flags:
    static

Template:
    delete_all_after_and_unzip(Zipper,NewZipper)
Mode and number of proofs:
    delete_all_after_and_unzip(+zipper,--sequence) - one

Protected predicates

(none)

Private predicates

(none)

Operators

(none)

See also

zlist

object

1.241.2 zlist

Zipper list predicates. Zippers should be regarded as opaque terms.

Availability:
logtalk_load(zippers(loader))

Author: Paulo Moura
Version: 1:0:1
Date: 2019-03-12

Compilation flags:
static, context_switching_calls

Implements:
public zipperp

Remarks:
(none)

Inherited public predicates:
apply/2 current/2 delete_all_after/2 delete_all_after_and_unzip/2 delete_all_before/2 delete_all_before_and_unzip/2 delete_and_next/2 delete_and_previous/2 delete_and_unzip/2 forward/2 forward/3 insert_after/3 insert_before/3 next/2 next/3 previous/2 previous/3 replace/3 rewind/2 rewind/3 unzip/2 zip/2 zip/3

- Public predicates
  - zip_at_index/4
- Protected predicates
- Private predicates

- Operators

Public predicates

zip_at_index/4

Adds a zipper to a list opened at the given index and also returns the element at the index. Fails if the list is empty or the index (starting at 1) does not exist.

Compilation flags:
    static

Template:
    zip_at_index(Index,List,Zipper,Element)
Mode and number of proofs:
    zip_at_index(+natural,+list,--zipper,--term) - zero_or_one

---

Protected predicates

(no local declarations; see entity ancestors if any)

Private predicates

(no local declarations; see entity ancestors if any)

Operators

(none)

CHAPTER

TWO

DIRECTORIES

To load an entity, always load the library that includes it using the goal logtalk_load(library_name(loader)) instead of using its path. The library loader file ensures that all the required dependencies are also loaded and that any required flags are used. The loading goal can be found in the entity documentation.

2.1 contributions/flags/

2.2 contributions/iso8601/

2.3 contributions/pddl_parser/

2.4 contributions/verdi_neruda/

2.5 contributions/xml_parser/

2.6 core/

2.7 library/

2.8 library/adaptive_boosting_classifier/

2.9 library/agglomerative_clusterer/

2.10 library/amqp/

2.11 library/anomaly_detection_protocols/

2.12 library/application/

2.13 library/apriori_pattern_miner/

2.14 library/arbitrary/

2.15 library/arrangements/

2.16 library/assignvars/

2.17 library/avro/

2.18 library/base32/

2.19 library/base58/

2.20 library/base64/

2.21 library/base85/

CHAPTER

THREE

ENTITIES

To load an entity, always load the library that includes it using the goal logtalk_load(library_name(loader)) instead of loading just the entity. The library loader file ensures that all the required dependencies are also loaded and that any required flags are used. The loading goal can be found in the entity documentation.

3.1 Categories

3.2 Objects

3.3 Protocols

CHAPTER

FOUR

PREDICATES

This index lists all entities declaring a given predicate. To load an entity providing the predicate that you want to call, always load the library that includes it using the goal logtalk_load(library_name(loader)) instead of loading just the entity. The library loader file ensures that all the required dependencies are also loaded and that any required flags are used. The loading goal can be found in the entity documentation.

4.1 (/)/2

- help

4.2 (//)/2

- help

4.3 (<)/2

- comparingp

4.4 (<=)/2

- assignvarsp
- streamvars

4.5 (=:=)/2

- comparingp

4.6 (=<)/2

- comparingp

4.7 (=>)/2

- assignvarsp
- streamvars

4.8 (=\=)/2

- comparingp

4.9 =~= / 2

- lgtunit
- number

4.10 (>)/2

- comparingp

4.11 (>=)/2

- comparingp

4.12 abbreviation/2

- tzif_protocol

4.13 abbreviation/3

- tzif_protocol

4.14 abbreviation/4

- tzif_protocol

4.15 abort_transaction/3

- stomp

4.16 absolute_file_name/2

- osp

4.17 ack/3

- stomp

4.18 acquire/1

- amqp_pool

4.19 activate_debug_handler/1

- logtalk

4.20 activate_monitor/0

- monitorp

4.21 active_debug_handler/1

- logtalk

4.22 add/1

- registries

4.23 add/2

- registries

4.24 add/3

- difflist
- registries

4.25 add/5

- memcached

4.26 add32/3

- hash_common_32

4.27 add32/4

- hash_common_32

4.28 add32/5

- hash_common_32

4.29 add64/3

- hash_common_64

4.30 addDependent/1

- subject

4.31 add_duration/3

- datep

4.32 add_edge/4

- unweighted_graph_protocol

4.33 add_edge/5

- weighted_graph_protocol

4.34 add_edges/3

- graph_protocol

4.35 add_matrices/3

- linear_algebra_protocol

4.36 add_rule/3

- datalog_protocol

4.37 add_scaled_outer_product/4

- linear_algebra_protocol

4.38 add_scaled_vector/4

- linear_algebra_protocol

4.39 add_type/2

- mime_types

4.40 add_type/3

- mime_types

4.41 add_vectors/3

- linear_algebra_protocol

4.42 add_vertex/3

- graph_protocol

4.43 add_vertices/3

- graph_protocol

4.44 adjacent/3

- geohash_protocol

4.45 adjacent_bits/4

- geohash_protocol

4.46 adjacent_int/4

- geohash_protocol

4.47 after/2

- datep
- interval_protocol

4.48 after/3

- monitoring

4.49 alias/1

- character_set_protocol

4.50 all/0

- code_metric
- dead_code_scanner
- lgtdocp

4.51 all/1

- code_metric
- dead_code_scanner
- lgtdocp

4.52 all_files/0

- diagram(Format)
- diagrams(Format)

4.53 all_files/1

- diagram(Format)
- diagrams(Format)

4.54 all_libraries/0

- diagram(Format)
- diagrams(Format)

4.55 all_libraries/1

- diagram(Format)
- diagrams(Format)

4.56 all_pairs_min_paths/2

- graph_protocol

4.57 all_pairs_min_predecessors/2

- graph_protocol

4.58 all_score/1

- code_metric

4.59 along_track_distance/4

- geospatial_protocol

4.60 alternating_subsequence/2

- subsequences_protocol

4.61 alternating_subsequences/2

- subsequences_protocol

4.62 ancestor/1

- hierarchyp

4.63 ancestors/1

- hierarchyp

4.64 and64/3

- hash_common_64

4.65 anomaly_detector_options/2

- anomaly_detector_protocol

4.66 apid/2

- ccsds_packets(SecondaryHeaderLength)

4.67 apis/0

- help

4.68 apis/1

- help

4.69 append/2

- listp

4.70 append/3

- listp
- memcached
- queuep
- varlistp

4.71 append/4

- redis

4.72 apply/2

- zipperp

4.73 apply/4

- dictionaryp

4.74 approximately_equal/2

- lgtunit
- number

4.75 approximately_equal/3

- lgtunit
- number

4.76 arbitrary/1

- arbitrary

4.77 arbitrary/2

- arbitrary

4.78 archive/1

- registry_protocol

4.79 arithmetic_mean/2

- statisticsp

4.80 arrangement/3

- arrangements_protocol

4.81 arrangement/4

- arrangements_protocol

4.82 arrangement_index/4

- arrangements_protocol

4.83 arrangement_index/5

- arrangements_protocol

4.84 arrangements/3

- arrangements_protocol

4.85 arrangements/4

- arrangements_protocol

4.86 array_list/2

- java_utils_protocol

4.87 array_to_list/2

- java_utils_protocol

4.88 array_to_terms/2

- java_utils_protocol

4.89 array_to_terms/3

- java_utils_protocol

4.90 articulation_points/2

- undirected_graph_common

4.91 as_curly_bracketed/2

- dictionaryp
- nested_dictionary_protocol

4.92 as_deque/2

- deque_protocol

4.93 as_dictionary/2

- dictionaryp

4.94 as_difflist/2

- list

4.95 as_heap/2

- heap_protocol

4.96 as_list/2

- deque_protocol
- dictionaryp
- difflist
- heap_protocol
- queuep
- setp

4.97 as_nested_dictionary/2

- nested_dictionary_protocol

4.98 as_set/2

- setp

4.99 ask_question/5

- logtalk

4.100 assert_fact/1

- datalog_protocol

4.101 assertion/1

- assertions(Mode)
- lgtunit

4.102 assertion/2

- assertions(Mode)
- lgtunit

4.103 assignable/1

- assignvarsp

4.104 assignable/2

- assignvarsp

4.105 atom_string/2

- string(Representation)

4.106 atomics_to_string/2

- string(Representation)

4.107 atomics_to_string/3

- string(Representation)

4.108 attribute_values/2

- anomaly_dataset_protocol
- clustering_dataset_protocol
- dataset_protocol
- dimension_reduction_dataset_protocol
- regression_dataset_protocol

4.109 available/0

- packs

4.110 available/1

- packs

4.111 available/2

- packs

4.112 average/2

- numberlistp

4.113 average_deviation/3

- statisticsp

4.114 backward_substitution/3

- linear_algebra_protocol

4.115 basic_ack/3

- amqp

4.116 basic_cancel/3

- amqp

4.117 basic_consume/3

- amqp

4.118 basic_get/3

- amqp

4.119 basic_nack/3

- amqp

4.120 basic_publish/4

- amqp

4.121 basic_qos/2

- amqp

4.122 basic_recover/2

- amqp

4.123 basic_reject/3

- amqp

4.124 basis_vector/3

- linear_algebra_protocol

4.125 bbox_contains/2

- geospatial_protocol

4.126 bbox_contains_polygon/2

- geospatial_protocol

4.127 bbox_expand/3

- geospatial_protocol

4.128 bbox_from_coordinates/2

- geospatial_protocol

4.129 bbox_intersects/2

- geospatial_protocol

4.130 bbox_intersects_polygon/2

- geospatial_protocol

4.131 bbox_intersects_polyline/2

- geospatial_protocol

4.132 bbox_overlaps/2

- geospatial
- geospatial_protocol

4.133 bbox_union/3

- geospatial_protocol

4.134 before/2

- datep
- interval_protocol

4.135 before/3

- monitoring

4.136 begin/0

- datalog_protocol

4.137 begin_transaction/3

- stomp

4.138 bell/2

- natural

4.139 bench_goal/1

- databasep

4.140 benchmark/2

- lgtunit

4.141 benchmark/3

- lgtunit

4.142 benchmark/4

- lgtunit

4.143 benchmark_reified/3

- lgtunit

4.144 bernoulli/2

- sampling_protocol

4.145 beta/3

- sampling_protocol

4.146 between/3

- integer
- random_protocol

4.147 between/4

- float

4.148 bic/5

- iso_9362_protocol

4.149 big_endian_word32/2

- hash_common_32

4.150 binomial/3

- natural
- sampling_protocol

4.151 bit//1

- number_grammars(Format)

4.152 bits//1

- number_grammars(Format)

4.153 blank//0

- blank_grammars(Format)

4.154 blanks//0

- blank_grammars(Format)

4.155 block_size/1

- hash_digest_protocol

4.156 body_pred/1

- metagol

4.157 bounding_box/2

- geohash_protocol

4.158 bounding_box/3

- geospatial_protocol

4.159 bounding_box_bits/3

- geohash_protocol

4.160 bounding_box_int/3

- geohash_protocol

4.161 branch/2

- git_protocol

4.162 breadth_first_order/3

- graph_protocol

4.163 bridges/2

- undirected_graph_common

4.164 built_date/1

- application_protocol

4.165 built_in_directive/4

- help

4.166 built_in_flag/2

- flags

4.167 built_in_method/4

- help

4.168 built_in_non_terminal/4

- help

4.169 built_in_predicate/4

- help

4.170 bypass_flag/2

- ccsds_tc_frames(FrameLength,SegmentHeaderLength,HasFECF)

4.171 byte_count/1

- ieee_754_fields_protocol
- ieee_754_protocol

4.172 bytes_hex/2

- hash_common_32

4.173 bytes_to_codes/2

- character_set_protocol

4.174 cache/1

- tzif_protocol

4.175 cache_source/1

- tzif_protocol

4.176 cached_tzif/1

- tzif_protocol

4.177 calendar_month/3

- iso8601

4.178 call_with_timeout/2

- timeout

4.179 call_with_timeout/3

- timeout

4.180 canonical_bic/2

- iso_9362_protocol

4.181 canonical_iban/2

- iso_13616_protocol

4.182 capabilities/1

- mcp_tool_protocol

4.183 cartesian_product/2

- cartesian_products_protocol

4.184 cartesian_product/3

- cartesian_products_protocol

4.185 cartesian_tuple/2

- cartesian_products_protocol

4.186 cartesian_tuple/3

- cartesian_products_protocol

4.187 cartesian_tuple_index/3

- cartesian_products_protocol

4.188 cartesian_tuple_index/4

- cartesian_products_protocol

4.189 cas/6

- memcached

4.190 cat/2

- maybe

4.191 catalan/2

- natural

4.192 cell_dimensions/3

- geohash_protocol

4.193 center_gram_matrix/4

- linear_algebra_protocol

4.194 center_gram_vector/4

- linear_algebra_protocol

4.195 central_moment/3

- statisticsp

4.196 change_directory/1

- osp

4.197 changed/0

- subject

4.198 changed/1

- subject

4.199 channel_close/1

- amqp

4.200 channel_close/3

- amqp

4.201 channel_open/3

- amqp

4.202 chebyshev_distance/3

- numberlistp

4.203 chebyshev_norm/2

- numberlistp

4.204 check/0

- command_line_option

4.205 check/1

- termp
- varlistp

4.206 check/2

- type

4.207 check/3

- type

4.208 check_anomaly_detector/1

- anomaly_detector_protocol

4.209 check_classifier/1

- classifier_protocol

4.210 check_clusterer/1

- clusterer_protocol

4.211 check_conversion/3

- time_scales_protocol

4.212 check_convert/4

- time_scales_protocol

4.213 check_dimension_reducer/1

- dimension_reducer_protocol

4.214 check_instant/1

- time_scales_protocol

4.215 check_offset/3

- time_scales_protocol

4.216 check_option/1

- options_protocol

4.217 check_options/1

- options_protocol

4.218 check_pattern_miner/1

- pattern_miner_protocol

4.219 check_ranker/1

- ranker_protocol

4.220 check_regressor/1

- regressor_protocol

4.221 checksum/2

- nmea_protocol

4.222 chi_squared/2

- sampling_protocol

4.223 children/2

- geohash_protocol

4.224 cholesky_decomposition/2

- linear_algebra_protocol

4.225 chr_is/2

- toychrdb

4.226 chr_no_spy/1

- toychrdb

4.227 chr_nospy/0

- toychrdb

4.228 chr_notrace/0

- toychrdb

4.229 chr_option/2

- toychrdb

4.230 chr_spy/1

- toychrdb

4.231 chr_trace/0

- toychrdb

4.232 circular_uniform_cartesian/3

- sampling_protocol

4.233 circular_uniform_polar/3

- sampling_protocol

4.234 class/1

- anomaly_dataset_protocol
- class_hierarchyp
- dataset_protocol
- supervised_dimension_reduction_dataset_protocol

4.235 class_values/1

- anomaly_dataset_protocol
- dataset_protocol
- supervised_dimension_reduction_dataset_protocol

4.236 classes/1

- class_hierarchyp

4.237 classifier_options/2

- classifier_protocol

4.238 classify/2

- ieee_754_fields_protocol

4.239 clause/5

- ports_profiler

4.240 clause_location/6

- ports_profiler

4.241 clean/0

- packs
- registries

4.242 clean/1

- packs
- registries

4.243 clean/2

- packs

4.244 clear/0

- datalog_protocol

4.245 clear_cache/0

- tzif_protocol

4.246 clear_dut1_override/0

- time_scales_data
- time_scales_protocol

4.247 clear_leap_seconds_override/0

- time_scales_data
- time_scales_protocol

4.248 client_open/4

- socket

4.249 client_open/5

- socket

4.250 clockwise_polygon/2

- geospatial_protocol

4.251 clone/1

- cloning
- registry_protocol

4.252 clone/3

- dictionaryp

4.253 clone/4

- dictionaryp

4.254 close/1

- amqp

4.255 close/2

- socket

4.256 close/3

- amqp

4.257 close_client/1

- linda_client

4.258 close_polygon/2

- geospatial_protocol

4.259 cluster/3

- clusterer_protocol

4.260 cluster_probabilities/3

- gaussian_mixture_clusterer

4.261 clusterer_options/2

- clusterer_protocol

4.262 coarse/2

- ccsds_cuc(CoarseOctets,FineOctets,Epoch)

4.263 codes_to_bytes/2

- character_set_protocol

4.264 coefficient_of_variation/2

- statisticsp

4.265 combination/3

- combinations_protocol

4.266 combination/4

- combinations_protocol

4.267 combination_index/4

- combinations_protocol

4.268 combination_index/5

- combinations_protocol

4.269 combinations/3

- combinations_protocol

4.270 combinations/4

- combinations_protocol

4.271 command_line_arguments/1

- osp

4.272 commit/0

- datalog_protocol

4.273 commit_author/2

- git_protocol

4.274 commit_date/2

- git_protocol

4.275 commit_hash/2

- git_protocol

4.276 commit_hash_abbreviated/2

- git_protocol

4.277 commit_log/3

- git_protocol

4.278 commit_message/2

- git_protocol

4.279 commit_transaction/3

- stomp

4.280 common_prefix/3

- geohash_protocol

4.281 common_subsequence/3

- subsequences_protocol

4.282 common_subsequences/3

- subsequences_protocol

4.283 compact/3

- json_ld_protocol

4.284 compare_date_time/3

- datep

4.285 compile_aux_clauses/1

- logtalk

4.286 compile_predicate_heads/4

- logtalk

4.287 compile_predicate_indicators/3

- logtalk

4.288 complement/2

- unweighted_graph_protocol

4.289 completion/2

- help

4.290 completions/2

- help

4.291 compose/3

- interval_algebra_protocol
- interval_relation_set_protocol
- unweighted_directed_graph(Dictionary)

4.292 compress/2

- geohash_protocol

4.293 confirm_select/1

- amqp

4.294 connect/1

- redis

4.295 connect/2

- memcached

4.296 connect/3

- memcached
- redis

4.297 connect/4

- amqp
- stomp

4.298 connected_components/2

- unweighted_undirected_graph(Dictionary)
- weighted_undirected_graph(Dictionary)

4.299 connection_alive/1

- amqp
- stomp

4.300 consensus_ranking/2

- kemeny_young_ranker

4.301 consensus_score/2

- kemeny_young_ranker

4.302 consistent/1

- interval_constraint_network_protocol

4.303 console/1

- redis

4.304 constraints/2

- interval_constraint_network_protocol

4.305 contains/2

- interval_protocol

4.306 contradiction/2

- interval_constraint_network_protocol

4.307 contradiction_explanations/2

- interval_constraint_network_protocol

4.308 control//0

- blank_grammars(Format)

4.309 control_command_flag/2

- ccsds_tc_frames(FrameLength,SegmentHeaderLength,HasFECF)

4.310 control_construct/4

- help

4.311 controls//0

- blank_grammars(Format)

4.312 converse/2

- interval_algebra_protocol
- interval_relation_set_protocol

4.313 convert/4

- time_scales_protocol

4.314 convert_zones/4

- dates_tz_protocol

4.315 convert_zones_with_resolution/5

- dates_tz_protocol

4.316 cooling_schedule/3

- simulated_annealing_protocol

4.317 coordinates_bounding_box/2

- geospatial_protocol

4.318 copy_file/2

- osp

4.319 correlation/3

- statisticsp

4.320 cosine_similarity/3

- string_distance(Representation)

4.321 count_arrangements/3

- arrangements_protocol

4.322 count_cartesian_product/2

- cartesian_products_protocol

4.323 count_combinations/3

- combinations_protocol

4.324 count_derangements/2

- derangements_protocol

4.325 count_distinct_arrangements/3

- arrangements_protocol

4.326 count_distinct_cartesian_product/2

- cartesian_products_protocol

4.327 count_distinct_combinations/3

- combinations_protocol

4.328 count_distinct_derangements/2

- derangements_protocol

4.329 count_distinct_multisets/3

- multisets_protocol

4.330 count_distinct_partitions/2

- partitions_protocol

4.331 count_distinct_partitions/3

- partitions_protocol

4.332 count_distinct_permutations/2

- permutations_protocol

4.333 count_distinct_subsequences/3

- subsequences_protocol

4.334 count_multisets/3

- multisets_protocol

4.335 count_partial_derangements/3

- derangements_protocol

4.336 count_partitions/2

- partitions_protocol

4.337 count_partitions/3

- partitions_protocol

4.338 count_permutations/2

- permutations_protocol

4.339 count_subsequences/2

- subsequences_protocol

4.340 counter/2

- counters
- mutations_store

4.341 counterclockwise_polygon/2

- geospatial_protocol

4.342 country/4

- iso_3166_1
- iso_3166_protocol

4.343 country_spec/3

- iso_13616_registry

4.344 covariance/3

- statisticsp

4.345 covariance_matrix/2

- linear_algebra_protocol

4.346 cover/1

- lgtunit

4.347 coverage_clause_mutator/0

- mutator_protocol

4.348 covering/3

- geohash_protocol

4.349 covering/4

- geohash_protocol

4.350 cpu_time/1

- osp
- timep

4.351 create/3

- process

4.352 creators/1

- application_protocol

4.353 cross_track_distance/4

- geospatial_protocol

4.354 crs_dimensions/2

- crs_projections_protocol

4.355 crs_epsg/2

- crs_projections_protocol

4.356 crs_kind/2

- crs_projections_protocol

4.357 crs_name/2

- crs_projections_protocol

4.358 crs_units/2

- crs_projections_protocol

4.359 currency/5

- iso_4217
- iso_4217_protocol

4.360 current/2

- zipperp

4.361 current_host/1

- socket

4.362 cut/3

- hierarchical_clustering

4.363 cycle/2

- directed_graph_protocol
- undirected_graph_common

4.364 damerau_levenshtein/3

- string_distance(Representation)

4.365 data/0

- ports_profiler

4.366 data/1

- ports_profiler

4.367 data/2

- nmea_protocol
- ports_profiler

4.368 data_field/2

- ccsds_frame_protocol
- ccsds_frames

4.369 data_length/2

- ccsds_packets(SecondaryHeaderLength)

4.370 date/4

- iso8601

4.371 date/5

- iso8601

4.372 date/6

- iso8601

4.373 date/7

- iso8601

4.374 date_string/3

- iso8601

4.375 date_time/7

- osp

4.376 date_time_string/3

- iso8601

4.377 date_time_to_unix/2

- datep

4.378 day/2

- ccsds_ccs(CalendarVariant,FractionOctets)

4.379 day_of_year/2

- ccsds_ccs(CalendarVariant,FractionOctets)
- datep

4.380 day_of_year_date/3

- datep

4.381 daylight_saving_time/2

- tzif_protocol

4.382 daylight_saving_time/3

- tzif_protocol

4.383 daylight_saving_time/4

- tzif_protocol

4.384 days/2

- ccsds_cds(DaySegmentOctets,SubmillisecondOctets,Epoch)

4.385 days_in_month/3

- datep

4.386 deactivate_debug_handler/0

- logtalk

4.387 debug/0

- debuggerp

4.388 debug_handler/1

- logtalk

4.389 debug_handler/3

- logtalk

4.390 debugging/0

- debuggerp

4.391 debugging/1

- debuggerp

4.392 decide/1

- fcube

4.393 decide/2

- fcube

4.394 decode/2

- geohash_protocol
- json_rpc

4.395 decode_bits/3

- geohash_protocol

4.396 decode_exception/2

- java_utils_protocol

4.397 decode_exception/3

- java_utils_protocol

4.398 decode_frame/2

- amqp

4.399 decode_int/3

- geohash_protocol

4.400 decompile_predicate_heads/4

- logtalk

4.401 decompile_predicate_indicators/4

- logtalk

4.402 decompose_file_name/3

- osp

4.403 decompose_file_name/4

- osp

4.404 decr/3

- redis

4.405 decr/4

- memcached

4.406 decrby/4

- redis

4.407 decrement_counter/1

- counters

4.408 default/1

- command_line_option

4.409 default_option/1

- options_protocol
- wrapper

4.410 default_options/1

- options_protocol
- wrapper

4.411 define_log_file/2

- loggingp

4.412 defined/4

- registries

4.413 defined_flag/6

- flags

4.414 degree/3

- unweighted_undirected_graph(Dictionary)
- weighted_undirected_graph(Dictionary)

4.415 del/3

- redis

4.416 del_monitors/0

- event_registryp

4.417 del_monitors/4

- event_registryp

4.418 del_spy_points/4

- monitorp

4.419 delete/0

- registries

4.420 delete/1

- registries

4.421 delete/2

- memcached
- registries

4.422 delete/3

- listp
- setp
- varlistp

4.423 delete/4

- dictionaryp
- heap_protocol

4.424 delete_all_after/2

- zipperp

4.425 delete_all_after_and_unzip/2

- zipperp

4.426 delete_all_before/2

- zipperp

4.427 delete_all_before_and_unzip/2

- zipperp

4.428 delete_and_next/2

- zipperp

4.429 delete_and_previous/2

- zipperp

4.430 delete_and_unzip/2

- zipperp

4.431 delete_directory/1

- osp

4.432 delete_directory_and_contents/1

- osp

4.433 delete_directory_contents/1

- osp

4.434 delete_edge/4

- unweighted_graph_protocol

4.435 delete_edge/5

- weighted_graph_protocol

4.436 delete_edges/3

- graph_protocol

4.437 delete_file/1

- osp

4.438 delete_in/4

- nested_dictionary_protocol

4.439 delete_matches/3

- listp

4.440 delete_max/4

- dictionaryp

4.441 delete_min/4

- dictionaryp

4.442 delete_vertex/3

- graph_protocol

4.443 delete_vertices/3

- graph_protocol

4.444 dependents/1

- packs
- subject

4.445 dependents/2

- packs

4.446 dependents/3

- packs

4.447 depth/2

- termp

4.448 depth_first_order/3

- graph_protocol

4.449 derangement/2

- derangements_protocol

4.450 derangement/3

- derangements_protocol

4.451 derangement_index/3

- derangements_protocol

4.452 derangement_index/4

- derangements_protocol

4.453 derangements/2

- derangements_protocol

4.454 derangements/3

- derangements_protocol

4.455 descendant/1

- hierarchyp

4.456 descendant_class/1

- class_hierarchyp

4.457 descendant_classes/1

- class_hierarchyp

4.458 descendant_instance/1

- class_hierarchyp

4.459 descendant_instances/1

- class_hierarchyp

4.460 descendants/1

- hierarchyp

4.461 describe/1

- packs
- registries

4.462 describe/2

- packs

4.463 description/1

- application_protocol
- pack_protocol
- registry_protocol

4.464 destination_point/4

- geospatial_protocol

4.465 destroy/0

- amqp_pool

4.466 determinant/2

- linear_algebra_protocol

4.467 deterministic/1

- lgtunit

4.468 deterministic/2

- lgtunit

4.469 diagnostic/2

- anomaly_detector_protocol
- classifier_protocol
- clusterer_protocol
- dimension_reducer_protocol
- pattern_miner_protocol

- ranker_protocol
- regressor_protocol
- tool_diagnostics_protocol

4.470 diagnostic/3

- tool_diagnostics_protocol

4.471 diagnostic_rule/5

- tool_diagnostics_protocol

4.472 diagnostic_rules/1

- tool_diagnostics_protocol

4.473 diagnostic_target/1

- tool_diagnostics_protocol

4.474 diagnostics/2

- anomaly_detector_protocol
- classifier_protocol
- clusterer_protocol
- dimension_reducer_protocol
- pattern_miner_protocol
- ranker_protocol
- regressor_protocol
- tool_diagnostics_protocol

4.475 diagnostics/3

- tool_diagnostics_protocol

4.476 diagnostics_preflight/2

- tool_diagnostics_protocol

4.477 diagnostics_preflight/3

- tool_diagnostics_protocol

4.478 diagnostics_summary/2

- tool_diagnostics_protocol

4.479 diagnostics_summary/3

- tool_diagnostics_protocol

4.480 diagnostics_tool/5

- tool_diagnostics_protocol

4.481 diagonal_matrix/2

- linear_algebra_protocol

4.482 diagonal_matrix/3

- linear_algebra_protocol

4.483 diagonal_matrix/4

- linear_algebra_protocol

4.484 diagram_description/1

- diagram(Format)

4.485 diagram_name_suffix/1

- diagram(Format)

4.486 dif/1

- coroutining
- dif

4.487 dif/2

- coroutining
- dif

4.488 difference_norm/3

- linear_algebra_protocol

4.489 digest/2

- hash_digest_protocol

4.490 digest/4

- hmac_protocol

4.491 digest/5

- hmac_protocol

4.492 digest_size/1

- hash_digest_protocol

4.493 digit//1

- number_grammars(Format)

4.494 digits//1

- number_grammars(Format)

4.495 dimension_reducer_options/2

- dimension_reducer_protocol

4.496 directories/1

- lgtdocp
- wrapper

4.497 directories/2

- diagram(Format)
- diagrams(Format)
- lgtdocp
- wrapper

4.498 directories/3

- diagram(Format)
- diagrams(Format)

4.499 directory/1

- code_metric
- dead_code_scanner
- diagram(Format)
- diagrams(Format)
- lgtdocp
- mutation_testing
- packs_common
- wrapper

4.500 directory/2

- code_metric
- dead_code_scanner
- diagram(Format)
- diagrams(Format)
- lgtdocp
- mutation_testing
- packs_common
- wrapper

4.501 directory/3

- diagram(Format)
- diagrams(Format)

4.502 directory_exists/1

- osp

4.503 directory_files/2

- osp

4.504 directory_files/3

- osp

4.505 directory_mutants/2

- mutation_testing

4.506 directory_mutants/3

- mutation_testing

4.507 directory_score/2

- code_metric

4.508 dirichlet/2

- sampling_protocol

4.509 disable/0

- linter_reporter

4.510 disable/1

- debug_messages

4.511 disable/2

- debug_messages

4.512 disable_logging/1

- loggingp

4.513 disconnect/1

- memcached
- redis

4.514 disconnect/2

- stomp

4.515 disjoint/2

- setp

4.516 disjoint_sets/2

- union_find_protocol

4.517 dispatch_map_id/2

- ccsds_tc_services

4.518 dispatch_service_units/2

- ccsds_tc_services

4.519 dispatch_service_units_by_map/2

- ccsds_tc_services

4.520 dispatch_service_units_by_map/3

- ccsds_tc_services

4.521 distance/4

- geospatial
- geospatial_protocol

4.522 distance/5

- geospatial
- geospatial_protocol

4.523 distinct_arrangement/3

- arrangements_protocol

4.524 distinct_arrangement/4

- arrangements_protocol

4.525 distinct_arrangement_index/4

- arrangements_protocol

4.526 distinct_arrangements/3

- arrangements_protocol

4.527 distinct_arrangements/4

- arrangements_protocol

4.528 distinct_cartesian_product/2

- cartesian_products_protocol

4.529 distinct_cartesian_product/3

- cartesian_products_protocol

4.530 distinct_cartesian_tuple/2

- cartesian_products_protocol

4.531 distinct_cartesian_tuple/3

- cartesian_products_protocol

4.532 distinct_cartesian_tuple_index/3

- cartesian_products_protocol

4.533 distinct_combination/3

- combinations_protocol

4.534 distinct_combination/4

- combinations_protocol

4.535 distinct_combination_index/4

- combinations_protocol

4.536 distinct_combinations/3

- combinations_protocol

4.537 distinct_combinations/4

- combinations_protocol

4.538 distinct_derangement/2

- derangements_protocol

4.539 distinct_derangement/3

- derangements_protocol

4.540 distinct_derangement_index/3

- derangements_protocol

4.541 distinct_derangements/2

- derangements_protocol

4.542 distinct_derangements/3

- derangements_protocol

4.543 distinct_multiset/3

- multisets_protocol

4.544 distinct_multiset/4

- multisets_protocol

4.545 distinct_multiset_index/4

- multisets_protocol

4.546 distinct_multisets/3

- multisets_protocol

4.547 distinct_multisets/4

- multisets_protocol

4.548 distinct_partition/2

- partitions_protocol

4.549 distinct_partition/3

- partitions_protocol

4.550 distinct_partition/4

- partitions_protocol

4.551 distinct_partition_index/3

- partitions_protocol

4.552 distinct_partition_index/4

- partitions_protocol

4.553 distinct_partition_index/5

- partitions_protocol

4.554 distinct_partitions/2

- partitions_protocol

4.555 distinct_partitions/3

- partitions_protocol

4.556 distinct_partitions/4

- partitions_protocol

4.557 distinct_permutation/2

- permutations_protocol

4.558 distinct_permutation/3

- permutations_protocol

4.559 distinct_permutation_index/3

- permutations_protocol

4.560 distinct_permutations/2

- permutations_protocol

4.561 distinct_permutations/3

- permutations_protocol

4.562 distribution/1

- application_protocol

4.563 doc_goal/1

- doclet

4.564 document/1

- sbom

4.565 document/2

- sbom

4.566 dot//1

- number_grammars(Format)

4.567 dot_product/3

- linear_algebra_protocol

4.568 double_metaphone/3

- string_distance(Representation)

4.569 double_metaphone_match/2

- string_distance(Representation)

4.570 dowhile/2

- loopp

4.571 drop/3

- listp

4.572 duration_between/3

- datep

4.573 duration_string/2

- iso8601

4.574 during/2

- interval_protocol

4.575 dut1_entries/1

- time_scales_data
- time_scales_protocol

4.576 dut1_offset_at_utc_unix/3

- time_scales_data

4.577 dut1_source/1

- time_scales_data
- time_scales_protocol

4.578 easter_day/3

- iso8601

4.579 edge/3

- unweighted_graph_protocol

4.580 edge/4

- weighted_graph_protocol

4.581 edge/6

- graph_language_protocol

4.582 edge_case/2

- arbitrary

4.583 edges/2

- graph_protocol

4.584 edit_similarity/3

- string_distance(Representation)

4.585 edit_similarity/4

- string_distance(Representation)

4.586 either/3

- expected(Expected)

4.587 elicit_request/5

- mcp_server

4.588 empty/1

- deque_protocol
- dictionaryp
- graph_protocol
- heap_protocol
- interval_relation_set_protocol
- listp
- nested_dictionary_protocol
- optional
- queuep
- setp
- varlistp

4.589 enable/0

- linter_reporter

4.590 enable/1

- debug_messages
- linter_reporter

4.591 enable/2

- debug_messages

4.592 enable_logging/1

- loggingp

4.593 enabled/1

- debug_messages

4.594 enabled/2

- debug_messages

4.595 encode/2

- json_rpc

4.596 encode/3

- geohash_protocol

4.597 encode_bits/3

- geohash_protocol

4.598 encode_frame/2

- amqp

4.599 encode_int/3

- geohash_protocol

4.600 encoding_suffix/2

- mime_types

4.601 ensure_directory/1

- osp

4.602 ensure_file/1

- osp

4.603 entailment_explanations/5

- interval_constraint_network_protocol

4.604 entails/4

- interval_constraint_network_protocol

4.605 entails/5

- interval_constraint_network_protocol

4.606 entity/1

- code_metric
- dead_code_scanner
- help
- mutation_testing
- xref_diagram(Format)

4.607 entity/2

- dead_code_scanner
- mutation_testing
- xref_diagram(Format)

4.608 entity_info_pair_score_hook/3

- doc_metric

4.609 entity_info_score_hook/2

- doc_metric

4.610 entity_mutants/2

- mutation_testing

4.611 entity_mutants/3

- mutation_testing

4.612 entity_predicates_weights_hook/2

- doc_metric

4.613 entity_prefix/2

- logtalk

4.614 entity_score/2

- code_metric

4.615 enumerate/2

- random_protocol

4.616 environment_variable/2

- osp

4.617 epoch/1

- ccsds_time_code_protocol

4.618 epoch/2

- ccsds_time_fields

4.619 epsg_crs/2

- crs_projections_protocol

4.620 epsilon/1

- lgtunit

4.621 equal/2

- interval_protocol
- setp

4.622 equirectangular_inverse/4

- geospatial_protocol

4.623 equirectangular_projection/4

- geospatial_protocol

4.624 equivalent/2

- interval_constraint_network_protocol

4.625 erase/1

- recorded_database_core

4.626 error/2

- json_rpc

4.627 error_code/2

- json_rpc

4.628 error_data/2

- json_rpc

4.629 error_message/2

- json_rpc

4.630 error_response/4

- json_rpc

4.631 error_response/5

- json_rpc

4.632 essentially_equal/3

- lgtunit
- number

4.633 estimate_temperature/1

- simulated_annealing(Problem,RandomAlgorithm)

4.634 estimate_temperature/2

- simulated_annealing(Problem,RandomAlgorithm)

4.635 euclidean_distance/3

- numberlistp

4.636 euclidean_norm/2

- linear_algebra_protocol
- numberlistp

4.637 evaluate/3

- json_pointer_protocol

4.638 exactly_representable/1

- ieee_754_protocol

4.639 example/2

- clustering_dataset_protocol
- dimension_reduction_dataset_protocol

4.640 example/3

- anomaly_dataset_protocol
- dataset_protocol
- regression_dataset_protocol
- supervised_dimension_reduction_dataset_protocol
- target_supervised_dimension_reduction_dataset_protocol

4.641 exchange_bind/4

- amqp

4.642 exchange_declare/3

- amqp

4.643 exchange_delete/3

- amqp

4.644 exchange_unbind/4

- amqp

4.645 exclude/3

- metap

4.646 excluded/4

- interval_constraint_network_protocol

4.647 execution_context/7

- logtalk

4.648 exists/3

- redis

4.649 expand/2

- geohash_protocol
- json_ld_protocol

4.650 expand_int/3

- geohash_protocol

4.651 expand_library_path/2

- logtalk

4.652 expected/1

- expected(Expected)

4.653 expecteds/2

- either

4.654 expire/4

- redis

4.655 explain/2

- datalog_protocol

4.656 explain//1

- tutor_explanations

4.657 exponential/2

- sampling_protocol

4.658 export/1

- sbom

4.659 export/2

- sbom

4.660 export_to_clauses/4

- anomaly_detector_protocol
- classifier_protocol
- clusterer_protocol
- dimension_reducer_protocol
- pattern_miner_protocol
- ranker_protocol
- regressor_protocol

4.661 export_to_file/4

- anomaly_detector_protocol
- classifier_protocol
- clusterer_protocol
- dimension_reducer_protocol
- pattern_miner_protocol
- ranker_protocol
- regressor_protocol

4.662 extension/1

- proto_hierarchyp

4.663 extension_type/2

- mime_types

4.664 extension_type/3

- mime_types

4.665 extensions/1

- proto_hierarchyp

4.666 external_reference/2

- application_protocol

4.667 extract_aos_packets/3

- ccsds_packet_services

4.668 extract_packets/3

- ccsds_frames

4.669 extract_packets/4

- ccsds_link_profiles

4.670 extract_tc_segment/2

- ccsds_tc_services

4.671 extract_tm_packets/3

- ccsds_packet_services

4.672 factorial/2

- natural

4.673 facts/1

- datalog_protocol

4.674 failed_test_reason//1

- lgtunit_messages

4.675 falling_factorial/3

- natural

4.676 false/1

- java_utils_protocol

4.677 fcube/0

- fcube

4.678 fecf/2

- ccsds_frame_protocol
- ccsds_frames

4.679 fibonacci/2

- natural

4.680 fields/2

- nmea_protocol

4.681 fields/5

- ieee_754_fields_protocol

4.682 file/1

- code_metric
- dead_code_scanner
- entity_diagram(Format)
- lgtdocp
- wrapper

4.683 file/2

- code_metric
- dead_code_scanner
- entity_diagram(Format)
- lgtdocp
- wrapper

4.684 file_exists/1

- osp

4.685 file_footer/3

- graph_language_protocol

4.686 file_header/3

- graph_language_protocol

4.687 file_modification_time/2

- osp

4.688 file_path_components/2

- url(Representation)

4.689 file_permission/2

- osp

4.690 file_score/2

- code_metric

4.691 file_size/2

- osp

4.692 file_to_bytes/2

- reader

4.693 file_to_bytes/3

- reader

4.694 file_to_chars/2

- reader

4.695 file_to_chars/3

- reader

4.696 file_to_codes/2

- reader

4.697 file_to_codes/3

- reader

4.698 file_to_terms/2

- reader

4.699 file_to_terms/3

- reader

4.700 file_type_extension/2

- logtalk

4.701 files/1

- diagram(Format)
- diagrams(Format)
- lgtdocp
- wrapper

4.702 files/2

- diagram(Format)
- diagrams(Format)
- lgtdocp
- wrapper

4.703 files/3

- diagram(Format)
- diagrams(Format)

4.704 filter/2

- optional(Optional)

4.705 filter/3

- expected(Expected)
- validation(Validation)

4.706 final_bearing/3

- geospatial_protocol

4.707 find/4

- union_find_protocol

4.708 find/5

- union_find_protocol

4.709 findall_in_noblock/3

- linda_client

4.710 findall_in_noblock/4

- linda_client

4.711 findall_member/4

- metap

4.712 findall_member/5

- metap

4.713 findall_rd_noblock/3

- linda_client

4.714 findall_rd_noblock/4

- linda_client

4.715 fine/2

- ccsds_cuc(CoarseOctets,FineOctets,Epoch)

4.716 finished_by/2

- interval_protocol

4.717 finishes/2

- interval_protocol

4.718 finite_binary_rational/4

- ieee_754_fields_protocol

4.719 first_header_pointer/2

- ccsds_tm_frames(FrameLength,SecondaryHeaderLength,HasFECF)

4.720 first_significant_component/2

- linear_algebra_protocol

4.721 first_significant_component/3

- linear_algebra_protocol

4.722 fisher/3

- sampling_protocol

4.723 flag_group_chk/1

- flags

4.724 flag_groups/1

- flags

4.725 flat_map/2

- expected(Expected)
- optional(Optional)
- validation(Validation)

4.726 flatten/1

- expected(Expected)
- optional(Optional)
- validation(Validation)

4.727 flatten/2

- json_ld_protocol
- listp
- varlistp

4.728 float//1

- number_grammars(Format)

4.729 flush_all/1

- memcached

4.730 flush_all/2

- memcached

4.731 fold_left/4

- metap

4.732 fold_left_1/3

- metap

4.733 fold_right/4

- metap

4.734 fold_right_1/3

- metap

4.735 fordownto/3

- loopp

4.736 fordownto/4

- loopp

4.737 fordownto/5

- loopp

4.738 foreach/3

- loopp

4.739 foreach/4

- loopp

4.740 format/1

- ccsds_time_code_protocol

4.741 format/2

- ccsds_time_codes
- ccsds_time_fields
- format

4.742 format/3

- format

4.743 format_date_time/4

- datep

4.744 format_entity_score//2

- code_metric

4.745 format_object/1

- diagram(Format)

4.746 format_report/1

- mutation_testing

4.747 format_report/2

- mutation_testing

4.748 format_report/3

- mutation_testing

4.749 format_to_atom/3

- term_io_protocol

4.750 format_to_chars/3

- term_io_protocol

4.751 format_to_chars/4

- term_io_protocol

4.752 format_to_codes/3

- term_io_protocol

4.753 format_to_codes/4

- term_io_protocol

4.754 formatted_iban/2

- iso_13616_protocol

4.755 forto/3

- loopp

4.756 forto/4

- loopp

4.757 forto/5

- loopp

4.758 forward/1

- forwarding

4.759 forward/2

- zipperp

4.760 forward/3

- zipperp

4.761 forward_substitution/3

- linear_algebra_protocol

4.762 fractile/3

- statisticsp

4.763 frame_body/2

- stomp

4.764 frame_command/2

- stomp

4.765 frame_header/3

- stomp

4.766 frame_headers/2

- stomp

4.767 frame_type/2

- ccsds_frame_protocol
- ccsds_frames

4.768 freeze/2

- coroutining

4.769 frequency_distribution/2

- statisticsp

4.770 frobenius_norm/2

- linear_algebra_protocol

4.771 from_expected/2

- validation

4.772 from_generator/2

- expected
- optional
- validation

4.773 from_generator/3

- expected
- optional
- validation

4.774 from_generator/4

- expected
- validation

4.775 from_goal/2

- expected
- optional
- validation

4.776 from_goal/3

- expected
- optional
- validation

4.777 from_goal/4

- expected
- validation

4.778 from_goal_or_throw/2

- optional

4.779 from_goal_or_throw/3

- optional

4.780 from_optional/3

- expected
- validation

4.781 from_unix_seconds/2

- ccsds_time_code_protocol

4.782 frozen/2

- coroutining

4.783 full_device_path/1

- osp

4.784 func_test/3

- metagol

4.785 functional/0

- metagol

4.786 fund_currency/5

- iso_4217
- iso_4217_protocol

4.787 game/4

- temporal_pairwise_ranking_dataset_protocol

4.788 gamma/3

- sampling_protocol

4.789 gat/4

- memcached

4.790 gats/5

- memcached

4.791 generate/1

- cuid2_protocol
- ids(Representation,Bytes)
- ksuid_protocol
- nanoid_protocol
- snowflakeid_protocol
- ulid_protocol

4.792 generate/2

- base32
- base58
- base64
- base64url
- base85

- cbor(StringRepresentation)
- ccsds_frame_protocol
- ccsds_packets(SecondaryHeaderLength)
- ccsds_time_code_protocol
- geojson_protocol
- html
- ieee_754_protocol
- json_ld_protocol
- json_lines_protocol
- json_pointer_protocol
- json_protocol
- message_pack(StringRepresentation)
- sarif
- toml_protocol
- toon_protocol
- ulid_protocol
- url(Representation)
- wkt_wkb_protocol
- yaml_protocol

4.793 generate/3

- avro
- ccsds_frame_protocol
- ccsds_packets(SecondaryHeaderLength)
- ccsds_time_code_protocol
- ccsds_time_fields
- ieee_754_protocol
- protobuf

4.794 generate/4

- avro
- protobuf
- sarif

4.795 generate/8

- ulid_protocol

4.796 generate_all/2

- yaml_protocol

4.797 generate_fragment/2

- json_pointer_protocol

4.798 generate_frame/3

- ccsds_link_profiles

4.799 generate_frames/3

- ccsds_link_profiles

4.800 generate_idle_packet/4

- ccsds_packetization

4.801 genint/2

- genint_core

4.802 gensym/2

- gensym_core

4.803 geohash_to_int/2

- geohash_protocol

4.804 geojson_to_json/2

- geojson_protocol

4.805 geometric/2

- sampling_protocol

4.806 geometric_mean/2

- statisticsp

4.807 get/1

- optional(Optional)

4.808 get/3

- memcached
- redis

4.809 get/4

- memcached

4.810 get_field/2

- java_access_protocol

4.811 get_flag_value/2

- flags

4.812 get_seed/1

- arbitrary
- pseudo_random_protocol

4.813 getrange/5

- redis

4.814 gets/4

- memcached

4.815 gets/5

- memcached

4.816 git_object_identifier/1

- application_protocol

4.817 gnu/0

- fcube

4.818 goal_expansion/2

- expanding

4.819 gram_matrix/2

- linear_algebra_protocol

4.820 graph_coloring/3

- undirected_graph_common

4.821 graph_footer/5

- graph_language_protocol

4.822 graph_header/5

- graph_language_protocol

4.823 ground/1

- termp

4.824 ground_track/5

- tle_orbits_protocol

4.825 ground_track/6

- tle_orbits_protocol

4.826 group/1

- ranking_dataset_protocol

4.827 group_by_key/2

- pairs

4.828 group_consecutive_by_key/2

- pairs

4.829 group_sorted_by_key/2

- pairs

4.830 grouped_dataset_groups/2

- ranking_dataset_common

4.831 grouped_dataset_items/2

- ranking_dataset_common

4.832 grouped_dataset_items/3

- ranking_dataset_common

4.833 grouped_dataset_summary/2

- ranking_dataset_common

4.834 guess_all_extensions/2

- mime_types

4.835 guess_all_extensions/3

- mime_types

4.836 guess_arity/2

- csv_protocol

4.837 guess_extension/2

- mime_types

4.838 guess_extension/3

- mime_types

4.839 guess_file_type/3

- mime_types

4.840 guess_file_type/4

- mime_types

4.841 guess_separator/2

- csv_protocol

4.842 guess_type/3

- mime_types

4.843 guess_type/4

- mime_types

4.844 gumbel/3

- sampling_protocol

4.845 halt/0

- logtalk

4.846 halt/1

- logtalk

4.847 halt_hook/2

- logtalk

4.848 hamming/3

- string_distance(Representation)

4.849 hamming_distance/3

- listp

4.850 handbook/0

- help

4.851 harmonic_mean/2

- statisticsp

4.852 has_cycle/1

- directed_graph_protocol
- undirected_graph_common

4.853 has_negative_cycle/1

- weighted_graph_protocol

4.854 has_path/3

- graph_protocol

4.855 hash/2

- hash_protocol

4.856 haversine_distance/3

- geospatial_protocol

4.857 hdel/4

- redis

4.858 head/2

- queuep

4.859 head_pred/1

- metagol

4.860 help/0

- help
- packs_common

4.861 help/1

- command_line_option

4.862 help/2

- command_line_options

4.863 help/3

- command_line_options

4.864 hex_digest/4

- hmac_protocol

4.865 hex_digest/5

- hmac_protocol

4.866 hex_digit//1

- number_grammars(Format)

4.867 hex_digits//1

- number_grammars(Format)

4.868 hexists/4

- redis

4.869 hget/4

- redis

4.870 hgetall/3

- redis

4.871 hkeys/3

- redis

4.872 hlen/3

- redis

4.873 home/1

- pack_protocol
- registry_protocol

4.874 homepage/1

- application_protocol

4.875 hset/5

- redis

4.876 hvals/3

- redis

4.877 hypergeometric/4

- sampling_protocol

4.878 iban/4

- iso_13616_protocol

4.879 ibk/3

- metagol

4.880 id/2

- json_rpc

4.881 identity_matrix/2

- linear_algebra_protocol

4.882 if_empty/1

- optional(Optional)

4.883 if_expected/1

- expected(Expected)

4.884 if_expected_or_else/2

- expected(Expected)

4.885 if_invalid/1

- validation(Validation)

4.886 if_present/1

- optional(Optional)

4.887 if_present_or_else/2

- optional(Optional)

4.888 if_unexpected/1

- expected(Expected)

4.889 if_valid/1

- validation(Validation)

4.890 if_valid_or_else/2

- validation(Validation)

4.891 in/1

- linda_client

4.892 in/2

- linda_client

4.893 in_degree/3

- directed_graph_protocol

4.894 in_list/2

- linda_client

4.895 in_list/3

- linda_client

4.896 in_noblock/1

- linda_client

4.897 in_noblock/2

- linda_client

4.898 include/3

- metap

4.899 incr/3

- redis

4.900 incr/4

- memcached

4.901 incrby/4

- redis

4.902 increase/1

- counter

4.903 increment/0

- counter

4.904 increment_counter/1

- counters

4.905 init/0

- shell(Interpreters)

4.906 init/2

- subsequences_protocol

4.907 init1/2

- subsequences_protocol

4.908 init_log_file/2

- loggingp

4.909 init_tail/2

- subsequences_protocol

4.910 init_tails/2

- subsequences_protocol

4.911 initial_bearing/3

- geospatial_protocol

4.912 initial_channel_reassembly_state/1

- ccsds_packet_services

4.913 initial_reassembly_state/1

- ccsds_link_profiles
- ccsds_packet_services
- ccsds_tc_services

4.914 initial_state/1

- ccsds_packetization
- simulated_annealing_protocol

4.915 initial_temperature/1

- simulated_annealing_protocol

4.916 initialize/1

- amqp_pool

4.917 inits/2

- subsequences_protocol

4.918 inits1/2

- subsequences_protocol

4.919 inorder/2

- bintree

4.920 insert/3

- setp

4.921 insert/4

- dictionaryp
- heap_protocol

4.922 insert_after/3

- zipperp

4.923 insert_all/3

- heap_protocol
- setp

4.924 insert_aos_packets/4

- ccsds_packet_services

4.925 insert_before/3

- zipperp

4.926 insert_in/4

- nested_dictionary_protocol

4.927 insert_packets/4

- ccsds_frames

4.928 insert_packets/5

- ccsds_link_profiles

4.929 insert_tc_segment/3

- ccsds_tc_services

4.930 insert_tm_packets/4

- ccsds_packet_services

4.931 insert_zone/2

- ccsds_aos_frames(FrameLength,InsertZoneLength,HasOCF,HasFECF)

4.932 install/1

- packs

4.933 install/2

- packs

4.934 install/3

- packs

4.935 install/4

- packs

4.936 installed/0

- packs

4.937 installed/1

- packs

4.938 installed/3

- packs

4.939 installed/4

- packs

4.940 instance/1

- class_hierarchyp

4.941 instance/2

- recorded_database_core

4.942 instances/1

- class_hierarchyp

4.943 instant_to_utc_date_time/2

- time_scales_protocol

4.944 int_to_geohash/3

- geohash_protocol

4.945 integer//1

- number_grammars(Format)

4.946 integer_to_big_endian_bytes32/2

- hash_common_32

4.947 integer_to_big_endian_bytes64/2

- hash_common_64

4.948 integer_to_little_endian_bytes32/2

- hash_common_32

4.949 internal_error/2

- json_rpc

4.950 internal_os_path/2

- osp

4.951 interpolate_great_circle/4

- geospatial_protocol

4.952 interpolate_rhumb/4

- geospatial_protocol

4.953 interquartile_range/2

- statisticsp

4.954 intersect/2

- setp

4.955 intersection/2

- dictionaryp

4.956 intersection/3

- dictionaryp
- interval_relation_set_protocol
- setp

4.957 intersection/4

- setp

4.958 interval_string/2

- iso8601

4.959 invalid/1

- validation(Validation)

4.960 invalid_params/2

- json_rpc

4.961 invalid_request/1

- json_rpc

4.962 invalids/2

- validated

4.963 inverse_matrix/2

- linear_algebra_protocol

4.964 invert_from_cholesky/2

- linear_algebra_protocol

4.965 invoke/1

- java_access_protocol

4.966 invoke/2

- java_access_protocol

4.967 ipv4//1

- ip_grammars(Format)

4.968 ipv6//1

- ip_grammars(Format)

4.969 is_absolute_file_name/1

- osp

4.970 is_acyclic/1

- directed_graph_protocol

4.971 is_alpha/1

- characterp

4.972 is_alphanumeric/1

- characterp

4.973 is_ascii/1

- characterp

4.974 is_batch/1

- json_rpc

4.975 is_bin_digit/1

- characterp

4.976 is_bipartite/1

- graph_protocol

4.977 is_clockwise_polygon/1

- geospatial_protocol

4.978 is_complete/1

- graph_protocol

4.979 is_connected/1

- unweighted_undirected_graph(Dictionary)
- weighted_undirected_graph(Dictionary)

4.980 is_control/1

- characterp

4.981 is_dec_digit/1

- characterp

4.982 is_empty/0

- optional(Optional)

4.983 is_end_of_line/1

- characterp

4.984 is_error_response/1

- json_rpc

4.985 is_expected/0

- expected(Expected)

4.986 is_false/1

- java_utils_protocol

4.987 is_hex_digit/1

- characterp

4.988 is_invalid/0

- validation(Validation)

4.989 is_layout/1

- characterp

4.990 is_letter/1

- characterp

4.991 is_lower_case/1

- characterp

4.992 is_newline/1

- characterp

4.993 is_notification/1

- json_rpc

4.994 is_null/1

- java_utils_protocol

4.995 is_object/1

- java_utils_protocol

4.996 is_octal_digit/1

- characterp

4.997 is_period/1

- characterp

4.998 is_prefix_of/2

- subsequences_protocol

4.999 is_present/0

- optional(Optional)

4.1000 is_punctuation/1

- characterp

4.1001 is_quote/1

- characterp

4.1002 is_request/1

- json_rpc

4.1003 is_response/1

- json_rpc

4.1004 is_sparse/1

- graph_protocol

4.1005 is_subsequence_of/2

- subsequences_protocol

4.1006 is_suffix_of/2

- subsequences_protocol

4.1007 is_tree/1

- undirected_graph_common

4.1008 is_true/1

- java_utils_protocol

4.1009 is_unexpected/0

- expected(Expected)

4.1010 is_upper_case/1

- characterp

4.1011 is_valid/0

- validation(Validation)

4.1012 is_valid_polygon/1

- geospatial_protocol

4.1013 is_void/1

- java_utils_protocol

4.1014 is_vowel/1

- characterp

4.1015 is_white_space/1

- characterp

4.1016 item/1

- pairwise_measurement_dataset_protocol
- pairwise_ranking_dataset_protocol
- temporal_pairwise_ranking_dataset_protocol

4.1017 item/2

- ranking_dataset_protocol

4.1018 items/1

- sequence_dataset_protocol
- transaction_dataset_protocol

4.1019 iterator_element/2

- java_utils_protocol

4.1020 jaccard_index/3

- string_distance(Representation)

4.1021 jaro/3

- string_distance(Representation)

4.1022 jaro_winkler/3

- string_distance(Representation)

4.1023 join/3

- queuep

4.1024 join_all/3

- queuep

4.1025 join_aos_packet_zone/3

- ccsds_packet_services

4.1026 join_packet_zone/4

- ccsds_packet_services

4.1027 json_to_geojson/2

- geojson_protocol

4.1028 jump/3

- queuep

4.1029 jump_all/3

- queuep

4.1030 jump_all_block/3

- queuep

4.1031 k_distinct_subsequence/3

- subsequences_protocol

4.1032 k_distinct_subsequences/3

- subsequences_protocol

4.1033 k_permutation/3

- permutations_protocol

4.1034 k_permutation/4

- permutations_protocol

4.1035 k_permutations/3

- permutations_protocol

4.1036 k_permutations/4

- permutations_protocol

4.1037 key/2

- pairs

4.1038 keys/2

- dictionaryp
- pairs

4.1039 keys/3

- redis

4.1040 keys_values/3

- pairs

4.1041 keysort/2

- listp

4.1042 kill/1

- process

4.1043 kill/2

- process

4.1044 known_zone_id/1

- tzif_zone_ids

4.1045 kurtosis/2

- statisticsp

4.1046 lambert_azimuthal_equal_area/3

- crs_projections_protocol

4.1047 lambert_azimuthal_equal_area_inverse/3

- crs_projections_protocol

4.1048 language/3

- iso_639_1
- iso_639_protocol

4.1049 language/5

- iso_639_3
- iso_639_protocol

4.1050 language_code/5

- iso_639_2
- iso_639_protocol

4.1051 language_group/2

- iso_639_5
- iso_639_protocol

4.1052 language_object/2

- graph_language_registry

4.1053 last/2

- listp
- varlistp

4.1054 leaf/1

- hierarchyp

4.1055 leaf_class/1

- class_hierarchyp

4.1056 leaf_classes/1

- class_hierarchyp

4.1057 leaf_instance/1

- class_hierarchyp

4.1058 leaf_instances/1

- class_hierarchyp

4.1059 leap_effective_date/2

- time_scales_data

4.1060 leap_offset_at_utc_unix/2

- time_scales_data

4.1061 leap_second_date/2

- time_scales_protocol

4.1062 leap_seconds_entries/1

- time_scales_data
- time_scales_protocol

4.1063 leap_seconds_source/1

- time_scales_data
- time_scales_protocol

4.1064 leap_year/1

- datep
- iso8601

4.1065 learn/0

- metagol_example_protocol

4.1066 learn/1

- metagol_example_protocol

4.1067 learn/2

- anomaly_detector_protocol
- classifier_protocol
- clusterer_protocol
- dimension_reducer_protocol
- metagol
- ranker_protocol
- regressor_protocol

4.1068 learn/3

- anomaly_detector_protocol
- classifier_protocol
- clusterer_protocol
- dimension_reducer_protocol
- metagol
- ranker_protocol
- regressor_protocol
- truncated_svd_projection

4.1069 learn_seq/2

- metagol

4.1070 learn_with_timeout/4

- metagol

4.1071 leash/1

- debuggerp

4.1072 leashing/1

- debuggerp

4.1073 least_common_multiple/2

- numberlistp

4.1074 least_squares/3

- linear_algebra_protocol

4.1075 leaves/1

- hierarchyp

4.1076 leaves/2

- unweighted_directed_graph(Dictionary)

4.1077 length/2

- deque_protocol
- listp
- queuep
- varlistp

4.1078 levenshtein/3

- string_distance(Representation)

4.1079 libraries/0

- help

4.1080 libraries/1

- diagram(Format)
- diagrams(Format)
- lgtdocp

4.1081 libraries/2

- diagram(Format)
- diagrams(Format)
- lgtdocp

4.1082 libraries/3

- diagram(Format)
- diagrams(Format)

4.1083 library/1

- code_metric
- dead_code_scanner
- diagram(Format)
- diagrams(Format)
- help
- lgtdocp
- mutation_testing

4.1084 library/2

- code_metric
- dead_code_scanner
- diagram(Format)
- diagrams(Format)
- lgtdocp
- mutation_testing

4.1085 library_mutants/2

- mutation_testing

4.1086 library_mutants/3

- mutation_testing

4.1087 library_score/2

- code_metric

4.1088 license/1

- application_protocol
- pack_protocol

4.1089 linda/0

- linda_server

4.1090 linda/1

- linda_server

4.1091 linda_client/1

- linda_client

4.1092 linda_client/2

- linda_client

4.1093 linda_timeout/2

- linda_client

4.1094 line_to_chars/2

- reader

4.1095 line_to_chars/3

- reader

4.1096 line_to_codes/2

- reader

4.1097 line_to_codes/3

- reader

4.1098 linear_regression/4

- numberlistp

4.1099 lint/0

- packs
- registries

4.1100 lint/1

- packs
- registries

4.1101 lint/2

- packs

4.1102 list/0

- registries

4.1103 list_to_array/2

- java_utils_protocol

4.1104 listing/0

- listing

4.1105 listing/1

- listing

4.1106 little_endian_word32/2

- hash_common_32

4.1107 llen/3

- redis

4.1108 load/1

- mime_types
- tzif_protocol

4.1109 load/2

- mime_types
- tzif_protocol

4.1110 load_config/1

- subprocess_mutation_hook

4.1111 load_coverage_config/1

- subprocess_coverage_hook

4.1112 load_dut1_override/1

- time_scales_data
- time_scales_protocol

4.1113 load_leap_seconds_override/1

- time_scales_data
- time_scales_protocol

4.1114 load_program/1

- datalog_protocol

4.1115 loaded_file/1

- logtalk

4.1116 loaded_file_property/2

- logtalk
- modules_diagram_support

4.1117 loaded_files_topological_sort/1

- logtalk

4.1118 loaded_files_topological_sort/2

- logtalk

4.1119 loaded_pack/3

- packs

4.1120 loaded_pack_dependency/6

- packs

4.1121 loaded_pack_file/4

- packs

4.1122 loader_file/1

- application_protocol

4.1123 local_abbreviation/2

- tzif_protocol

4.1124 local_abbreviation/3

- tzif_protocol

4.1125 local_abbreviation/4

- tzif_protocol

4.1126 local_abbreviation_reified/2

- tzif_protocol

4.1127 local_abbreviation_reified/3

- tzif_protocol

4.1128 local_abbreviation_reified/4

- tzif_protocol

4.1129 local_abbreviation_with_resolution/3

- tzif_protocol

4.1130 local_abbreviation_with_resolution/4

- tzif_protocol

4.1131 local_abbreviation_with_resolution/5

- tzif_protocol

4.1132 local_daylight_saving_time/2

- tzif_protocol

4.1133 local_daylight_saving_time/3

- tzif_protocol

4.1134 local_daylight_saving_time/4

- tzif_protocol

4.1135 local_daylight_saving_time_reified/2

- tzif_protocol

4.1136 local_daylight_saving_time_reified/3

- tzif_protocol

4.1137 local_daylight_saving_time_reified/4

- tzif_protocol

4.1138 local_daylight_saving_time_with_resolution/3

- tzif_protocol

4.1139 local_daylight_saving_time_with_resolution/4

- tzif_protocol

4.1140 local_daylight_saving_time_with_resolution/5

- tzif_protocol

4.1141 local_offset/2

- tzif_protocol

4.1142 local_offset/3

- tzif_protocol

4.1143 local_offset/4

- tzif_protocol

4.1144 local_offset_reified/2

- tzif_protocol

4.1145 local_offset_reified/3

- tzif_protocol

4.1146 local_offset_reified/4

- tzif_protocol

4.1147 local_offset_with_resolution/3

- tzif_protocol

4.1148 local_offset_with_resolution/4

- tzif_protocol

4.1149 local_offset_with_resolution/5

- tzif_protocol

4.1150 local_tangent_plane/3

- crs_projections_protocol

4.1151 local_tangent_plane_inverse/3

- crs_projections_protocol

4.1152 local_time_type/2

- tzif_protocol

4.1153 local_time_type/3

- tzif_protocol

4.1154 local_time_type/4

- tzif_protocol

4.1155 local_time_type_reified/2

- tzif_protocol

4.1156 local_time_type_reified/3

- tzif_protocol

4.1157 local_time_type_reified/4

- tzif_protocol

4.1158 local_time_type_with_resolution/3

- tzif_protocol

4.1159 local_time_type_with_resolution/4

- tzif_protocol

4.1160 local_time_type_with_resolution/5

- tzif_protocol

4.1161 local_to_utc/3

- datep

4.1162 local_to_utc_tz/3

- dates_tz_protocol

4.1163 local_to_utc_tz_with_resolution/4

- dates_tz_protocol

4.1164 locked_pairs/2

- ranked_pairs

4.1165 log/3

- debuggerp

4.1166 log_event/2

- loggingp

4.1167 log_file/2

- loggingp

4.1168 logging/1

- loggingp

4.1169 logging/3

- debuggerp

4.1170 logistic/3

- sampling_protocol

4.1171 lognormal/3

- sampling_protocol

4.1172 logseries/2

- sampling_protocol

4.1173 logtalk_packs/0

- packs_common

4.1174 logtalk_packs/1

- packs_common

4.1175 long_flags/1

- command_line_option

4.1176 longest_common_increasing_subsequence/3

- subsequences_protocol

4.1177 longest_common_subsequence/3

- string_distance(Representation)
- subsequences_protocol

4.1178 longest_common_subsequence_length/3

- string_distance(Representation)

4.1179 longest_common_substring/3

- string_distance(Representation)

4.1180 longest_decreasing_subsequence/2

- subsequences_protocol

4.1181 longest_increasing_subsequence/2

- subsequences_protocol

4.1182 longest_repeating_subsequence/2

- subsequences_protocol

4.1183 lookup/2

- dictionaryp

4.1184 lookup/3

- dictionaryp

4.1185 lookup/4

- dictionaryp

4.1186 lookup_in/3

- nested_dictionary_protocol

4.1187 lower_triangular_part/2

- linear_algebra_protocol

4.1188 lower_triangular_part/3

- linear_algebra_protocol

4.1189 lower_upper/2

- characterp

4.1190 lpop/3

- redis

4.1191 lpush/4

- redis

4.1192 lrange/5

- redis

4.1193 lrem/5

- redis

4.1194 ltrim/5

- redis

4.1195 lucas/2

- natural

4.1196 magic/2

- magic

4.1197 magicise/4

- magic

4.1198 make_directory/1

- osp

4.1199 make_directory_path/1

- osp

4.1200 make_set/3

- union_find_protocol

4.1201 man/1

- help

4.1202 manhattan_distance/3

- numberlistp

4.1203 manhattan_norm/2

- numberlistp

4.1204 map/2

- deque_protocol
- dictionaryp
- expected(Expected)
- metap
- optional(Optional)
- queuep
- validation(Validation)

4.1205 map/3

- deque_protocol
- dictionaryp
- metap
- pairs
- queuep
- validated

4.1206 map/4

- metap
- validated

4.1207 map/5

- metap

4.1208 map/6

- metap

4.1209 map/7

- metap

4.1210 map/8

- metap

4.1211 map_both/3

- expected(Expected)
- validation(Validation)

4.1212 map_catching/2

- expected(Expected)
- validation(Validation)

4.1213 map_element/2

- java_utils_protocol

4.1214 map_id/2

- ccsds_tc_services

4.1215 map_invalid/2

- validation(Validation)

4.1216 map_or_else/3

- expected(Expected)
- optional(Optional)
- validation(Validation)

4.1217 map_reduce/5

- metap

4.1218 map_unexpected/2

- expected(Expected)

4.1219 mask32/1

- hash_common_32

4.1220 mask64/1

- hash_common_64

4.1221 master_channel_frame_count/2

- ccsds_tm_frames(FrameLength,SecondaryHeaderLength,HasFECF)

4.1222 materialize/0

- datalog_protocol

4.1223 matrix_column/3

- linear_algebra_protocol

4.1224 matrix_column_means/2

- linear_algebra_protocol

4.1225 matrix_diagonal/2

- linear_algebra_protocol

4.1226 matrix_diagonal/3

- linear_algebra_protocol

4.1227 matrix_matrix_product/3

- linear_algebra_protocol

4.1228 matrix_rank/2

- linear_algebra_protocol

4.1229 matrix_rank/3

- linear_algebra_protocol

4.1230 matrix_row/3

- linear_algebra_protocol

4.1231 matrix_row_means/2

- linear_algebra_protocol

4.1232 matrix_trace/2

- linear_algebra_protocol

4.1233 matrix_value/4

- linear_algebra_protocol

4.1234 matrix_vector_product/3

- linear_algebra_protocol

4.1235 max/2

- listp
- numberlistp
- statisticsp

4.1236 max/3

- dictionaryp

4.1237 max_clauses/1

- metagol

4.1238 max_inv_preds/1

- metagol

4.1239 max_path/5

- graph_protocol

4.1240 max_size/1

- arbitrary

4.1241 max_tree/3

- weighted_undirected_graph(Dictionary)

4.1242 maximal_cliques/2

- undirected_graph_common

4.1243 maximum_cliques/2

- undirected_graph_common

4.1244 maybe/0

- random_protocol

4.1245 maybe/1

- random_protocol

4.1246 maybe/2

- random_protocol

4.1247 maybe_call/1

- random_protocol

4.1248 maybe_call/2

- random_protocol

4.1249 mean_center/2

- geospatial_protocol

4.1250 mean_deviation/2

- statisticsp

4.1251 mean_squared_error/3

- statisticsp

4.1252 measurement/4

- pairwise_measurement_dataset_protocol

4.1253 median/2

- numberlistp
- statisticsp

4.1254 median_deviation/2

- statisticsp

4.1255 meets/2

- interval_protocol

4.1256 member/2

- interval_relation_set_protocol
- listp
- random_protocol
- setp

4.1257 memberchk/2

- listp
- setp
- varlistp

4.1258 merge/3

- heap_protocol

4.1259 message_body/2

- amqp

4.1260 message_delivery_tag/2

- amqp

4.1261 message_exchange/2

- amqp

4.1262 message_hook/4

- logtalk

4.1263 message_prefix_file/6

- logtalk

4.1264 message_prefix_stream/4

- logtalk

4.1265 message_properties/2

- amqp

4.1266 message_property/3

- amqp

4.1267 message_routing_key/2

- amqp

4.1268 message_tokens//2

- logtalk

4.1269 met_by/2

- interval_protocol

4.1270 meta/1

- command_line_option

4.1271 meta_type/3

- type

4.1272 metaphone/2

- string_distance(Representation)

4.1273 metaphone_match/2

- string_distance(Representation)

4.1274 metarule/6

- metagol

4.1275 metarule_next_id/1

- metagol

4.1276 method/2

- json_rpc

4.1277 method_not_found/2

- json_rpc

4.1278 mget/3

- memcached
- redis

4.1279 mibenum/1

- character_set_protocol

4.1280 midpoint/3

- geospatial_protocol

4.1281 milliseconds/2

- ccsds_cds(DaySegmentOctets,SubmillisecondOctets,Epoch)

4.1282 min/2

- listp
- numberlistp
- statisticsp

4.1283 min/3

- dictionaryp

4.1284 min_clauses/1

- metagol

4.1285 min_distances/3

- graph_protocol

4.1286 min_max/3

- numberlistp
- statisticsp

4.1287 min_max_normalization/2

- statisticsp

4.1288 min_path/5

- graph_protocol

4.1289 min_path_bellman_ford/5

- weighted_graph_protocol

4.1290 min_paths/3

- weighted_directed_graph(Dictionary)

4.1291 min_predecessors/3

- graph_protocol

4.1292 min_tree/3

- weighted_undirected_graph(Dictionary)

4.1293 mine/2

- pattern_miner_protocol

4.1294 mine/3

- pattern_miner_protocol

4.1295 minimum_enclosing_circle/3

- geospatial_protocol

4.1296 minkowski_distance/4

- numberlistp

4.1297 modes/2

- numberlistp
- statisticsp

4.1298 module_property/2

- modules_diagram_support

4.1299 monitor/1

- event_registryp

4.1300 monitor/4

- event_registryp

4.1301 monitor_activated/0

- monitorp

4.1302 monitored/1

- event_registryp

4.1303 monitors/1

- event_registryp

4.1304 month/2

- ccsds_ccs(CalendarVariant,FractionOctets)

4.1305 month_weekday_date/5

- datep

4.1306 mset/3

- redis

4.1307 msort/2

- listp

4.1308 msort/3

- listp

4.1309 mul32/3

- hash_common_32

4.1310 mul64/3

- hash_common_64

4.1311 multinomial/2

- natural

4.1312 multiset/3

- multisets_protocol

4.1313 multiset/4

- multisets_protocol

4.1314 multiset_index/4

- multisets_protocol

4.1315 multiset_index/5

- multisets_protocol

4.1316 multisets/3

- multisets_protocol

4.1317 multisets/4

- multisets_protocol

4.1318 mutation/2

- mutator_protocol

4.1319 mutation/3

- mutations
- mutations_store

4.1320 nack/3

- stomp

4.1321 name/1

- application_protocol
- character_set_protocol
- command_line_option
- pack_protocol
- registry_protocol

4.1322 name_of_day/3

- datep

4.1323 name_of_month/3

- datep

4.1324 nan_kind/2

- ieee_754_fields_protocol

4.1325 nan_payload/2

- ieee_754_fields_protocol

4.1326 nan_representation/1

- ieee_754_protocol

4.1327 natural//1

- number_grammars(Format)

4.1328 nearest_coordinate/5

- geospatial_protocol

4.1329 nearest_point_on_polyline/4

- geospatial_protocol

4.1330 nearest_point_on_segment/4

- geospatial_protocol

4.1331 neighbor_state/2

- simulated_annealing_protocol

4.1332 neighbor_state/3

- simulated_annealing_protocol

4.1333 neighbors/2

- geohash_protocol

4.1334 neighbors/3

- graph_protocol

4.1335 neighbors_bits/3

- geohash_protocol

4.1336 neighbors_int/3

- geohash_protocol

4.1337 network/1

- interval_constraint_network_protocol

4.1338 new/1

- graph_protocol
- java_access_protocol
- nested_dictionary_protocol
- streamvars
- termp

4.1339 new/2

- graph_protocol
- interval_constraint_network_protocol
- java_access_protocol
- streamvars
- union_find_protocol

4.1340 new/3

- graph_protocol
- interval_protocol

4.1341 new_line//0

- blank_grammars(Format)

4.1342 new_lines//0

- blank_grammars(Format)

4.1343 new_matrix/4

- linear_algebra_protocol

4.1344 new_vector/3

- linear_algebra_protocol

4.1345 new_vector_like/2

- linear_algebra_protocol

4.1346 next/2

- zipperp

4.1347 next/3

- zipperp

4.1348 next/4

- dictionaryp

4.1349 next_arrangement/3

- arrangements_protocol

4.1350 next_cartesian_tuple/3

- cartesian_products_protocol

4.1351 next_combination/3

- combinations_protocol

4.1352 next_derangement/3

- derangements_protocol

4.1353 next_multiset/3

- multisets_protocol

4.1354 next_partition/3

- partitions_protocol

4.1355 next_partition/4

- partitions_protocol

4.1356 next_permutation/2

- permutations_protocol

4.1357 nextto/3

- listp
- varlistp

4.1358 node/7

- graph_language_protocol

4.1359 nodebug/0

- debuggerp

4.1360 nodes/2

- interval_constraint_network_protocol

4.1361 nolog/3

- debuggerp

4.1362 nologall/0

- debuggerp

4.1363 non_blank//1

- blank_grammars(Format)

4.1364 non_blanks//1

- blank_grammars(Format)

4.1365 nonempty_subsequence/2

- subsequences_protocol

4.1366 nonempty_subsequences/2

- subsequences_protocol

4.1367 normal/3

- sampling_protocol

4.1368 normal_element/2

- html

4.1369 normalize/2

- interval_relation_set_protocol
- url(Representation)

4.1370 normalize_coordinate/2

- geospatial_protocol

4.1371 normalize_date_time/2

- datep

4.1372 normalize_polygon_orientation/3

- geospatial_protocol

4.1373 normalize_range/2

- numberlistp

4.1374 normalize_range/4

- numberlistp

4.1375 normalize_scalar/2

- numberlistp

4.1376 normalize_unit/2

- numberlistp

4.1377 normalize_vector/2

- linear_algebra_protocol

4.1378 normalize_vector/3

- linear_algebra_protocol

4.1379 nospy/1

- debuggerp

4.1380 nospy/3

- debuggerp

4.1381 nospy/4

- debuggerp

4.1382 nospyall/0

- debuggerp

4.1383 not64/2

- hash_common_64

4.1384 note/2

- registry_protocol

4.1385 note/3

- pack_protocol

4.1386 notification/2

- json_rpc

4.1387 notification/3

- json_rpc

4.1388 notrace/0

- debuggerp

4.1389 now/3

- timep

4.1390 nth0/3

- listp
- varlistp

4.1391 nth0/4

- listp
- varlistp

4.1392 nth1/3

- listp
- varlistp

4.1393 nth1/4

- listp
- varlistp

4.1394 nth_arrangement/4

- arrangements_protocol

4.1395 nth_arrangement/5

- arrangements_protocol

4.1396 nth_cartesian_tuple/3

- cartesian_products_protocol

4.1397 nth_cartesian_tuple/4

- cartesian_products_protocol

4.1398 nth_combination/4

- combinations_protocol

4.1399 nth_combination/5

- combinations_protocol

4.1400 nth_derangement/3

- derangements_protocol

4.1401 nth_derangement/4

- derangements_protocol

4.1402 nth_distinct_arrangement/4

- arrangements_protocol

4.1403 nth_distinct_cartesian_tuple/3

- cartesian_products_protocol

4.1404 nth_distinct_combination/4

- combinations_protocol

4.1405 nth_distinct_derangement/3

- derangements_protocol

4.1406 nth_distinct_multiset/4

- multisets_protocol

4.1407 nth_distinct_partition/3

- partitions_protocol

4.1408 nth_distinct_partition/4

- partitions_protocol

4.1409 nth_distinct_partition/5

- partitions_protocol

4.1410 nth_distinct_permutation/3

- permutations_protocol

4.1411 nth_multiset/4

- multisets_protocol

4.1412 nth_multiset/5

- multisets_protocol

4.1413 nth_partition/3

- partitions_protocol

4.1414 nth_partition/4

- partitions_protocol

4.1415 nth_partition/5

- partitions_protocol

4.1416 nth_permutation/3

- permutations_protocol

4.1417 nth_permutation/4

- permutations_protocol

4.1418 null/1

- java_utils_protocol

4.1419 null_device_path/1

- osp

4.1420 null_space/2

- linear_algebra_protocol

4.1421 null_space/3

- linear_algebra_protocol

4.1422 number//1

- number_grammars(Format)

4.1423 number_of_edges/2

- graph_protocol

4.1424 number_of_tests/1

- lgtunit

4.1425 number_of_vertices/2

- graph_protocol

4.1426 number_string/2

- string(Representation)

4.1427 numbervars/1

- termp

4.1428 numbervars/3

- termp

4.1429 occurrences/2

- listp

4.1430 occurrences/3

- listp

4.1431 occurs/2

- termp

4.1432 ocf/2

- ccsds_frame_protocol
- ccsds_frames

4.1433 ocf_flag/2

- ccsds_tm_frames(FrameLength,SecondaryHeaderLength,HasFECF)

4.1434 of/2

- optional

4.1435 of_expected/2

- expected

4.1436 of_invalid/2

- validation

4.1437 of_invalids/2

- validation

4.1438 of_unexpected/2

- expected

4.1439 of_valid/2

- validation

4.1440 offset/2

- tzif_protocol

4.1441 offset/3

- time_scales_protocol
- tzif_protocol

4.1442 offset/4

- tzif_protocol

4.1443 one_or_more//0

- sequence_grammars

4.1444 one_or_more//1

- sequence_grammars

4.1445 one_or_more//2

- sequence_grammars

4.1446 operating_system_machine/1

- osp

4.1447 operating_system_name/1

- osp

4.1448 operating_system_release/1

- osp

4.1449 operating_system_type/1

- osp

4.1450 option/2

- options_protocol

4.1451 option/3

- options_protocol

4.1452 or/2

- expected(Expected)
- optional(Optional)
- validation(Validation)

4.1453 or64/3

- hash_common_64

4.1454 or_else/2

- expected(Expected)
- optional(Optional)
- validation(Validation)

4.1455 or_else_call/2

- expected(Expected)
- optional(Optional)
- validation(Validation)

4.1456 or_else_fail/1

- expected(Expected)
- optional(Optional)
- validation(Validation)

4.1457 or_else_get/2

- expected(Expected)
- optional(Optional)
- validation(Validation)

4.1458 or_else_throw/1

- expected(Expected)
- validation(Validation)

4.1459 or_else_throw/2

- expected(Expected)
- optional(Optional)
- validation(Validation)

4.1460 order/1

- ieee_754_fields_protocol
- ieee_754_protocol

4.1461 originator/1

- application_protocol

4.1462 orphaned/0

- packs

4.1463 orphaned/2

- packs

4.1464 out/1

- linda_client

4.1465 out/2

- linda_client

4.1466 out_degree/3

- directed_graph_protocol

4.1467 outdated/0

- packs

4.1468 outdated/1

- packs

4.1469 outdated/2

- packs

4.1470 outdated/4

- packs

4.1471 outdated/5

- packs

4.1472 outer_product/3

- linear_algebra_protocol

4.1473 output_file_name/2

- graph_language_protocol

4.1474 output_schema/2

- mcp_tool_protocol

4.1475 overlapped_by/2

- interval_protocol

4.1476 overlaps/2

- interval_protocol

4.1477 pack_dependency/6

- packs

4.1478 pack_metadata/4

- packs

4.1479 pack_object/3

- packs

4.1480 pack_property/4

- packs

4.1481 package/1

- application_protocol

4.1482 packet_order_flag/2

- ccsds_tm_frames(FrameLength,SecondaryHeaderLength,HasFECF)

4.1483 packetize_aos_frames/7

- ccsds_packetization

4.1484 packetize_aos_frames/8

- ccsds_packetization

4.1485 packetize_aos_packets/6

- ccsds_packetization

4.1486 packetize_aos_packets/7

- ccsds_packetization

4.1487 packetize_tm_frames/7

- ccsds_packetization

4.1488 packetize_tm_frames/8

- ccsds_packetization

4.1489 packetize_tm_packets/6

- ccsds_packetization

4.1490 packetize_tm_packets/7

- ccsds_packetization

4.1491 packets/2

- ccsds_packet_services

4.1492 pad_md/4

- hash_common_32

4.1493 pairwise_dataset_connected_components/2

- ranking_dataset_common

4.1494 pairwise_dataset_declared_items/2

- ranking_dataset_common

4.1495 pairwise_dataset_items/2

- ranking_dataset_common

4.1496 pairwise_dataset_matchups/2

- ranking_dataset_common

4.1497 pairwise_dataset_preferences/2

- ranking_dataset_common

4.1498 pairwise_dataset_summary/2

- ranking_dataset_common

4.1499 pairwise_dataset_win_totals/2

- ranking_dataset_common

4.1500 pairwise_measurement_dataset_connected_components/2

- ranking_dataset_common

4.1501 pairwise_measurement_dataset_declared_items/2

- ranking_dataset_common

4.1502 pairwise_measurement_dataset_items/2

- ranking_dataset_common

4.1503 pairwise_measurement_dataset_measurements/2

- ranking_dataset_common

4.1504 pairwise_measurement_dataset_summary/2

- ranking_dataset_common

4.1505 params/2

- json_rpc

4.1506 parent/1

- proto_hierarchyp

4.1507 parent/2

- geohash_protocol

4.1508 parenthesis/2

- characterp

4.1509 parents/1

- proto_hierarchyp

4.1510 parse/2

- avro
- base32
- base58
- base64
- base64url
- base85
- cbor(StringRepresentation)
- ccsds_frame_protocol
- ccsds_packets(SecondaryHeaderLength)
- ccsds_time_code_protocol
- geojson_protocol

- ieee_754_protocol
- json_ld_protocol
- json_lines_protocol
- json_pointer_protocol
- json_protocol
- json_schema_protocol
- message_pack(StringRepresentation)
- nmea_protocol
- protobuf
- tle_orbits_protocol
- toml_protocol
- toon_protocol
- url(Representation)
- wkt_wkb_protocol
- xml
- yaml_protocol

4.1511 parse/3

- avro
- ccsds_time_fields
- nmea_protocol
- protobuf
- xml

4.1512 parse/4

- command_line_options

4.1513 parse/5

- command_line_options

4.1514 parse_all/2

- yaml_protocol

4.1515 parse_domain/2

- pddl

4.1516 parse_domain/3

- pddl

4.1517 parse_error/1

- json_rpc

4.1518 parse_fragment/2

- json_pointer_protocol

4.1519 parse_frame/3

- ccsds_link_profiles

4.1520 parse_frames/3

- ccsds_link_profiles

4.1521 parse_lines/4

- tle_orbits_protocol

4.1522 parse_problem/2

- pddl

4.1523 parse_problem/3

- pddl

4.1524 partial_map/4

- rbtree

4.1525 partition/2

- partitions_protocol

4.1526 partition/3

- either
- partitions_protocol
- validated

4.1527 partition/4

- metap
- partitions_protocol

4.1528 partition/5

- listp

4.1529 partition/6

- metap

4.1530 partition_index/3

- partitions_protocol

4.1531 partition_index/4

- partitions_protocol

4.1532 partition_index/5

- partitions_protocol

4.1533 partition_number/2

- natural

4.1534 partitions/2

- partitions_protocol

4.1535 partitions/3

- partitions_protocol

4.1536 partitions/4

- partitions_protocol

4.1537 path/3

- graph_protocol

4.1538 path_concat/3

- osp

4.1539 path_consistency/2

- interval_constraint_network_protocol

4.1540 pattern_miner_options/2

- pattern_miner_protocol

4.1541 peek_back/2

- deque_protocol

4.1542 peek_front/2

- deque_protocol

4.1543 pending_data/2

- ccsds_packet_services

4.1544 pending_fragments/2

- ccsds_link_profiles
- ccsds_packet_services
- ccsds_tc_services

4.1545 pending_packets/2

- ccsds_packetization

4.1546 percentile/3

- statisticsp

4.1547 period/1

- temporal_pairwise_ranking_dataset_protocol

4.1548 permutation/2

- listp
- permutations_protocol
- random_protocol
- varlistp

4.1549 permutation/3

- permutations_protocol

4.1550 permutation_index/3

- permutations_protocol

4.1551 permutation_index/4

- permutations_protocol

4.1552 permutations/2

- permutations_protocol

4.1553 permutations/3

- permutations_protocol

4.1554 persist/3

- redis

4.1555 pid/1

- osp

4.1556 pin/0

- packs_common

4.1557 pin/1

- packs_common

4.1558 pinned/1

- packs_common

4.1559 plus/3

- integer

4.1560 point_in_polygon/2

- geospatial_protocol

4.1561 point_to_polyline_distance/3

- geospatial_protocol

4.1562 poisson/2

- sampling_protocol

4.1563 polygon_area/2

- geospatial_protocol

4.1564 polygon_bounding_box/2

- geospatial_protocol

4.1565 polygon_centroid/2

- geospatial_protocol

4.1566 polygon_covering/4

- geohash_protocol

4.1567 polygon_orientation/2

- geospatial_protocol

4.1568 polygon_perimeter/2

- geospatial_protocol

4.1569 polygon_perimeter/3

- geospatial_protocol

4.1570 polygons_intersect/2

- geospatial_protocol

4.1571 polyline_covering/4

- geohash_protocol

4.1572 polyline_length/2

- geospatial_protocol

4.1573 polyline_length/3

- geospatial_protocol

4.1574 polyline_resample/3

- geospatial_protocol

4.1575 polyline_simplify/3

- geospatial_protocol

4.1576 polyline_split_at_distance/4

- geospatial_protocol

4.1577 pop_back/3

- deque_protocol

4.1578 pop_front/3

- deque_protocol

4.1579 port/5

- ports_profiler

4.1580 portray_clause/1

- listing

4.1581 possible/4

- interval_constraint_network_protocol

4.1582 postorder/2

- bintree

4.1583 power/2

- sampling_protocol

4.1584 power_sequence/4

- integer

4.1585 power_set/2

- subsequences_protocol

4.1586 powerset/2

- setp

4.1587 pp/1

- xml

4.1588 pprint/1

- metagol

4.1589 precision/1

- ieee_754_fields_protocol
- ieee_754_protocol

4.1590 precision/3

- geohash_protocol

4.1591 predicate/1

- caller_diagram(Format)

4.1592 predicate/2

- caller_diagram(Format)
- dead_code_scanner
- mutation_testing

4.1593 predicate/3

- dead_code_scanner
- mutation_testing

4.1594 predicate_info_pair_score_hook/4

- doc_metric

4.1595 predicate_info_score_hook/3

- doc_metric

4.1596 predicate_mode_score_hook/3

- doc_metric

4.1597 predicate_mode_score_hook/5

- doc_metric

4.1598 predicate_mutants/3

- mutation_testing

4.1599 predicate_mutants/4

- mutation_testing

4.1600 predicate_stratum/3

- datalog_protocol

4.1601 predicates/2

- dead_code_scanner

4.1602 predicates/3

- dead_code_scanner

4.1603 predict/3

- anomaly_detector_protocol
- classifier_protocol
- regressor_protocol

4.1604 predict/4

- anomaly_detector_protocol
- knn_classifier
- nearest_centroid_classifier

4.1605 predict_distribution/3

- bayesian_ridge_regression
- gaussian_process_regression

4.1606 predict_probabilities/3

- probabilistic_classifier_protocol

4.1607 predict_probabilities/4

- knn_classifier
- nearest_centroid_classifier

4.1608 predict_scores/3

- lda_classifier
- qda_classifier

4.1609 preference/3

- pairwise_ranking_dataset_protocol

4.1610 preferred_mime_name/1

- character_set_protocol

4.1611 prefix/0

- packs_common

4.1612 prefix/1

- packs_common

4.1613 prefix/2

- listp
- varlistp

4.1614 prefix/3

- listp

4.1615 prefix_data/2

- ccsds_packet_services

4.1616 preorder/2

- bintree

4.1617 prepend/3

- memcached

4.1618 previous/2

- zipperp

4.1619 previous/3

- zipperp

4.1620 previous/4

- dictionaryp

4.1621 previous_arrangement/3

- arrangements_protocol

4.1622 previous_cartesian_tuple/3

- cartesian_products_protocol

4.1623 previous_combination/3

- combinations_protocol

4.1624 previous_derangement/3

- derangements_protocol

4.1625 previous_multiset/3

- multisets_protocol

4.1626 previous_partition/3

- partitions_protocol

4.1627 previous_partition/4

- partitions_protocol

4.1628 previous_permutation/2

- permutations_protocol

4.1629 primary_office_bic/1

- iso_9362_protocol

4.1630 print_anomaly_detector/1

- anomaly_detector_protocol

4.1631 print_classifier/1

- classifier_protocol

4.1632 print_clusterer/1

- clusterer_protocol

4.1633 print_dimension_reducer/1

- dimension_reducer_protocol

4.1634 print_flags/0

- flags
- flags_validator

4.1635 print_flags/1

- flags

4.1636 print_message/3

- logtalk

4.1637 print_message_token/4

- logtalk

4.1638 print_message_tokens/3

- logtalk

4.1639 print_pattern_miner/1

- pattern_miner_protocol

4.1640 print_ranker/1

- ranker_protocol

4.1641 print_regressor/1

- regressor_protocol

4.1642 product/2

- numberlistp
- statisticsp

4.1643 product/3

- setp

4.1644 program_to_clauses/2

- metagol

4.1645 progress/5

- simulated_annealing_protocol

4.1646 prompt_get/3

- mcp_prompt_protocol

4.1647 prompts/1

- mcp_prompt_protocol

4.1648 propagate/2

- interval_constraint_network_protocol

4.1649 propagate/3

- interval_constraint_network_protocol
- tle_orbits_protocol

4.1650 propagate/4

- tle_orbits_protocol

4.1651 propagate/5

- tle_orbits_protocol

4.1652 propagate_state/4

- tle_orbits_protocol

4.1653 propagate_state/5

- tle_orbits_protocol

4.1654 propagation_triple/2

- interval_constraint_network_protocol

4.1655 propagation_triples/2

- interval_constraint_network_protocol

4.1656 proper_prefix/2

- listp

4.1657 proper_prefix/3

- listp

4.1658 proper_subsequence/2

- subsequences_protocol

4.1659 proper_suffix/2

- listp

4.1660 proper_suffix/3

- listp

4.1661 property/1

- application_protocol

4.1662 prove/2

- interpreterp

4.1663 prove/3

- interpreterp

4.1664 provides/2

- registries

4.1665 prune/3

- c45_classifier

4.1666 prune/5

- c45_classifier

4.1667 pseudo_inverse/2

- linear_algebra_protocol

4.1668 pseudo_inverse/3

- linear_algebra_protocol

4.1669 push_back/3

- deque_protocol

4.1670 push_front/3

- deque_protocol

4.1671 qr_decomposition/3

- linear_algebra_protocol

4.1672 quartiles/4

- statisticsp

4.1673 query/1

- datalog_protocol

4.1674 query/2

- datalog_protocol

4.1675 question_hook/6

- logtalk

4.1676 question_prompt_stream/4

- logtalk

4.1677 queue_bind/4

- amqp

4.1678 queue_declare/3

- amqp

4.1679 queue_delete/3

- amqp

4.1680 queue_purge/2

- amqp

4.1681 queue_unbind/4

- amqp

4.1682 quick_check/1

- lgtunit

4.1683 quick_check/2

- lgtunit

4.1684 quick_check/3

- lgtunit

4.1685 random/1

- random_protocol

4.1686 random/3

- random_protocol

4.1687 random_arrangement/3

- arrangements_protocol

4.1688 random_cartesian_tuple/2

- cartesian_products_protocol

4.1689 random_combination/3

- combinations_protocol

4.1690 random_derangement/2

- derangements_protocol

4.1691 random_distinct_arrangement/3

- arrangements_protocol

4.1692 random_distinct_cartesian_tuple/2

- cartesian_products_protocol

4.1693 random_distinct_combination/3

- combinations_protocol

4.1694 random_distinct_derangement/2

- derangements_protocol

4.1695 random_distinct_multiset/3

- multisets_protocol

4.1696 random_distinct_partition/2

- partitions_protocol

4.1697 random_distinct_partition/3

- partitions_protocol

4.1698 random_distinct_permutation/2

- permutations_protocol

4.1699 random_multiset/3

- multisets_protocol

4.1700 random_node/1

- uuid_protocol

4.1701 random_partition/2

- partitions_protocol

4.1702 random_partition/3

- partitions_protocol

4.1703 random_permutation/2

- permutations_protocol

4.1704 random_subsequence/2

- subsequences_protocol

4.1705 random_tree/1

- benchmark_generators

4.1706 randomize/1

- fast_random(Algorithm)
- random(Algorithm)

4.1707 randseq/4

- random_protocol

4.1708 randset/4

- random_protocol

4.1709 range/2

- statisticsp

4.1710 rank/3

- ranker_protocol

4.1711 rank_correlation/3

- statisticsp

4.1712 ranker_options/2

- ranker_protocol

4.1713 rd/1

- linda_client

4.1714 rd/2

- linda_client

4.1715 rd_list/2

- linda_client

4.1716 rd_list/3

- linda_client

4.1717 rd_noblock/1

- linda_client

4.1718 rd_noblock/2

- linda_client

4.1719 rdirectories/1

- lgtdocp

4.1720 rdirectories/2

- lgtdocp

4.1721 rdirectory/1

- code_metric
- dead_code_scanner
- diagram(Format)
- diagrams(Format)
- lgtdocp
- wrapper

4.1722 rdirectory/2

- code_metric
- dead_code_scanner
- diagram(Format)
- diagrams(Format)
- lgtdocp
- wrapper

4.1723 rdirectory/3

- diagram(Format)
- diagrams(Format)

4.1724 rdirectory_score/2

- code_metric

4.1725 reachable/3

- graph_protocol

4.1726 read_file/2

- csv_protocol
- read_file
- tsv_protocol

4.1727 read_file/3

- csv_protocol
- tsv_protocol

4.1728 read_file_by_line/2

- csv_protocol
- tsv_protocol

4.1729 read_file_by_line/3

- csv_protocol
- tsv_protocol

4.1730 read_framed_message/2

- json_rpc

4.1731 read_from_atom/2

- term_io_protocol

4.1732 read_from_chars/2

- term_io_protocol

4.1733 read_from_codes/2

- term_io_protocol

4.1734 read_message/2

- json_rpc

4.1735 read_mime_types/2

- mime_types

4.1736 read_only_device_path/1

- osp

4.1737 read_stream/2

- csv_protocol
- tsv_protocol

4.1738 read_stream/3

- csv_protocol
- tsv_protocol

4.1739 read_stream_by_line/2

- csv_protocol
- tsv_protocol

4.1740 read_stream_by_line/3

- csv_protocol
- tsv_protocol

4.1741 read_term_from_atom/3

- term_io_protocol

4.1742 read_term_from_chars/3

- term_io_protocol

4.1743 read_term_from_chars/4

- term_io_protocol

4.1744 read_term_from_codes/3

- term_io_protocol

4.1745 read_term_from_codes/4

- term_io_protocol

4.1746 readme/1

- packs_common

4.1747 readme/2

- packs_common

4.1748 reassemble_aos_frames/5

- ccsds_packet_services

4.1749 reassemble_aos_frames/6

- ccsds_packet_services

4.1750 reassemble_aos_frames/7

- ccsds_packet_services

4.1751 reassemble_aos_packets/5

- ccsds_packet_services

4.1752 reassemble_aos_packets/6

- ccsds_packet_services

4.1753 reassemble_aos_packets/7

- ccsds_packet_services

4.1754 reassemble_frames/6

- ccsds_link_profiles

4.1755 reassemble_frames/7

- ccsds_link_profiles

4.1756 reassemble_frames/8

- ccsds_link_profiles

4.1757 reassemble_packet_zone/5

- ccsds_packet_services

4.1758 reassemble_packets/6

- ccsds_link_profiles

4.1759 reassemble_packets/7

- ccsds_link_profiles

4.1760 reassemble_packets/8

- ccsds_link_profiles

4.1761 reassemble_tc_frame/4

- ccsds_tc_services

4.1762 reassemble_tc_frame/5

- ccsds_tc_services

4.1763 reassemble_tc_frame/6

- ccsds_tc_services

4.1764 reassemble_tc_frame_with_provenance/5

- ccsds_tc_services

4.1765 reassemble_tc_frame_with_provenance/6

- ccsds_tc_services

4.1766 reassemble_tc_frames/4

- ccsds_tc_services

4.1767 reassemble_tc_frames/5

- ccsds_tc_services

4.1768 reassemble_tc_frames/6

- ccsds_tc_services

4.1769 reassemble_tc_frames_with_provenance/5

- ccsds_tc_services

4.1770 reassemble_tc_frames_with_provenance/6

- ccsds_tc_services

4.1771 reassemble_tm_frames/5

- ccsds_packet_services

4.1772 reassemble_tm_frames/6

- ccsds_packet_services

4.1773 reassemble_tm_frames/7

- ccsds_packet_services

4.1774 reassemble_tm_packets/5

- ccsds_packet_services

4.1775 reassemble_tm_packets/6

- ccsds_packet_services

4.1776 reassemble_tm_packets/7

- ccsds_packet_services

4.1777 receive/3

- amqp
- stomp

4.1778 recorda/2

- recorded_database_core

4.1779 recorda/3

- recorded_database_core

4.1780 recorded/2

- recorded_database_core

4.1781 recorded/3

- recorded_database_core

4.1782 recordz/2

- recorded_database_core

4.1783 recordz/3

- recorded_database_core

4.1784 refine/5

- interval_constraint_network_protocol

4.1785 refine_propagate/3

- interval_constraint_network_protocol

4.1786 refine_propagate/4

- interval_constraint_network_protocol

4.1787 refine_propagate/5

- interval_constraint_network_protocol

4.1788 refine_propagate/6

- interval_constraint_network_protocol

4.1789 regressor_options/2

- regressor_protocol

4.1790 relation/1

- interval_algebra_protocol

4.1791 relation/3

- interval_protocol

4.1792 relation/4

- interval_constraint_network_protocol

4.1793 relation_set/1

- interval_relation_set_protocol

4.1794 relative_standard_deviation/2

- statisticsp

4.1795 release/1

- amqp_pool

4.1796 release_date/1

- application_protocol

4.1797 relevance/3

- ranking_dataset_protocol

4.1798 removeDependent/1

- subject

4.1799 remove_duplicates/2

- listp
- varlistp

4.1800 remove_rule/1

- datalog_protocol

4.1801 rename/4

- redis

4.1802 rename_file/2

- osp

4.1803 replace/3

- zipperp

4.1804 replace/5

- memcached

4.1805 replace_sub_atom/4

- atom

4.1806 report_directory/3

- mutation_testing

4.1807 report_entity/3

- mutation_testing

4.1808 report_library/3

- mutation_testing

4.1809 report_predicate/4

- mutation_testing

4.1810 repository/1

- application_protocol

4.1811 repository_branch/1

- application_protocol

4.1812 repository_commit/1

- application_protocol

4.1813 repository_commit_abbreviated/1

- application_protocol

4.1814 repository_commit_author/1

- application_protocol

4.1815 repository_commit_date/1

- application_protocol

4.1816 repository_commit_message/1

- application_protocol

4.1817 request/3

- json_rpc

4.1818 request/4

- json_rpc

4.1819 rescale/3

- numberlistp

4.1820 reset/0

- counter
- debuggerp
- linter_reporter
- mime_types
- mutator_protocol
- packs_common
- ports_profiler

4.1821 reset/1

- ports_profiler

4.1822 reset_counter/1

- counters

4.1823 reset_counters/0

- counters

4.1824 reset_flags/0

- flags

4.1825 reset_flags/1

- flags

4.1826 reset_genint/0

- genint_core

4.1827 reset_genint/1

- genint_core

4.1828 reset_gensym/0

- gensym_core

4.1829 reset_gensym/1

- gensym_core

4.1830 reset_monitor/0

- monitorp

4.1831 reset_seed/0

- fast_random(Algorithm)
- random(Algorithm)

4.1832 residuals/2

- hodge_rank

4.1833 resize/2

- amqp_pool

4.1834 resource_read/3

- mcp_resource_protocol

4.1835 resources/1

- mcp_resource_protocol

4.1836 response/3

- json_rpc

4.1837 restore/1

- packs

4.1838 restore/2

- packs

4.1839 result/2

- json_rpc

4.1840 retract_fact/1

- datalog_protocol

4.1841 reverse/2

- listp
- varlistp

4.1842 rewind/2

- zipperp

4.1843 rewind/3

- zipperp

4.1844 rhumb_bearing/3

- geospatial_protocol

4.1845 rhumb_destination_point/4

- geospatial_protocol

4.1846 rhumb_distance/3

- geospatial_protocol

4.1847 rhumb_midpoint/3

- geospatial_protocol

4.1848 rising_factorial/3

- natural

4.1849 rlibraries/1

- lgtdocp

4.1850 rlibraries/2

- lgtdocp

4.1851 rlibrary/1

- code_metric
- dead_code_scanner
- diagram(Format)
- diagrams(Format)
- lgtdocp

4.1852 rlibrary/2

- code_metric
- dead_code_scanner
- diagram(Format)
- diagrams(Format)
- lgtdocp

4.1853 rlibrary_score/2

- code_metric

4.1854 rol32/3

- hash_common_32

4.1855 rol64/3

- hash_common_64

4.1856 rollback/0

- datalog_protocol

4.1857 root_mean_squared_error/3

- statisticsp

4.1858 ror32/3

- hash_common_32

4.1859 route_distance/2

- geospatial_protocol

4.1860 route_distance/3

- geospatial_protocol

4.1861 route_distance/4

- geospatial_protocol

4.1862 rpop/3

- redis

4.1863 rpush/4

- redis

4.1864 rule/2

- databasep

4.1865 rule/3

- databasep

4.1866 rule/4

- databasep

4.1867 rules/1

- datalog_protocol

4.1868 run/0

- lgtunit

4.1869 run/1

- lgtunit

4.1870 run/2

- lgtunit
- simulated_annealing(Problem,RandomAlgorithm)

4.1871 run/3

- simulated_annealing(Problem,RandomAlgorithm)

4.1872 run/4

- simulated_annealing(Problem,RandomAlgorithm)

4.1873 run_test_sets/1

- lgtunit

4.1874 sadd/4

- redis

4.1875 same_instant/2

- datep

4.1876 same_length/2

- listp
- varlistp

4.1877 same_length/3

- listp

4.1878 sample_arrangements/4

- arrangements_protocol

4.1879 sample_cartesian_tuples/3

- cartesian_products_protocol

4.1880 sample_combinations/4

- combinations_protocol

4.1881 sample_derangements/3

- derangements_protocol

4.1882 sample_distinct_arrangements/4

- arrangements_protocol

4.1883 sample_distinct_cartesian_tuples/3

- cartesian_products_protocol

4.1884 sample_distinct_combinations/4

- combinations_protocol

4.1885 sample_distinct_derangements/3

- derangements_protocol

4.1886 sample_distinct_multisets/4

- multisets_protocol

4.1887 sample_distinct_partitions/3

- partitions_protocol

4.1888 sample_distinct_partitions/4

- partitions_protocol

4.1889 sample_distinct_permutations/3

- permutations_protocol

4.1890 sample_multisets/4

- multisets_protocol

4.1891 sample_partitions/3

- partitions_protocol

4.1892 sample_partitions/4

- partitions_protocol

4.1893 sample_permutations/3

- permutations_protocol

4.1894 save/0

- wrapper

4.1895 save/1

- packs
- tzif_protocol
- wrapper

4.1896 save/2

- packs
- tzif_protocol

4.1897 save_dut1_entries/1

- time_scales_data
- time_scales_protocol

4.1898 save_leap_seconds_entries/1

- time_scales_data
- time_scales_protocol

4.1899 scalar_product/3

- numberlistp

4.1900 scale_matrix/3

- linear_algebra_protocol

4.1901 scale_vector/3

- linear_algebra_protocol

4.1902 scan_left/4

- metap

4.1903 scan_left_1/3

- metap

4.1904 scan_right/4

- metap

4.1905 scan_right_1/3

- metap

4.1906 scard/3

- redis

4.1907 score/3

- anomaly_detector_protocol
- isolation_forest_anomaly_detector

4.1908 score_all/3

- anomaly_detector_protocol
- isolation_forest_anomaly_detector

4.1909 scores/2

- ranker_protocol

4.1910 search/1

- packs

4.1911 secondary_header/2

- ccsds_packets(SecondaryHeaderLength)
- ccsds_tm_frames(FrameLength,SecondaryHeaderLength,HasFECF)

4.1912 secondary_header_flag/2

- ccsds_packets(SecondaryHeaderLength)
- ccsds_tm_frames(FrameLength,SecondaryHeaderLength,HasFECF)

4.1913 secondary_header_time/3

- ccsds_packets(SecondaryHeaderLength)

4.1914 segment_data/2

- ccsds_tc_services

4.1915 segment_header/2

- ccsds_tc_frames(FrameLength,SegmentHeaderLength,HasFECF)

4.1916 segment_header_suffix/2

- ccsds_tc_services

4.1917 segment_header_suffixes/2

- ccsds_tc_services

4.1918 segment_length_identifier/2

- ccsds_tm_frames(FrameLength,SecondaryHeaderLength,HasFECF)

4.1919 select/3

- listp
- random_protocol
- setp
- varlistp

4.1920 select/4

- listp
- random_protocol

4.1921 selectchk/3

- listp
- setp

4.1922 selectchk/4

- listp

4.1923 send/3

- redis

4.1924 send/4

- stomp

4.1925 send_heartbeat/1

- amqp
- stomp

4.1926 sentence_type/2

- nmea_protocol

4.1927 sequence/2

- either
- maybe
- sequence_dataset_protocol
- validated

4.1928 sequence/3

- integer

4.1929 sequence/4

- float
- integer
- random_protocol

4.1930 sequence/5

- float

4.1931 sequence_count/2

- ccsds_packets(SecondaryHeaderLength)

4.1932 sequence_flags/2

- ccsds_packets(SecondaryHeaderLength)
- ccsds_tc_services

4.1933 sequence_number/2

- ccsds_tc_frames(FrameLength,SegmentHeaderLength,HasFECF)

4.1934 sequential_occurrences/2

- listp

4.1935 sequential_occurrences/3

- listp

4.1936 serve/3

- queuep

4.1937 server_accept/4

- socket

4.1938 server_accept/5

- socket

4.1939 server_close/1

- socket

4.1940 server_open/2

- socket

4.1941 server_open/3

- socket

4.1942 server_open/4

- socket

4.1943 set/1

- counter

4.1944 set/3

- memcached

4.1945 set/4

- random_protocol
- redis

4.1946 set/5

- memcached

4.1947 set_element/2

- java_utils_protocol

4.1948 set_field/2

- java_access_protocol

4.1949 set_flag_value/2

- flags

4.1950 set_flag_value/3

- flags

4.1951 set_monitor/4

- event_registryp

4.1952 set_seed/1

- arbitrary
- pseudo_random_protocol

4.1953 set_spy_point/4

- monitorp

4.1954 set_write_max_depth/1

- debuggerp

4.1955 setrange/5

- redis

4.1956 setup/0

- packs_common

4.1957 shell/1

- osp

4.1958 shell/2

- osp

4.1959 shell_command/1

- doclet

4.1960 shift_matrix_diagonal/3

- linear_algebra_protocol

4.1961 shl64/3

- hash_common_64

4.1962 short_flags/1

- command_line_option

4.1963 shr64/3

- hash_common_64

4.1964 shrink/3

- arbitrary

4.1965 shrink_sequence/3

- arbitrary

4.1966 shrinker/1

- arbitrary

4.1967 shutdown_server/1

- linda_client

4.1968 sign//1

- number_grammars(Format)

4.1969 signaling_field/2

- ccsds_aos_frames(FrameLength,InsertZoneLength,HasOCF,HasFECF)

4.1970 singleton/2

- interval_relation_set_protocol

4.1971 singletons/2

- termp

4.1972 sismember/4

- redis

4.1973 size/2

- dictionaryp
- heap_protocol
- setp

4.1974 skewness/2

- statisticsp

4.1975 sleep/1

- osp

4.1976 sliding_window/3

- subsequences_protocol

4.1977 smembers/3

- redis

4.1978 softmax/2

- numberlistp

4.1979 softmax/3

- numberlistp

4.1980 software_heritage_identifier/1

- application_protocol

4.1981 solve_cholesky/3

- linear_algebra_protocol

4.1982 solve_linear_system/3

- linear_algebra_protocol

4.1983 solve_linear_systems/3

- linear_algebra_protocol

4.1984 solve_lower_triangular_matrix/3

- linear_algebra_protocol

4.1985 solve_upper_triangular/3

- linear_algebra_protocol

4.1986 solve_upper_triangular_matrix/3

- linear_algebra_protocol

4.1987 sort/2

- listp

4.1988 sort/3

- listp

4.1989 sort/4

- listp

4.1990 soundex/2

- string_distance(Representation)

4.1991 soundex_match/2

- string_distance(Representation)

4.1992 source_file_extension/1

- modules_diagram_support

4.1993 space//0

- blank_grammars(Format)

4.1994 spacecraft_id/2

- ccsds_frame_protocol
- ccsds_frames

4.1995 spaces//0

- blank_grammars(Format)

4.1996 split/3

- atom

4.1997 split/4

- listp

4.1998 split_aos_packet_zone/3

- ccsds_packet_services

4.1999 split_packet_zone/4

- ccsds_packet_services

4.2000 split_string/4

- string(Representation)

4.2001 spy/1

- debuggerp

4.2002 spy/3

- debuggerp

4.2003 spy/4

- debuggerp

4.2004 spy_point/4

- monitorp

4.2005 spying/1

- debuggerp

4.2006 spying/3

- debuggerp

4.2007 spying/4

- debuggerp

4.2008 srem/4

- redis

4.2009 stabilize_vector_sign/2

- linear_algebra_protocol

4.2010 stabilize_vector_sign/3

- linear_algebra_protocol

4.2011 standard_cauchy/3

- sampling_protocol

4.2012 standard_deviation/2

- statisticsp

4.2013 standard_error/2

- statisticsp

4.2014 standard_exponential/1

- sampling_protocol

4.2015 standard_gamma/2

- sampling_protocol

4.2016 standard_normal/1

- sampling_protocol

4.2017 standard_t/2

- sampling_protocol

4.2018 start/0

- ports_profiler
- shell

4.2019 start/2

- mcp_server

4.2020 start/3

- mcp_server

4.2021 start/4

- mcp_server

4.2022 start/5

- mcp_server

4.2023 start_redirect_to_file/2

- dump_trace

4.2024 started_by/2

- interval_protocol

4.2025 starts/2

- interval_protocol

4.2026 state_energy/2

- simulated_annealing_protocol

4.2027 stats/1

- amqp_pool

4.2028 stats/2

- memcached

4.2029 stats/3

- memcached

4.2030 stem/2

- stemmer_protocol

4.2031 stems/2

- stemmer_protocol

4.2032 stirling_first/3

- natural

4.2033 stirling_second/3

- natural

4.2034 stop/0

- ports_profiler

4.2035 stop_condition/3

- simulated_annealing_protocol

4.2036 stop_redirect_to_file/0

- dump_trace

4.2037 strata/1

- datalog_protocol

4.2038 stream_to_bytes/2

- reader

4.2039 stream_to_bytes/3

- reader

4.2040 stream_to_chars/2

- reader

4.2041 stream_to_chars/3

- reader

4.2042 stream_to_codes/2

- reader

4.2043 stream_to_codes/3

- reader

4.2044 stream_to_terms/2

- reader

4.2045 stream_to_terms/3

- reader

4.2046 string_chars/2

- string(Representation)

4.2047 string_codes/2

- string(Representation)

4.2048 string_concat/3

- string(Representation)

4.2049 string_length/2

- string(Representation)

4.2050 string_lower/2

- string(Representation)

4.2051 string_upper/2

- string(Representation)

4.2052 strlen/3

- redis

4.2053 strongest_paths/2

- schulze_ranker

4.2054 strongly_connected_components/2

- directed_graph_protocol

4.2055 sub_string/5

- string(Representation)

4.2056 subclass/1

- class_hierarchyp

4.2057 subclasses/1

- class_hierarchyp

4.2058 subdivision/4

- iso_3166_2
- iso_3166_protocol

4.2059 subfactorial/2

- natural

4.2060 sublist/2

- listp
- varlistp

4.2061 submilliseconds/2

- ccsds_cds(DaySegmentOctets,SubmillisecondOctets,Epoch)

4.2062 subscribe/4

- stomp

4.2063 subsequence/2

- subsequences_protocol

4.2064 subsequence/3

- listp
- subsequences_protocol

4.2065 subsequence/4

- listp

4.2066 subsequence_at_indices/3

- subsequences_protocol

4.2067 subsequence_length/2

- subsequences_protocol

4.2068 subsequences/2

- subsequences_protocol

4.2069 subsequences/3

- subsequences_protocol

4.2070 subsequences_with_min_span/3

- subsequences_protocol

4.2071 subset/2

- interval_relation_set_protocol
- setp

4.2072 subslices/2

- subsequences_protocol

4.2073 substitute/4

- listp

4.2074 subsumes/2

- interval_constraint_network_protocol
- termp

4.2075 subterm/2

- termp
- xml

4.2076 subtract/3

- listp
- setp
- varlistp

4.2077 subtract_duration/3

- datep

4.2078 subtract_matrices/3

- linear_algebra_protocol

4.2079 subtract_vectors/3

- linear_algebra_protocol

4.2080 succ/2

- integer

4.2081 suffix/2

- listp
- varlistp

4.2082 suffix/3

- listp

4.2083 suffix_alias/2

- mime_types

4.2084 suffix_data/2

- ccsds_packet_services

4.2085 sum/2

- numberlistp
- statisticsp

4.2086 sum_of_squares/2

- statisticsp

4.2087 summary/1

- linter_reporter

4.2088 superclass/1

- class_hierarchyp

4.2089 superclasses/1

- class_hierarchyp

4.2090 supplier/1

- application_protocol

4.2091 supported_range/2

- time_scales_protocol

4.2092 suspend_monitor/0

- monitorp

4.2093 swap/1

- expected(Expected)
- validation(Validation)

4.2094 swap/2

- random_protocol

4.2095 swap_consecutive/2

- random_protocol

4.2096 symdiff/3

- setp

4.2097 symmetric_closure/2

- directed_graph_protocol

4.2098 symmetric_eigen/3

- linear_algebra_protocol

4.2099 symmetric_eigen/4

- linear_algebra_protocol

4.2100 symmetric_eigen/5

- linear_algebra_protocol

4.2101 synchronization_flag/2

- ccsds_tm_frames(FrameLength,SecondaryHeaderLength,HasFECF)

4.2102 tab//0

- blank_grammars(Format)

4.2103 tabs//0

- blank_grammars(Format)

4.2104 tai_minus_utc_for_tai_unix/2

- time_scales_data

4.2105 tail/2

- subsequences_protocol

4.2106 tail1/2

- subsequences_protocol

4.2107 tails/2

- subsequences_protocol

4.2108 tails1/2

- subsequences_protocol

4.2109 take/3

- listp

4.2110 take/4

- listp

4.2111 talker/2

- nmea_protocol

4.2112 target/1

- regression_dataset_protocol
- target_supervised_dimension_reduction_dataset_protocol

4.2113 tcb_minus_tdb_approx/3

- time_scales_data

4.2114 tcg_minus_tt_approx/3

- time_scales_data

4.2115 tdb_minus_tt_approx/3

- time_scales_data

4.2116 temporal_pairwise_dataset_connected_components/2

- ranking_dataset_common

4.2117 temporal_pairwise_dataset_declared_items/2

- ranking_dataset_common

4.2118 temporal_pairwise_dataset_games/2

- ranking_dataset_common

4.2119 temporal_pairwise_dataset_games/3

- ranking_dataset_common

4.2120 temporal_pairwise_dataset_items/2

- ranking_dataset_common

4.2121 temporal_pairwise_dataset_periods/2

- ranking_dataset_common

4.2122 temporal_pairwise_dataset_summary/2

- ranking_dataset_common

4.2123 temporary_directory/1

- osp

4.2124 term/2

- sarif

4.2125 term/4

- sarif

4.2126 term_expansion/2

- expanding

4.2127 terms_to_array/2

- java_utils_protocol

4.2128 test/1

- lgtunit

4.2129 time_of_day/4

- ccsds_ccs(CalendarVariant,FractionOctets)

4.2130 time_stamp/1

- osp

4.2131 time_string/3

- iso8601

4.2132 time_type/2

- tzif_protocol

4.2133 time_type/3

- tzif_protocol

4.2134 time_type/4

- tzif_protocol

4.2135 timeout/1

- metagol

4.2136 timestamp/2

- ulid_protocol

4.2137 timestamp/8

- ulid_protocol

4.2138 to_expected/1

- validation(Validation)

4.2139 to_expected/2

- optional(Optional)

4.2140 to_optional/1

- expected(Expected)
- validation(Validation)

4.2141 today/3

- datep

4.2142 tolerance_equal/4

- lgtunit
- number

4.2143 tool/1

- help

4.2144 tool_call/3

- mcp_tool_protocol

4.2145 tool_call/4

- mcp_tool_protocol

4.2146 tools/0

- help

4.2147 tools/1

- mcp_tool_protocol

4.2148 top/3

- heap_protocol

4.2149 top_next/5

- heap_protocol

4.2150 topological_sort/2

- directed_graph_protocol

4.2151 topological_sort/3

- unweighted_directed_graph(Dictionary)

4.2152 touch/3

- memcached

4.2153 trace/0

- debuggerp

4.2154 trace_event/2

- logtalk

4.2155 transaction/2

- transaction_dataset_protocol

4.2156 transform/3

- dimension_reducer_protocol

4.2157 transform/4

- crs_projections_protocol

4.2158 transitive_closure/2

- directed_graph_protocol

4.2159 transitive_reduction/2

- unweighted_directed_graph(Dictionary)

4.2160 transpose/2

- directed_graph_protocol
- pairs

4.2161 transpose_matrix/2

- linear_algebra_protocol

4.2162 traverse/3

- either
- maybe
- validated

4.2163 triangular/4

- sampling_protocol

4.2164 trim/2

- string(Representation)

4.2165 trim/3

- string(Representation)

4.2166 trim_left/2

- string(Representation)

4.2167 trim_left/3

- string(Representation)

4.2168 trim_right/2

- string(Representation)

4.2169 trim_right/3

- string(Representation)

4.2170 trimmed_mean/3

- statisticsp

4.2171 true/1

- java_utils_protocol

4.2172 tt_minus_tai/2

- time_scales_data

4.2173 ttl/3

- redis

4.2174 tx_commit/1

- amqp

4.2175 tx_rollback/1

- amqp

4.2176 tx_select/1

- amqp

4.2177 type/1

- command_line_option
- type

4.2178 type/2

- ccsds_packets(SecondaryHeaderLength)

4.2179 type/3

- redis

4.2180 tzdb_version/1

- tzif_zone_ids

4.2181 unexpected/1

- expected(Expected)

4.2182 unexpecteds/2

- either

4.2183 uniform/1

- sampling_protocol

4.2184 uniform/3

- sampling_protocol

4.2185 uninstall/0

- packs

4.2186 uninstall/1

- packs

4.2187 uninstall/2

- packs

4.2188 union/3

- interval_relation_set_protocol
- setp
- two3tree
- unweighted_directed_graph(Dictionary)

4.2189 union/4

- setp
- union_find_protocol

4.2190 union_all/3

- union_find_protocol

4.2191 universal/1

- interval_relation_set_protocol

4.2192 unix_seconds/2

- ccsds_time_code_protocol

4.2193 unix_to_date_time/2

- datep

4.2194 unpin/0

- packs_common

4.2195 unpin/1

- packs_common

4.2196 unsubscribe/3

- stomp

4.2197 unzip/2

- zipperp

4.2198 update/0

- doclet
- packs
- registries

4.2199 update/1

- observer
- packs
- registries

4.2200 update/2

- packs
- registries

4.2201 update/3

- datalog_protocol
- dictionaryp
- packs

4.2202 update/4

- dictionaryp

4.2203 update/5

- dictionaryp

4.2204 update_fecf/2

- ccsds_aos_frames(FrameLength,InsertZoneLength,HasOCF,HasFECF)
- ccsds_frame_protocol
- ccsds_frames
- ccsds_tc_frames(FrameLength,SegmentHeaderLength,HasFECF)
- ccsds_tm_frames(FrameLength,SecondaryHeaderLength,HasFECF)

4.2205 update_in/4

- nested_dictionary_protocol

4.2206 update_in/5

- nested_dictionary_protocol

4.2207 upper_triangular_part/2

- linear_algebra_protocol

4.2208 upper_triangular_part/3

- linear_algebra_protocol

4.2209 user_data/2

- ccsds_packets(SecondaryHeaderLength)

4.2210 utc_date_time_to_instant/2

- time_scales_protocol

4.2211 utc_to_local/3

- datep

4.2212 utc_to_local_tz/3

- dates_tz_protocol

4.2213 utm_crs/2

- crs_projections_protocol

4.2214 utm_zone/2

- crs_projections_protocol

4.2215 uuid_max/1

- uuid_protocol

4.2216 uuid_nil/1

- uuid_protocol

4.2217 uuid_null/1

- uuid_protocol

4.2218 uuid_v1/2

- uuid_protocol

4.2219 uuid_v3/3

- uuid_protocol

4.2220 uuid_v4/1

- uuid_protocol

4.2221 uuid_v5/3

- uuid_protocol

4.2222 uuid_v7/1

- uuid_protocol

4.2223 valid/0

- command_line_option

4.2224 valid/1

- ccsds_frame_protocol
- ccsds_frames
- ccsds_packet_services
- ccsds_time_code_protocol
- ccsds_time_codes
- ieee_754_protocol
- interval_protocol
- statisticsp
- termp
- url(Representation)
- validation(Validation)
- varlistp

4.2225 valid/2

- type

4.2226 valid/3

- datep
- timep

4.2227 valid_anomaly_detector/1

- anomaly_detector_protocol

4.2228 valid_channel_reassembly_state/1

- ccsds_packet_services

4.2229 valid_classifier/1

- classifier_protocol

4.2230 valid_clusterer/1

- clusterer_protocol

4.2231 valid_conversion/3

- time_scales_protocol

4.2232 valid_coordinate/1

- geospatial_protocol

4.2233 valid_coordinate/2

- crs_projections_protocol

4.2234 valid_crs/1

- crs_projections_protocol

4.2235 valid_date/3

- iso8601

4.2236 valid_date_time/1

- datep

4.2237 valid_descriptor/1

- ccsds_time_fields

4.2238 valid_dimension_reducer/1

- dimension_reducer_protocol

4.2239 valid_discontinuity_policy/1

- ccsds_link_profiles
- ccsds_packet_services
- ccsds_tc_services

4.2240 valid_geohash/1

- geohash_protocol

4.2241 valid_instant/1

- time_scales_protocol

4.2242 valid_map_dispatch/1

- ccsds_tc_services

4.2243 valid_option/1

- options_protocol

4.2244 valid_options/1

- options_protocol

4.2245 valid_pattern_miner/1

- pattern_miner_protocol

4.2246 valid_profile/1

- ccsds_link_profiles

4.2247 valid_ranker/1

- ranker_protocol

4.2248 valid_reassembled_segment/1

- ccsds_tc_services

4.2249 valid_reassembly_state/1

- ccsds_link_profiles
- ccsds_packet_services
- ccsds_tc_services

4.2250 valid_regressor/1

- regressor_protocol

4.2251 valid_scale/1

- time_scales_protocol

4.2252 valid_segment/1

- ccsds_tc_services

4.2253 valid_until_date/1

- application_protocol

4.2254 validate/1

- flags_validator
- geojson_protocol
- wkt_wkb_protocol

4.2255 validate/2

- geojson_protocol
- json_schema_protocol
- wkt_wkb_protocol

4.2256 validate/3

- json_schema_protocol

4.2257 validate_grouped_dataset/1

- ranking_dataset_common

4.2258 validate_grouped_dataset/2

- ranking_dataset_common

4.2259 validate_pairwise_dataset/1

- ranking_dataset_common

4.2260 validate_pairwise_dataset/2

- ranking_dataset_common

4.2261 validate_pairwise_measurement_dataset/1

- ranking_dataset_common

4.2262 validate_pairwise_measurement_dataset/2

- ranking_dataset_common

4.2263 validate_temporal_pairwise_dataset/1

- ranking_dataset_common

4.2264 validate_temporal_pairwise_dataset/2

- ranking_dataset_common

4.2265 valids/2

- validated

4.2266 value/1

- counter

4.2267 value/3

- pairs

4.2268 value_reference/2

- java_utils_protocol

4.2269 values/2

- dictionaryp
- pairs

4.2270 variables/2

- termp

4.2271 variance/2

- statisticsp

4.2272 variant/2

- lgtunit
- termp

4.2273 varnumbers/2

- termp

4.2274 varnumbers/3

- termp

4.2275 vector_norm/3

- linear_algebra_protocol

4.2276 vector_value/3

- linear_algebra_protocol

4.2277 verify_commands_availability/0

- packs_common

4.2278 verify_fecf/1

- ccsds_aos_frames(FrameLength,InsertZoneLength,HasOCF,HasFECF)
- ccsds_frame_protocol
- ccsds_frames
- ccsds_tc_frames(FrameLength,SegmentHeaderLength,HasFECF)
- ccsds_tm_frames(FrameLength,SecondaryHeaderLength,HasFECF)

4.2279 version/1

- application_protocol

4.2280 version/2

- ccsds_frame_protocol
- ccsds_frames
- ccsds_packets(SecondaryHeaderLength)
- memcached

4.2281 version/6

- pack_protocol

4.2282 versions/3

- packs

4.2283 vertices/2

- graph_protocol

4.2284 vincenty_distance/3

- geospatial_protocol

4.2285 virtual_channel_frame_count/2

- ccsds_aos_frames(FrameLength,InsertZoneLength,HasOCF,HasFECF)
- ccsds_tm_frames(FrameLength,SecondaryHeaderLength,HasFECF)

4.2286 virtual_channel_id/2

- ccsds_frame_protocol
- ccsds_frames

4.2287 void/1

- java_utils_protocol

4.2288 void_element/1

- html

4.2289 von_mises/3

- sampling_protocol

4.2290 wait/2

- process

4.2291 wald/3

- sampling_protocol

4.2292 wall_time/1

- osp

4.2293 warning/1

- linter_reporter

4.2294 warnings/1

- linter_reporter

4.2295 weakly_connected_components/2

- directed_graph_protocol

4.2296 week_of_year_iso/2

- datep

4.2297 weekday/2

- datep

4.2298 weibull/3

- sampling_protocol

4.2299 weight_variances/2

- bayesian_ridge_regression

4.2300 weighted_mean/3

- statisticsp

4.2301 welcome/0

- shell

4.2302 when/2

- coroutining

4.2303 whiledo/2

- loopp

4.2304 white_space//0

- blank_grammars(Format)

4.2305 white_spaces//0

- blank_grammars(Format)

4.2306 with_connection/1

- amqp_pool

4.2307 with_output_to/2

- term_io_protocol

4.2308 within_distance/4

- geospatial_protocol

4.2309 without//2

- sequence_grammars

4.2310 wneighbors/3

- weighted_graph_protocol

4.2311 word32_hex/2

- hash_common_32

4.2312 word64_hex/2

- hash_common_64

4.2313 working_directory/1

- osp

4.2314 write_file/3

- csv_protocol
- tsv_protocol

4.2315 write_framed_message/2

- json_rpc

4.2316 write_max_depth/1

- debuggerp

4.2317 write_message/2

- json_rpc

4.2318 write_stream/3

- csv_protocol
- tsv_protocol

4.2319 write_term_to_atom/3

- term_io_protocol

4.2320 write_term_to_chars/3

- term_io_protocol

4.2321 write_term_to_chars/4

- term_io_protocol

4.2322 write_term_to_codes/3

- term_io_protocol

4.2323 write_term_to_codes/4

- term_io_protocol

4.2324 write_to_atom/2

- term_io_protocol

4.2325 write_to_chars/2

- term_io_protocol

4.2326 write_to_codes/2

- term_io_protocol

4.2327 xor64/3

- hash_common_64

4.2328 year/2

- ccsds_ccs(CalendarVariant,FractionOctets)

4.2329 z_normalization/2

- statisticsp

4.2330 zadd/5

- redis

4.2331 zcard/3

- redis

4.2332 zero_or_more//0

- sequence_grammars

4.2333 zero_or_more//1

- sequence_grammars

4.2334 zero_or_more//2

- sequence_grammars

4.2335 zip/2

- zipperp

4.2336 zip/3

- expected(Expected)
- optional(Optional)
- validation(Validation)
- zipperp

4.2337 zip_at_index/4

- zlist

4.2338 zone/3

- tzif_protocol

4.2339 zone_id_kind/2

- tzif_zone_ids

4.2340 zones/1

- tzif_protocol

4.2341 zones/2

- tzif_protocol

4.2342 zrange/5

- redis

4.2343 zrank/4

- redis

4.2344 zrem/4

- redis

4.2345 zscore/4

- redis

CHAPTER

FIVE

INDICES AND TABLES

- genindex
- search

Generated on Thu May 14 00:23:41 WEST 2026

INDEX

Symbols

(/)/2, 1014
(//)/2, 1015
(<)/2, 2271
(<=)/2, 84, 1381
(=:=)/2, 2272
(=<)/2, 2271
(=>)/2, 84, 1381
(=\=)/2, 2272
(>)/2, 2271
(>=)/2, 2272
=~= / 2, 1320, 2318

A

a_star_interpreter(W), 2441
abbreviation/2, 2375
abbreviation/3, 2375
abbreviation/4, 2375
abort_transaction/3, 2119
absolute_file_name/2, 1669
absorb/2, 948
acc_info/5, 714
acc_info/7, 714
accept_hook_/1, 1400
accumulate_feature_statistics/5, 1999
accumulate_item_scores/3, 1935
accumulate_ridge_system/5, 2002
ack/3, 2117
acquire/1, 26
activate_debug_handler/1, 394
activate_monitor/0, 730
active_collection_mode_/1, 1280
active_debug_handler/1, 393
active_debug_handler_/1, 402
active_diagnostic_/1, 1279
active_feature_count/2, 2000
active_preflight_issue_/1, 1279
adaptive_boosting_classifier, 1
add/1, 1745
add/2, 1744
add/3, 1743, 2276
add/5, 1496
add32/3, 954
add32/4, 954
add32/5, 954
add64/3, 960
add_directive_/2, 2489
add_directive_/3, 2489
add_directive_before_entity_/2, 2488
add_duration/3, 464
add_edge/4, 910
add_edge/5, 923
add_edges/3, 889
add_library_documentation_url/4, 629
add_link_options/3, 574
add_matrices/3, 1419
add_node_zoom_option/4, 575
add_outer_product/4, 2002
add_rule/3, 453
add_scaled_outer_product/4, 1440
add_scaled_vector/4, 1411
add_squared_vector/3, 2000
add_type/2, 1536
add_type/3, 1536
add_vectors/3, 1410
add_vertex/3, 887
add_vertices/3, 888
addDependent/1, 524
adjacent/3, 797
adjacent_bits/4, 804
adjacent_int/4, 801
after/2, 470, 1119
after/3, 404
after_event_registry, 719
agglomerative_clusterer, 3
aggregate_body_predicate/2, 448
alias/1, 227
all/0, 317, 488, 1292
all/1, 316, 488, 1292
all_files/0, 564, 588
all_files/1, 564, 588
all_libraries/0, 558, 582
all_libraries/1, 558, 582
all_pairs_min_paths/2, 894

all_pairs_min_predecessors/2, 894
all_score/1, 319
along_track_distance/4, 828
alternating_subsequence/2, 2161
alternating_subsequences/2, 2161
amqp, 4
amqp_pool, 24
ancestor/1, 1030
ancestor/4, 327
ancestors/1, 1030
and64/3, 961
anomaly_dataset_protocol, 29
anomaly_detector_common, 32
anomaly_detector_diagnostics_data/2, 34
anomaly_detector_export_template/2, 34
anomaly_detector_options/2, 41
anomaly_detector_protocol, 37
anomaly_detector_term_template/2, 34
apid/2, 184
apis/0, 1012
apis/1, 1012
append/2, 2289
append/3, 1497, 1832, 2289, 2354
append/4, 1960
application_/1, 1485
application_capabilities_/1, 1487
application_common, 44
application_protocol, 46
apply/2, 2511
apply/4, 663
apply_float_sign/3, 1085
approximately_equal/2, 1318, 2317
approximately_equal/3, 1318, 2317
apriori_pattern_miner, 57
arbitrary, 58
arbitrary/1, 61
arbitrary/2, 61
archive/1, 1756
arithmetic_mean/2, 2095
arithmetic_mean/5, 2087
arithmetic_operator_replacement(Entity,Predicate,ClauseIndex,Occurrence,PrintMutation), 1552
arrangement/3, 68
arrangement/4, 69
arrangement_index/4, 71
arrangement_index/5, 72
arrangements, 65
arrangements/3, 68
arrangements/4, 68
arrangements_protocol, 66
array_list/2, 1196
array_to_list/2, 1196
array_to_terms/2, 1195
array_to_terms/3, 1195
articulation_points/2, 900
as_curly_bracketed/2, 654, 1624
as_curly_bracketed_impl/2, 674
as_curly_bracketed_impl/3, 674
as_deque/2, 532
as_dictionary/2, 654
as_difflist/2, 2284
as_heap/2, 1000
as_list/2, 532, 654, 1000, 1833, 2039, 2276
as_nested_dictionary/2, 1624
as_set/2, 2038
ask_question/5, 391
assert_fact/1, 454
assertion/1, 79, 1313
assertion/2, 79, 1313
assertions, 76
assertions(Mode), 77
assertions_messages, 80
assignable/1, 83
assignable/2, 84
assignvars, 81
assignvarsp, 82
atom, 2255
atom_string/2, 2133
atomic, 2257
atomics_to_string/2, 2137
atomics_to_string/3, 2137
attribute_occurrences/4, 1994
attribute_value/3, 293, 688
attribute_values/2, 31, 279, 304, 702, 1977
augment_rows/3, 2003
automation_report, 1293
auxiliary_predicate_counter_/1, 1351
available/0, 1692
available/1, 28, 1692
available/2, 1691
average/2, 2324
average_deviation/3, 2097
avltree, 648
avro, 86

B

back_substitution/2, 2006
backend_adapter_hook, 1046
backend_random, 1835
backward_substitution/3, 1443
base32, 88
base58, 90
base64, 92
base64url, 94
base85, 96
base_/2, 791
base_dimension_reducer_diagnostics/6, 689
base_regressor_diagnostics/6, 1985

baseline_coverage_/4, 1575
baseline_training_examples/3, 36
basic_ack/3, 16
basic_cancel/3, 15
basic_consume/3, 14
basic_get/3, 15
basic_nack/3, 16
basic_publish/4, 14
basic_qos/2, 17
basic_recover/2, 18
basic_reject/3, 17
basis_initial_vectors/3, 691
basis_vector/3, 691, 1413
bayesian_ridge_regression, 98
bbox_contains/2, 834
bbox_contains_polygon/2, 836
bbox_expand/3, 836
bbox_from_coordinates/2, 837
bbox_intersects/2, 835
bbox_intersects_polygon/2, 836
bbox_intersects_polyline/2, 837
bbox_overlaps/2, 817, 835
bbox_union/3, 835
before/2, 469, 1118
before/3, 404
before_event_registry, 720
begin/0, 454
begin_transaction/3, 2118
bell/2, 2313
bench_goal/1, 2454
benchmark/2, 1316
benchmark/3, 1317
benchmark/4, 1317
benchmark_generators, 2443
benchmark_reified/3, 1316
bernoulli/2, 1862
best_first, 2444
beta/3, 1862
between/3, 1849, 2280
between/4, 2278
bfs_interpreter, 2446
bic/5, 1175
big_endian_word32/2, 956
binary_file_assertion/3, 1347
binary_heap(Order), 991
binary_heap_max, 992
binary_heap_min, 994
binary_input_assertion/2, 1332
binary_input_assertion/3, 1331
binary_output_assertion/2, 1342
binary_output_assertion/3, 1341
binary_output_contents/1, 1343
binary_output_contents/2, 1342
binomial/3, 1862, 2315
bintree, 649
bit//1, 865
bits//1, 865
bits_fields/5, 1079
bits_finite_binary_rational/4, 1080
bits_nan_kind/2, 1080
bits_nan_payload/2, 1080
blank//0, 860
blank_grammars(Format), 856
blanks//0, 860
block_size/1, 965
body_goal_negation(Entity,Predicate,ClauseIndex,Occurrence,Printl
1553
body_pred/1, 1526
body_pred_call/2, 1528
borda_ranker, 101
bounding_box/2, 796
bounding_box/3, 841
bounding_box_bits/3, 804
bounding_box_int/3, 801
bradley_terry_ranker, 103
branch/2, 846
breadth_first_order/3, 890
bridges/2, 901
build_auto_search_index/3, 306
build_dataset_model/5, 1896
build_direct_strengths/5, 1880
build_encoders/4, 293, 688
build_grid_index/3, 307
build_linear_encoders/4, 269, 1997
build_metric_tree/3, 306
build_ridge_system/5, 2002
build_score_ranker/5, 1934
built_date/1, 51
built_in_directive/4, 1016
built_in_flag/2, 766
built_in_method/4, 1017
built_in_non_terminal/4, 1018
built_in_predicate/4, 1017
bup_interpreter, 2447
bypass_flag/2, 140
byte_count/1, 1092, 1097
bytes_dword/6, 224
bytes_hex/2, 953
bytes_tail/3, 1083
bytes_to_codes/2, 228
bytes_to_unsigned_integer/2, 1083
bytes_to_unsigned_integer/3, 1083
bytes_word/4, 224

C

c/1, 2451
c45_classifier, 105
cache/1, 2368

cache_source/1, 2370
cached_run_/5, 1279
cached_tzif/1, 2370
cached_tzif_/2, 2365
calendar_month/3, 1145
call_with_timeout/2, 2202
call_with_timeout/3, 2203
callable, 2258
caller_diagram, 544
caller_diagram(Format), 545
canonical_bic/2, 1175
canonical_iban/2, 1149
canonical_nan_bits/1, 1081
canonical_order_bytes/3, 1082
capabilities/1, 1489
captured_mutated_terms_/5, 1574
capturing_mutated_terms_/0, 1574
cartesian_product/2, 111
cartesian_product/3, 111
cartesian_products, 108
cartesian_products_protocol, 109
cartesian_tuple/2, 111
cartesian_tuple/3, 112
cartesian_tuple_index/3, 115
cartesian_tuple_index/4, 115
cas/6, 1498
cat/2, 1638
catalan/2, 2312
cbor, 119
cbor(StringRepresentation), 120
cc_metric, 309
ccsds_aos_frames(FrameLength,InsertZoneLength,HasOCF,HasFECF), 122
ccsds_ccs(CalendarVariant,FractionOctets), 201
ccsds_cds(DaySegmentOctets,SubmillisecondOctets,Epoch), 204
ccsds_cuc(CoarseOctets,FineOctets,Epoch), 207
ccsds_frame_protocol, 126
ccsds_frames, 132
ccsds_frames_types, 138
ccsds_link_profiles, 148
ccsds_packet_services, 158
ccsds_packetization, 173
ccsds_packets, 179
ccsds_packets(SecondaryHeaderLength), 180
ccsds_packets_types, 187
ccsds_tc_frames(FrameLength,SegmentHeaderLength,HasFECF), 139
ccsds_tc_services, 188
ccsds_time_code_protocol, 209
ccsds_time_codes, 214
ccsds_time_codes_types, 216
ccsds_time_fields, 217
ccsds_tm_frames(FrameLength,SecondaryHeaderLength,HasFECF), 143
cdata_generation//1, 2498
cell_dimensions/3, 797
center_gram_matrix/4, 1417
center_gram_vector/4, 1418
central_moment/3, 2104
central_moment_sum/7, 2089
change_directory/1, 1673
changed/0, 523
changed/1, 524
channel_close/1, 9
channel_close/3, 9
channel_open/3, 8
character, 2259
character_data_format/3, 2497
character_set, 221
character_set_protocol, 225
characterp, 2261
chebyshev_distance/3, 2327
chebyshev_norm/2, 2326
check/0, 373
check/1, 2343, 2362
check/2, 2350
check/3, 2350
check_anomaly_detector/1, 39
check_attribute_bindings/2, 293, 1992
check_attribute_declarations/1, 1992
check_binary_file/2, 1347
check_binary_input/1, 1331
check_binary_input/2, 1331
check_binary_output/1, 1341
check_binary_output/2, 1340
check_categorical_value/3, 1996
check_classifier/1, 274
check_cluster_count/2, 296
check_clusterer/1, 300
check_complete_examples/2, 268
check_complete_examples_nonvar/2, 269
check_component_count/3, 684
check_continuous_attributes/1, 291, 687
check_conversion/3, 2196
check_convert/4, 2198
check_declared_attribute_bindings/2, 1993
check_dimension_reducer/1, 698
check_encoded_attribute_bindings/2, 295
check_example_attributes/2, 292, 688
check_example_values/2, 292, 687
check_examples/2, 268, 1984
check_examples/3, 292
check_examples_non_empty/2, 35, 268, 291, 687
check_instant/1, 2195
check_item_domain/1, 1779
check_offset/3, 2200

check_option/1, 1657
check_options/1, 1658
check_pattern_miner/1, 1783
check_ranker/1, 1910
check_regressor/1, 2010
check_sequences/4, 2030
check_text_file/2, 1345
check_text_file/3, 1345
check_text_input/1, 1327
check_text_input/2, 1327
check_text_output/1, 1335
check_text_output/2, 1335
check_text_output/3, 1334
check_transactions/4, 776
check_undeclared_attribute_bindings/2, 1993
checksum/2, 1633
chi_squared/2, 1860
children/2, 800
cholesky_decomposition/2, 1441
chr_is/2, 2233
chr_next_state/1, 2236
chr_no_spy/1, 2233
chr_nospy/0, 2233
chr_notrace/0, 2233
chr_option/2, 2234
chr_option_allow_deep_guards/0, 2235
chr_option_optimization_level/1, 2234
chr_option_print_trace/0, 2234
chr_option_show_history/0, 2235
chr_option_show_id/0, 2235
chr_option_show_stack/0, 2235
chr_option_show_store/0, 2235
chr_option_trace_interactive/0, 2234
chr_rule_/1, 2236
chr_spy/1, 2233
chr_spy_point/1, 2236
chr_trace/0, 2233
circular_uniform_cartesian/3, 1867
circular_uniform_polar/3, 1867
class/1, 31, 279, 705, 1023
class_/5, 1296
class_hierarchy, 1021
class_hierarchyp, 1022
class_values/1, 31, 280, 705
classes/1, 1023
classifier_common, 262
classifier_diagnostics_data/2, 264
classifier_export_template/4, 264
classifier_options/2, 276
classifier_protocol, 272
classifier_term_template/2, 264
classify/2, 1088
clause/5, 1814
clause_/5, 1815
clause_breakpoint_/2, 502
clause_location/6, 1813
clause_location_/6, 1814
clause_mutator_protocol, 1555
clauses_/1, 1557
clauses_reordering(Entity,Predicate,ClauseIndex,Occurrence,PrintMu
1556
clean/0, 1717, 1750
clean/1, 1716, 1749
clean/2, 1716
clean_binary_input/0, 1332
clean_binary_output/0, 1343
clean_directory/1, 1348
clean_file/1, 1347
clean_text_input/0, 1329
clean_text_output/0, 1339
cleanup/0, 1322
clear/0, 452
clear_cache/0, 2370
clear_dut1_override/0, 2184, 2193
clear_leap_seconds_override/0, 2182, 2191
client_connection_/3, 1399
client_connection_alias_/2, 1396
client_connection_input_/2, 1395
client_connection_output_/2, 1395
client_engine_/2, 1401
client_open/4, 2076
client_open/5, 2075
client_timeout_/1, 1396
clo_span_pattern_miner, 284
clockwise_polygon/2, 833
clone/1, 1375, 1755
clone/3, 655
clone/4, 655
clone_impl/3, 671
clone_impl2/4, 671
cloning, 1374
close/1, 7
close/2, 2079
close/3, 8
close_client/1, 1387
close_polygon/2, 832
closed_input_stream/2, 1348
closed_output_stream/2, 1349
cluster/3, 299
cluster_probabilities/3, 781
clusterer_common, 286
clusterer_diagnostics_data/2, 297
clusterer_options/2, 301
clusterer_protocol, 297
clustering_dataset_protocol, 303
coarse/2, 208
cobertura_report, 1294
code_metric, 310

code_metrics, 323
code_metrics_messages, 325
code_metrics_utilities, 326
codes_to_bytes/2, 227
coefficient_of_variation/2, 2099
cogc_metric, 331
collect/4, 679
collection_options_/1, 1452
colley_ranker, 359
combination/3, 364
combination/4, 364
combination_index/4, 367
combination_index/5, 367
combinations, 361
combinations/3, 363
combinations/4, 364
combinations_protocol, 362
command/2, 1734
command_line_arguments/1, 1683
command_line_option, 372
command_line_options, 376
commit/0, 454
commit_author/2, 846
commit_date/2, 847
commit_hash/2, 847
commit_hash_abbreviated/2, 847
commit_log/3, 848
commit_message/2, 848
commit_transaction/3, 2119
common_prefix/3, 800
common_subsequence/3, 2157
common_subsequences/3, 2156
compact/3, 1211
compare_date_time/3, 470
comparingp, 2270
compile_aux_clauses/1, 399
compile_predicate_heads/4, 400
compile_predicate_indicators/3, 400
compiled_pred_call/2, 1528
complement/2, 910
completion/2, 1016
completions/2, 1016
component_iteration_diagnostics/4, 690
compose/3, 905, 1102, 1129
compound, 2273
compress/2, 799
compress_features/3, 2001
compress_rows/3, 2001
compress_vector/3, 2001
compute_ranks/3, 2090
condition/0, 1322
conditional_breakpoint_/3, 504
condorcet_victory_common, 1879
confirm_select/1, 20
connect/1, 1958
connect/2, 1494
connect/3, 1494, 1958
connect/4, 7, 2114
connected_components/2, 914, 929
connection_alive/1, 8, 2115
consensus_ranking/2, 1248
consensus_score/2, 1249
consistent/1, 1111
console/1, 1959
constraints/2, 1115
containment_edge_/2, 552
contains/2, 1121
context_breakpoint_/4, 503
context_summaries/2, 2218
continuation_byte/1, 222
continuous_stats/5, 1989
contradiction/2, 1109
contradiction_explanations/2, 1110
control//0, 861
control_command_flag/2, 140
control_construct/4, 1017
controls//0, 861
converse/2, 1102, 1129
convert/4, 2198
convert_zones/4, 479
convert_zones_with_resolution/5, 479
cooling_schedule/3, 2059
coordinates_bounding_box/2, 830
copeland_ranker, 380
copy_file/2, 1678
core_messages, 382
coroutining, 406
correlation/3, 2103
cosine_similarity/3, 2128
count_arrangements/3, 72
count_cartesian_product/2, 113
count_combinations/3, 368
count_derangements/2, 540
count_distinct_arrangements/3, 72
count_distinct_cartesian_product/2, 114
count_distinct_combinations/3, 368
count_distinct_derangements/2, 540
count_distinct_multisets/3, 1548
count_distinct_partitions/2, 1768
count_distinct_partitions/3, 1771
count_distinct_permutations/2, 1801
count_distinct_subsequences/3, 2157
count_frequencies/4, 2090
count_items/4, 777, 2031
count_less_equal/6, 2091
count_multisets/3, 1548
count_partial_derangements/3, 540
count_partitions/2, 1765

count_partitions/3, 1768
count_permutations/2, 1801
count_subsequences/2, 2163
counter, 2448
counter/2, 1377, 1609
counter_/2, 787, 1379, 1610
counterclockwise_polygon/2, 833
counters, 1376
country/4, 1154, 1157
country_spec/3, 1151
coupling_metric, 333
covariance/3, 2102
covariance_matrix/2, 1440
cover/1, 1309
coverage_clause_mutator/0, 1587
coverage_entry_/4, 1593
coverage_file_/1, 1593
coverage_report, 1298
covered_/4, 1354
covering/3, 798
covering/4, 798
cpu_time/1, 474, 1681
crc32_non_reflected(Polynomial,Initial,FinalXor,AppendLength), 932
crc32_reflected(Polynomial), 933
crc32b, 934
crc32bzip2, 936
crc32c, 937
crc32mpeg2, 938
crc32posix, 940
crc32q, 941
create/3, 1820
create_binary_file/2, 1344
create_text_file/2, 1344
create_text_file/3, 1344
creators/1, 50
cross_deviation_sum/6, 2088
cross_track_distance/4, 828
crs_dimensions/2, 414
crs_epsg/2, 414
crs_kind/2, 413
crs_name/2, 413
crs_projections, 409
crs_projections_protocol, 411
crs_units/2, 414
csv, 419
csv(Header,Separator,IgnoreQuotes), 420
csv(Header,Separator,IgnoreQuotes,Comments), 422
csv_guess_questions, 423
csv_protocol, 424
ctrf_output, 1300
ctrf_report, 1301
cuid2, 435
cuid2(Representation,Size,Alphabet), 436
cuid2_protocol, 438
currency/5, 1159, 1161
current/2, 2508
current_entity/1, 328
current_host/1, 2080
current_predicate_clause_index_/2, 1582
current_predicate_directive_index_/2, 1583
current_prog/1, 2234
current_scope_directive_index_/2, 1583
current_uses_directive_index_/2, 1584
cusum_anomaly_detector, 440
cut/3, 1020
cycle/2, 879, 900
cycle_edge_/2, 577
cytoscapejs_graph_language, 548

D

d2_graph_language, 552
damerau_levenshtein/3, 2123
data/0, 1811
data/1, 1812
data/2, 1633, 1812
data_field/2, 131, 135
data_length/2, 185
databasep, 2452
datalog, 441
datalog_protocol, 451
dataset_attributes/2, 35, 267, 288, 686, 1983
dataset_examples/2, 267, 1983
dataset_protocol, 278
date, 459
date/4, 1134
date/5, 1134
date/6, 1135
date/7, 1137
date_string/3, 1138
date_time/7, 1681
date_time_string/3, 1140
date_time_to_unix/2, 464
datep, 460
dates_tz, 475
dates_tz_protocol, 476
day/2, 203
day_of_year/2, 202, 467
day_of_year_date/3, 467
daylight_saving_time/2, 2374
daylight_saving_time/3, 2374
daylight_saving_time/4, 2374
days/2, 205
days_in_month/3, 463
dbscan_clusterer, 480
deactivate_debug_handler/0, 394
dead_code_scanner, 482
dead_code_scanner_messages, 490

debug/0, 509
debug_expansion(Mode), 2454
debug_handler/1, 393
debug_handler/3, 395
debug_messages, 492
debugger, 496
debugger_messages, 506
debuggerp, 507
debugging/0, 510
debugging/1, 510
debugging_/0, 498
decide/1, 759
decide/2, 759
declared_attribute_names/2, 1994
declares_predicate/2, 328
decode/2, 796, 1232
decode_bits/3, 803
decode_exception/2, 1198
decode_exception/3, 1198
decode_field_value/3, 24
decode_frame/2, 23
decode_int/3, 801
decode_url_spaces/2, 1737
decompile_predicate_heads/4, 401
decompile_predicate_indicators/4, 401
decompose_file_name/3, 1670
decompose_file_name/4, 1670
decr/3, 1962
decr/4, 1501
decrby/4, 1963
decrement_counter/1, 1377
default/1, 375
default_atom_mutations, 1599
default_compound_mutations, 1600
default_float_mutations, 1602
default_integer_mutations, 1603
default_list_mutations, 1604
default_option/1, 1659, 2484
default_options/1, 1659, 2485
default_workflow_hook, 1047
define_flag/1, 766
define_flag/2, 767
define_log_file/2, 1463
defined/4, 1742
defined_flag/6, 765
defined_flag_/6, 767
defines_predicate/2, 328
defines_predicate/3, 329
degree/3, 914, 928
del/3, 1963
del_monitors/0, 726
del_monitors/4, 726
del_spy_points/4, 731
delete/0, 1749
delete/1, 1748
delete/2, 1500, 1747
delete/3, 2039, 2289, 2355
delete/4, 656, 998
delete_all_after/2, 2514
delete_all_after_and_unzip/2, 2515
delete_all_before/2, 2514
delete_all_before_and_unzip/2, 2514
delete_and_next/2, 2513
delete_and_previous/2, 2513
delete_and_unzip/2, 2513
delete_directory/1, 1672
delete_directory_and_contents/1, 1673
delete_directory_contents/1, 1672
delete_edge/4, 910
delete_edge/5, 923
delete_edges/3, 889
delete_file/1, 1679
delete_impl/5, 672
delete_in/4, 1626
delete_matches/3, 2290
delete_max/4, 661
delete_min/4, 661
delete_vertex/3, 888
delete_vertices/3, 888
demodb, 2455
dependent_/1, 525
dependents/1, 524, 1722
dependents/2, 1722
dependents/3, 1721
depth/2, 2340
depth_first_order/3, 891
deque, 526
deque_protocol, 527
derangement/2, 536
derangement/3, 537
derangement_index/3, 539
derangement_index/4, 539
derangements, 533
derangements/2, 536
derangements/3, 536
derangements_protocol, 534
derive_aggregate_goals/1, 449
derive_aggregate_literal/2, 449
descendant/1, 1031
descendant_class/1, 1028
descendant_classes/1, 1028
descendant_instance/1, 1027
descendant_instances/1, 1027
descendants/1, 1032
describe/1, 1700, 1741
describe/2, 1699
description/1, 48, 1686, 1754
destination_point/4, 827

destroy/0, 26
determinant/2, 1430
deterministic/1, 1312
deterministic/2, 1312
dfs_interpreter, 2457
diagnostic/2, 40, 275, 300, 699, 1782, 1909, 2011, 2222
diagnostic/3, 2221
diagnostic_rule/5, 2221
diagnostic_rules/1, 2221
diagnostic_target/1, 2220
diagnostics/2, 40, 275, 299, 699, 1782, 1909, 2010, 2223
diagnostics/3, 2222
diagnostics_breakdown/2, 2218
diagnostics_preflight/2, 2224
diagnostics_preflight/3, 2224
diagnostics_summary/2, 2223
diagnostics_summary/3, 2223
diagnostics_tool/5, 2220
diagonal_matrix/2, 1424
diagonal_matrix/3, 1424
diagonal_matrix/4, 1425
diagram(Format), 554
diagram_caption/3, 566
diagram_description/1, 565
diagram_name_suffix/1, 565
diagrams, 578
diagrams(Format), 579
dictionaryp, 652
dif, 680
dif/1, 407, 681
dif/2, 407, 681
difference_norm/3, 1437
difflist, 2274
digest/2, 965
digest/4, 1039
digest/5, 1040
digest_size/1, 965
digit//1, 866
digits//1, 866
dimension_reducer_common, 682
dimension_reducer_data/3, 685
dimension_reducer_diagnostics_data/2, 685
dimension_reducer_options/2, 700
dimension_reducer_protocol, 696
dimension_reduction_dataset_protocol, 701
directed_graph_common, 874
directed_graph_protocol, 875
directive_mutator_protocol, 1558
directories/1, 1289, 2482
directories/2, 560, 584, 1288, 2481
directories/3, 560, 584
directory/1, 314, 485, 562, 586, 1290, 1566, 1730, 2481
directory/2, 313, 485, 562, 586, 1289, 1566, 1730, 2481
directory/3, 562, 586
directory_dependency_diagram, 589
directory_dependency_diagram(Format), 590
directory_diagram(Format), 592
directory_entity_/4, 1278
directory_exists/1, 1676
directory_files/2, 1675
directory_files/3, 1676
directory_load_diagram, 596
directory_load_diagram(Format), 597
directory_mutants/2, 1573
directory_mutants/3, 1573
directory_score/2, 318
dirichlet/2, 1866
disable/0, 1449
disable/1, 493
disable/2, 494
disable_logging/1, 1465
disconnect/1, 1494, 1958
disconnect/2, 2115
disjoint/2, 2039
disjoint_sets/2, 2405
dispatch_map_id/2, 191
dispatch_service_units/2, 192
dispatch_service_units_by_map/2, 200
dispatch_service_units_by_map/3, 200
distance/4, 308, 817, 825
distance/5, 817, 825
distinct_arrangement/3, 70
distinct_arrangement/4, 70
distinct_arrangement_index/4, 73
distinct_arrangements/3, 69
distinct_arrangements/4, 70
distinct_cartesian_product/2, 112
distinct_cartesian_product/3, 113
distinct_cartesian_tuple/2, 112
distinct_cartesian_tuple/3, 113
distinct_cartesian_tuple_index/3, 116
distinct_combination/3, 365
distinct_combination/4, 366
distinct_combination_index/4, 369
distinct_combinations/3, 365
distinct_combinations/4, 366
distinct_derangement/2, 537
distinct_derangement/3, 538
distinct_derangement_index/3, 541
distinct_derangements/2, 537
distinct_derangements/3, 538
distinct_multiset/3, 1545
distinct_multiset/4, 1546

distinct_multiset_index/4, 1549
distinct_multisets/3, 1545
distinct_multisets/4, 1546
distinct_partition/2, 1763
distinct_partition/3, 1764
distinct_partition/4, 1764
distinct_partition_index/3, 1769
distinct_partition_index/4, 1770
distinct_partition_index/5, 1770
distinct_partitions/2, 1763
distinct_partitions/3, 1763
distinct_partitions/4, 1764
distinct_permutation/2, 1796
distinct_permutation/3, 1797
distinct_permutation_index/3, 1802
distinct_permutations/2, 1796
distinct_permutations/3, 1797
distribution/1, 49
dit_metric, 335
djb2_32, 942
djb2_64, 944
doc_goal/1, 709
doc_metric, 336
doclet, 708
doctype/1, 1068
document/1, 2023
document/2, 2022
document_generation//2, 2497
dot//1, 869
dot_graph_language, 599
dot_product/3, 1412
double_metaphone/3, 2130
double_metaphone_match/2, 2131
dowhile/2, 1470
downcase_text/2, 2409
drop/3, 2309
dump_trace, 519
duration_between/3, 465
duration_string/2, 1141
during/2, 1121
dut1_entries/1, 2185, 2194
dut1_offset_at_utc_unix/3, 2186
dut1_source/1, 2184, 2193
dword_bytes/6, 224
dynamic_directive_/3, 2488

E

easter_day/3, 1145
eclat_pattern_miner, 710
edb_fact_/1, 443
edcg, 712
edge/3, 909
edge/4, 923
edge/5, 570
edge/6, 619
edge_/5, 577
edge_case/2, 63
edge_counter_/1, 551
edge_index_/1, 636
edges/2, 887
edit_similarity/3, 2125
edit_similarity/4, 2126
effective_maximum_pattern_length/3, 1779
effective_support_count/3, 1779
either, 735
either/3, 749
elastic_net_regression, 716
elicit_counter_/1, 1487
elicit_request/5, 1485
eliminate_rows/5, 2005
elo_ranker, 717
empty/1, 528, 657, 886, 999, 1126, 1624, 1641, 1829, 2040, 2290, 2355
empty_map/1, 2494
enable/0, 1449
enable/1, 493, 1449
enable/2, 494
enable_logging/1, 1465
enabled/1, 493
enabled/2, 494
enabled_/0, 1451
enabled_/1, 495
enabled_/2, 495
encode/2, 1232
encode/3, 795
encode_bits/3, 803
encode_frame/2, 23
encode_instance/3, 294, 693, 1989
encode_instance_checked/3, 1995
encode_int/3, 800
encode_linear_instance/3, 270
encoded_feature_count/2, 1988
encoder_attribute_names/2, 693, 1994
encoding_suffix/2, 1538
ensure_directory/1, 1677
ensure_file/1, 1679
ensure_non_zero/1, 2005
entailment_explanations/5, 1110
entails/4, 1108
entails/5, 1108
entity/1, 312, 484, 646, 1014, 1567
entity/2, 484, 646, 1567
entity_calls/3, 329
entity_defines_/2, 1815
entity_diagram, 601
entity_diagram(Format), 602
entity_file_/2, 1304
entity_info_pair_score_hook/3, 338

entity_info_score_hook/2, 338
entity_kind/2, 330
entity_mutants/2, 1571
entity_mutants/3, 1571
entity_predicate_breakpoint_/4, 503
entity_predicates_weights_hook/2, 338
entity_prefix/2, 399
entity_property/2, 330
entity_score/2, 317
entity_updates/3, 330
enumerate/2, 1852
environment_variable/2, 1680
epoch/1, 212
epoch/2, 220
epsg_crs/2, 415
epsilon/1, 1320
equal/2, 1122, 2040
equirectangular_inverse/4, 822
equirectangular_projection/4, 821
equivalent/2, 1115
erase/1, 1953
error/2, 1236
error_code/2, 1236
error_data/2, 1237
error_message/2, 1236
error_response/4, 1229
error_response/5, 1230
essentially_equal/3, 1319, 2318
estimate_temperature/1, 2053
estimate_temperature/2, 2054
euclidean_distance/3, 2326
euclidean_norm/2, 1412, 2325
evaluate/3, 1225
event_registry, 722
event_registryp, 723
ewma_anomaly_detector, 732
exactly_representable/1, 1096
example/2, 304, 703
example/3, 32, 280, 705, 707, 1978
example_attribute_values/2, 686
examples_to_linear_rows/3, 270
examples_to_rows/3, 294, 692, 1989
exception/4, 2496
exchange_bind/4, 11
exchange_declare/3, 10
exchange_delete/3, 10
exchange_unbind/4, 11
exclude/3, 1511
excluded/4, 1109
execution_context/7, 402
exists/3, 1964
expand/2, 805, 1211
expand_int/3, 805
expand_library_alias_paths, 734
expand_library_path/2, 395
expand_weights/3, 2007
expanding, 383
expected, 739
expected(Expected), 744
expected/1, 748
expecteds/2, 737
expire/4, 1965
explain//1, 2254
explain/2, 457
explicit_tracing_/0, 498
exponential/2, 1861
export/1, 2023
export/2, 2023
export_to_clauses/4, 43, 276, 301, 700, 1784, 1910, 2011
export_to_file/4, 43, 277, 301, 700, 1784, 1911, 2012
extension/1, 1035
extension_type/2, 1534
extension_type/3, 1534
extensions/1, 1036
external_predicate_/1, 647
external_reference/2, 52
extract_aos_packets/3, 167
extract_components/5, 692
extract_packets/3, 137
extract_packets/4, 154
extract_scores/2, 36
extract_tc_segment/2, 195
extract_tm_packets/3, 165

F

f/4, 2445
factorial/2, 2311
facts/1, 457
fail_insertion(Entity,Predicate,ClauseIndex,Occurrence,PrintMutatio 1559
failed_/3, 1353
failed_count_/1, 1596
failed_test_reason_//1, 1356
falling_factorial/3, 2312
false/1, 1192
fast_random, 1836
fast_random(Algorithm), 1838
fault/5, 2496
fcube, 757
fcube/0, 759
feature_penalty_profiles/5, 2000
fecf/2, 131, 135
fibonacci/2, 2311
fields/2, 1632
fields/5, 1089
file/1, 313, 485, 604, 1291, 2483
file/2, 313, 484, 603, 1291, 2483

file_being_advised_/4, 2490
file_dependency_diagram, 606
file_dependency_diagram(Format), 607
file_diagram(Format), 609
file_exists/1, 1677
file_footer/3, 618
file_header/3, 618
file_line_hit_count_/3, 505
file_load_diagram, 613
file_load_diagram(Format), 614
file_modification_time/2, 1677
file_path/2, 1323
file_path_components/2, 2408
file_permission/2, 1678
file_score/2, 318
file_size/2, 1678
file_to_bytes/2, 1942
file_to_bytes/3, 1943
file_to_chars/2, 1941
file_to_chars/3, 1941
file_to_codes/2, 1940
file_to_codes/3, 1941
file_to_terms/2, 1942
file_to_terms/3, 1942
file_type_extension/2, 397
file_url/2, 1324
files/1, 564, 588, 1290, 2483
files/2, 563, 587, 1290, 2482
files/3, 563, 587
filter/2, 1648
filter/3, 752, 2432
filter_external_file_extension/3, 574
filter_file_extension/3, 573
filter_patterns/3, 776, 2031
final_bearing/3, 826
find/4, 2404
find/5, 2405
findall_in_noblock/3, 1394
findall_in_noblock/4, 1394
findall_member/4, 1511
findall_member/5, 1511
findall_rd_noblock/3, 1394
findall_rd_noblock/4, 1393
fine/2, 208
finished_by/2, 1122
finishes/2, 1122
finite_binary_rational/4, 1089
fips202_fixed_hash(RateBytes,Suffix,OutputBytes), 945
fips202_hash(RateBytes,Suffix,OutputBytes), 946
fired_/3, 1354
first_header_pointer/2, 146
first_significant_component/2, 1439
first_significant_component/3, 1439
fisher/3, 1860
fit_linear_model/7, 1990
fit_ridge_model/7, 1990
fix_node2_left_underflow/6, 672
fix_node2_right_underflow/6, 672
fix_node3_left_underflow/9, 673
fix_node3_middle_underflow/9, 673
fix_node3_right_underflow/9, 673
fix_option/2, 1661
fix_options/2, 1661
flag_group_chk/1, 764
flag_groups/1, 764
flag_value_/2, 768
flags, 761
flags_validator, 769
flaky_/1, 1354
flat_map/2, 749, 1649, 2433
flatten/1, 756, 1653, 2440
flatten/2, 1211, 2290, 2355
flatten_goals//1, 2459
flatting, 2458
float, 2277
float//1, 868
flush_all/1, 1503
flush_all/2, 1503
fnv1a_32, 949
fnv1a_64, 950
fold_left/4, 1512
fold_left_1/3, 1513
fold_right/4, 1514
fold_right_1/3, 1514
fordownto/3, 1473
fordownto/4, 1473
fordownto/5, 1474
foreach/3, 1471
foreach/4, 1471
format, 771
format/1, 212
format/2, 215, 220, 772
format/3, 772
format_date_time/4, 466
format_entity_score//2, 320
format_object/1, 565
format_report/1, 1570
format_report/2, 1570
format_report/3, 1570
format_to_atom/3, 2174
format_to_chars/3, 2174
format_to_chars/4, 2175
format_to_codes/3, 2175
format_to_codes/4, 2175
formatted_iban/2, 1149
forto/3, 1472
forto/4, 1472

forto/5, 1472
forward/1, 386
forward/2, 2510
forward/3, 2511
forward_substitution/3, 1441
forwarding, 385
fp_growth_pattern_miner, 773
fractile/3, 2101
frame_body/2, 2121
frame_command/2, 2120
frame_header/3, 2120
frame_headers/2, 2120
frame_type/2, 129, 133
freeze/2, 408
frequency_distribution/2, 2105
frequent_pattern_mining_common, 774
frobenius_norm/2, 1420
from_expected/2, 2427
from_generator/2, 743, 1643, 2426
from_generator/3, 742, 1643, 2426
from_generator/4, 742, 2425
from_goal/2, 742, 1642, 2425
from_goal/3, 741, 1642, 2425
from_goal/4, 741, 2424
from_goal_or_throw/2, 1644
from_goal_or_throw/3, 1643
from_optional/3, 743, 2427
from_unix_seconds/2, 213
frozen/2, 408
full_device_path/1, 1675
func_test/3, 1526
functional/0, 1526
fund_currency/5, 1159, 1161

G

game/4, 1938
gamma/3, 1863
gat/4, 1502
gats/5, 1502
gaussian_mixture_clusterer, 780
gaussian_process_regression, 782
generate/1, 439, 1075, 1270, 1616, 2068, 2397
generate/2, 90, 92, 94, 96, 97, 121, 127, 182, 210, 811, 1067, 1094, 1204, 1210, 1218, 1223, 1506, 2020, 2215, 2230, 2397, 2408, 2475, 2502
generate/3, 87, 128, 182, 211, 219, 1095, 1824
generate/4, 88, 1824, 2020
generate/8, 2397
generate_all/2, 2503
generate_fragment/2, 1224
generate_frame/3, 151
generate_frames/3, 152
generate_idle_packet/4, 178
generated_predicate_/1, 1521
generating_/0, 1364
genint, 784
genint/2, 787
genint_core, 785
gensym, 788
gensym/2, 791
gensym_core, 789
geohash, 792
geohash_protocol, 793
geohash_to_int/2, 802
geojson, 807
geojson(ObjectRepresentation,PairRepresentation,StringRepresentati 808
geojson_protocol, 809
geojson_to_json/2, 814
geometric/2, 1861
geometric_mean/2, 2095
geospatial, 815
geospatial_protocol, 818
get/1, 1650
get/3, 1498, 1959
get/4, 1499
get_field/2, 1186
get_flag_value/2, 762
get_seed/1, 63, 1842
getrange/5, 1960
gets/4, 1499
gets/5, 1500
git, 844
git_object_identifier/1, 55
git_protocol, 845
glicko2_common, 1882
glicko2_periodic_ranker, 849
glicko2_ranker, 851
gnu/0, 759
goal_expansion/2, 384
gradient_boosting_classifier, 853
gradient_boosting_regression, 855
gram_matrix/2, 1416
grammar_rules_hook, 1049
graph_coloring/3, 900
graph_common, 881
graph_footer/5, 618
graph_header/5, 618
graph_language_protocol, 616
graph_language_registry, 620
graph_protocol, 883
graph_types, 896
ground/1, 2341
ground_entity_identifier/3, 573
ground_track/5, 2209
ground_track/6, 2209
group/1, 1928
group_by_key/2, 2337

group_consecutive_by_key/2, 2336
group_sorted_by_key/2, 2336
grouped_dataset_groups/2, 1921
grouped_dataset_item_relevance_scores/3, 1926
grouped_dataset_item_relevances/4, 1925
grouped_dataset_items/2, 1921
grouped_dataset_items/3, 1921
grouped_dataset_relevance_frequencies/2, 1925
grouped_dataset_summary/2, 1922
grouped_dataset_tie_blocks/4, 1926
grouped_strength_ranker_common, 1885
gsp_pattern_miner, 930
guess_all_extensions/2, 1535
guess_all_extensions/3, 1536
guess_arity/2, 434
guess_extension/2, 1534
guess_extension/3, 1535
guess_file_type/3, 1533
guess_file_type/4, 1533
guess_separator/2, 433
guess_type/3, 1532
guess_type/4, 1533
gumbel/3, 1866

H

halstead_metric, 341
halstead_metric(Stroud), 342
halt/0, 398
halt/1, 398
halt_hook/2, 399
hamming/3, 2124
hamming_distance/3, 2291
handbook/0, 1012
handle_root_underflow/2, 671
harmonic_mean/2, 2096
has_aggregate_rules/0, 448
has_cycle/1, 879, 899
has_negative_cycle/1, 925
has_path/3, 892
hash/2, 967
hash_common_32, 951
hash_common_64, 958
hash_digest_protocol, 964
hash_protocol, 966
haversine_distance/3, 822
hdbscan_clusterer, 989
hdel/4, 1967
head/2, 1830
head_arguments_mutation(Entity,Predicate,ClauseIndex,Occurrences,PrintMutation), 1560
head_arguments_reordering(Entity,Predicate,ClauseIndex,Occurrences,PrintMutation), 1562
head_pred/1, 1526
heap(Order), 995
heap_protocol, 996
heapp, 1001
help, 1010
help/0, 1012, 1727
help/1, 375
help/2, 378
help/3, 379
heuristic_expansion(Mode), 2459
hex_digest/4, 1039
hex_digest/5, 1040
hex_digit//1, 866
hex_digits//1, 867
hexists/4, 1968
hget/4, 1967
hgetall/3, 1967
hierarchical_clustering, 1018
hierarchyp, 1029
high_surrogate/1, 223
hkeys/3, 1968
hlen/3, 1969
hmac, 1036
hmac_protocol, 1038
hodge_rank, 1041
home/1, 1687, 1755
homepage/1, 49
hook_pipeline(Pipeline), 1043
hook_set(Set), 1044
hset/5, 1966
html, 1066
html5, 1069
hvals/3, 1968
hypergeometric/4, 1861

I

iban/4, 1149
ibk/3, 1526
ica_projection, 1071
id/2, 1234
idb_fact_/1, 444
iddfs_interpreter(Increment), 2461
identity_hook, 1050
identity_matrix/2, 1415
ids, 1073
ids(Representation,Bytes), 1074
ieee_754(Precision,ByteOrder,NaNRepresentation), 1076
ieee_754_common(Precision,ByteOrder), 1077
ieee_754_fields(Precision,ByteOrder), 1086
ieee_754_parts(Precision,ByteOrder), 1087
ieee_754_protocol, 1093
if_expected/1, 746
if_expected_or_else/2, 747
if_invalid/1, 2431

if_present/1, 1647
if_present_or_else/2, 1648
if_unexpected/1, 747
if_valid/1, 2430
if_valid_or_else/2, 2431
in/1, 1389
in/2, 1389
in_degree/3, 878
in_list/2, 1390
in_list/3, 1390
in_noblock/1, 1390
in_noblock/2, 1389
in_nodes_section_/0, 550
in_use/2, 29
include/3, 1510
included_caller_/1, 548
included_directory_/1, 595
included_entity_/1, 604
included_file_/1, 612
included_library_/2, 630
included_module_/1, 604
included_predicate_/1, 647
incr/3, 1962
incr/4, 1501
incrby/4, 1963
increase/1, 2450
increment/0, 2449
increment_counter/1, 1377
index_items/3, 1880, 1896, 1900
inheritance_diagram, 621
inheritance_diagram(Format), 623
init/0, 2471
init/2, 2148
init1/2, 2150
init_log_file/2, 1463
init_tail/2, 2152
init_tails/2, 2151
initial_bearing/3, 826
initial_channel_reassembly_state/1, 162
initial_reassembly_state/1, 153, 160, 194
initial_state/1, 174, 2057
initial_strengths/2, 1886, 1895
initial_temperature/1, 2059
initial_vectors/2, 691
initialize/1, 25
initialize_scores/2, 1935
initialized_/0, 1485
inits/2, 2147
inits1/2, 2149
inorder/2, 651
insert/3, 2040
insert/4, 655, 998
insert_after/3, 2512
insert_all/3, 998, 2041
insert_aos_packets/4, 167
insert_before/3, 2512
insert_empty/4, 675
insert_impl/4, 674
insert_in/4, 1626
insert_node2_2/4, 675
insert_node2_4/4, 675
insert_node3_4/4, 676
insert_node3_7/4, 676
insert_packets/4, 137
insert_packets/5, 155
insert_tc_segment/3, 195
insert_tm_packets/4, 165
insert_zone/2, 124
install/1, 1710
install/2, 1710
install/3, 1709
install/4, 1707
installed/0, 1694
installed/1, 1694
installed/3, 1693
installed/4, 1692
instance/1, 1024
instance/2, 1953
instances/1, 1024
instant_to_utc_date_time/2, 2198
int_to_geohash/3, 803
integer, 2279
integer//1, 867
integer_to_big_endian_bytes32/2, 957
integer_to_big_endian_bytes64/2, 963
integer_to_bytes/3, 1084
integer_to_little_endian_bytes32/2, 956
internal_error/2, 1232
internal_os_path/2, 1671
interpolate_great_circle/4, 827
interpolate_rhumb/4, 824
interpreterp, 2462
interquartile_range/2, 2102
intersect/2, 2041
intersection/2, 659
intersection/3, 659, 1128, 2041
intersection/4, 2042
intersection_impl/4, 676
interval, 1098
interval_algebra, 1100
interval_algebra_protocol, 1101
interval_constraint_network, 1103
interval_constraint_network_protocol, 1105
interval_protocol, 1116
interval_relation_set, 1123
interval_relation_set_protocol, 1125
interval_string/2, 1142
invalid/1, 2432

invalid_params/2, 1231
invalid_request/1, 1231
invalids/2, 2419
inverse_matrix/2, 1430
invert_from_cholesky/2, 1443
invocation_number_/1, 500
invoke/1, 1187
invoke/2, 1188
ip_grammars(Format), 862
ipv4//1, 863
ipv6//1, 863
iqr_anomaly_detector, 1130
is_absolute_file_name/1, 1669
is_acyclic/1, 878
is_alpha/1, 2263
is_alphanumeric/1, 2262
is_ascii/1, 2262
is_batch/1, 1234
is_bin_digit/1, 2263
is_bipartite/1, 895
is_clockwise_polygon/1, 832
is_complete/1, 895
is_connected/1, 914, 928
is_control/1, 2267
is_dec_digit/1, 2264
is_empty/0, 1646
is_end_of_line/1, 2268
is_error_response/1, 1234
is_expected/0, 746
is_false/1, 1193
is_hex_digit/1, 2264
is_invalid/0, 2430
is_layout/1, 2266
is_letter/1, 2263
is_lower_case/1, 2265
is_newline/1, 2268
is_node4/1, 677
is_notification/1, 1233
is_null/1, 1194
is_object/1, 1194
is_octal_digit/1, 2264
is_period/1, 2267
is_prefix_of/2, 2158
is_present/0, 1646
is_punctuation/1, 2267
is_quote/1, 2266
is_request/1, 1233
is_response/1, 1233
is_sparse/1, 895
is_subsequence_of/2, 2155
is_suffix_of/2, 2158
is_tree/1, 899
is_true/1, 1193
is_unexpected/0, 746
is_upper_case/1, 2265
is_valid/0, 2430
is_valid_polygon/1, 834
is_validator/1, 768
is_void/1, 1193
is_vowel/1, 2265
is_white_space/1, 2266
iso8601, 1132
iso_13616, 1146
iso_13616_protocol, 1147
iso_13616_registry, 1150
iso_3166, 1152
iso_3166_1, 1153
iso_3166_2, 1154
iso_3166_protocol, 1156
iso_4217, 1158
iso_4217_protocol, 1160
iso_639, 1162
iso_639_1, 1164
iso_639_2, 1165
iso_639_3, 1167
iso_639_5, 1168
iso_639_protocol, 1170
iso_8859_1, 229
iso_8859_10, 230
iso_8859_13, 231
iso_8859_14, 232
iso_8859_15, 233
iso_8859_16, 234
iso_8859_2, 236
iso_8859_3, 237
iso_8859_4, 238
iso_8859_9, 239
iso_9362, 1173
iso_9362_protocol, 1174
isolation_forest_anomaly_detector, 1177
issue_creator, 1179
item/1, 1890, 1892, 1938
item/2, 1928
item_denominator/5, 1899
items/1, 779, 2028
iterative_dimension_reducer_diagnostics/11, 689
iterator_element/2, 1197

J

jaccard_index/3, 2128
jaro/3, 2124
jaro_winkler/3, 2125
java, 1181
java(Reference), 1182
java(Reference,ReturnValue), 1183
java_access_protocol, 1185
java_hook, 1188
java_utils_protocol, 1189

join/3, 1830
join_all/3, 1830
join_aos_packet_zone/3, 166
join_packet_zone/4, 165
json, 1199
json(ObjectRepresentation,PairRepresentation,StringRepresentation), 1201
json(StringRepresentation), 1200
json_ld, 1205
json_ld(ObjectRepresentation,PairRepresentation,StringRepresentation), 1207
json_ld(StringRepresentation), 1206
json_ld_protocol, 1209
json_lines, 1212
json_lines(ObjectRepresentation,PairRepresentation,StringRepresentation), 1215
json_lines(StringRepresentation), 1214
json_lines_protocol, 1216
json_pointer, 1219
json_pointer(StringRepresentation), 1220
json_pointer_protocol, 1221
json_protocol, 1203
json_rpc, 1226
json_schema, 1239
json_schema(ObjectRepresentation,PairRepresentation,StringRepresentation), 1241
json_schema(StringRepresentation), 1240
json_schema_protocol, 1243
json_to_geojson/2, 814
jump/3, 1831
jump_all/3, 1831
jump_all_block/3, 1832
jump_to_invocation_number_/1, 501

K

k_distinct_subsequence/3, 2162
k_distinct_subsequences/3, 2162
k_permutation/3, 1798
k_permutation/4, 1798
k_permutations/3, 1797
k_permutations/4, 1798
kcenters_clusterer, 1245
kemeny_young_ranker, 1247
kernel_pca_projection, 1249
kernel_svm_classifier, 1251
key/2, 2334
keys/2, 662, 2334
keys/3, 1964
keys_values/3, 2334
keysort/2, 2291
kill/1, 1821
kill/2, 1821
kmeans_clusterer, 1253
kmedians_clusterer, 1254
kmedoids_clusterer, 1256
kmodes_clusterer, 1257
knn_classifier, 1259
knn_distance_anomaly_detector, 1261
knn_regression, 1263
known_attribute_values/3, 294, 692, 1992
known_zone_id/1, 2392
kprototypes_clusterer, 1265
ksuid, 1266
ksuid(Representation,Alphabet), 1267
ksuid_protocol, 1269
kurtosis/2, 2100

L

lambert_azimuthal_equal_area/3, 416
lambert_azimuthal_equal_area_inverse/3, 417
language/3, 1165, 1171
language/5, 1168, 1172
language_code/5, 1166, 1171
language_group/2, 1170, 1172
language_object/2, 621
lasso_regression, 1271
last/2, 2291, 2356
last_time_sequence_/2, 2064
lcom_metric, 3444
lcov_report, 1303
lda_classifier, 1272
lda_projection, 1274
leading_magnitude/2, 2004
leaf/1, 1031
leaf_class/1, 1026
leaf_classes/1, 1027
leaf_instance/1, 1026
leaf_instances/1, 1026
leap_effective_date/2, 2186
leap_offset_at_utc_unix/2, 2185
leap_second_date/2, 2191
leap_seconds_entries/1, 2183, 2192
leap_seconds_source/1, 2182, 2192
leap_year/1, 462, 1144
leaping_/1, 500
learn/0, 1530
learn/1, 1530
learn/2, 39, 273, 299, 697, 1524, 1908, 2009
learn/3, 39, 273, 298, 697, 1523, 1907, 2009, 2238
learn_seq/2, 1524
learn_with_timeout/4, 1524
leash/1, 511
leashing/1, 512
leashing_/1, 500
least_common_multiple/2, 2330
least_squares/3, 1431
leaves/1, 1031
leaves/2, 906

length/2, 530, 1832, 2292, 2356
levenshtein/3, 2123
lgtdoc, 1276
lgtdoc_messages, 1280
lgtdocp, 1281
lgtunit, 1305
lgtunit_messages, 1355
libraries/0, 1013
libraries/1, 557, 581, 1286
libraries/2, 557, 581, 1285
libraries/3, 557, 581
library/1, 315, 487, 559, 583, 1014, 1286, 1565
library/2, 315, 486, 559, 583, 1286, 1566
library_dependency_diagram, 624
library_dependency_diagram(Format), 625
library_diagram(Format), 627
library_entity_/4, 1277
library_load_diagram, 631
library_load_diagram(Format), 633
library_mutants/2, 1572
library_mutants/3, 1572
library_score/2, 317
license/1, 49, 1686
linda, 1382
linda/0, 1399
linda/1, 1398
linda_client, 1384
linda_client/1, 1386
linda_client/2, 1386
linda_server, 1397
linda_timeout/2, 1387
line_to_chars/2, 1946
line_to_chars/3, 1946
line_to_codes/2, 1947
line_to_codes/3, 1947
linear_algebra, 1405
linear_algebra_protocol, 1407
linear_encoders_feature_count/2, 270
linear_regression, 1444
linear_regression/4, 2332
linear_svm_classifier, 1446
lines_covered_/1, 1296
lines_metric, 345
lines_total_/1, 1297
lint/0, 1724, 1751
lint/1, 1724, 1751
lint/2, 1723
linter_reporter, 1447
linter_warning_sequence_/1, 1352
list, 2283
list(Type), 2285
list/0, 1741
list_to_array/2, 1196
listing, 1453
listing/0, 1454
listing/1, 1454
listp, 2286
literal_bucket/2, 450
little_endian_word32/2, 956
llen/3, 1971
load/1, 1537, 2368
load/2, 1537, 2368
load_config/1, 1595
load_coverage_config/1, 1593
load_dut1_override/1, 2183, 2193
load_leap_seconds_override/1, 2182, 2191
load_program/1, 452
load_registry/1, 1735
loaded_file/1, 395
loaded_file_property/2, 396, 638
loaded_files_topological_sort/1, 397
loaded_files_topological_sort/2, 397
loaded_pack/3, 1703
loaded_pack_dependency/6, 1706
loaded_pack_file/4, 1704
loader_file/1, 50
local_abbreviation/2, 2384
local_abbreviation/3, 2383
local_abbreviation/4, 2383
local_abbreviation_reified/2, 2390
local_abbreviation_reified/3, 2389
local_abbreviation_reified/4, 2389
local_abbreviation_with_resolution/3, 2385
local_abbreviation_with_resolution/4, 2385
local_abbreviation_with_resolution/5, 2384
local_daylight_saving_time/2, 2381
local_daylight_saving_time/3, 2381
local_daylight_saving_time/4, 2381
local_daylight_saving_time_reified/2, 2389
local_daylight_saving_time_reified/3, 2388
local_daylight_saving_time_reified/4, 2388
local_daylight_saving_time_with_resolution/3, 2383
local_daylight_saving_time_with_resolution/4, 2382
local_daylight_saving_time_with_resolution/5, 2382
local_offset/2, 2379
local_offset/3, 2379
local_offset/4, 2378
local_offset_reified/2, 2387
local_offset_reified/3, 2387
local_offset_reified/4, 2387
local_offset_with_resolution/3, 2380
local_offset_with_resolution/4, 2380
local_offset_with_resolution/5, 2379
local_tangent_plane/3, 416
local_tangent_plane_inverse/3, 416

local_time_type/2, 2377
local_time_type/3, 2376
local_time_type/4, 2376
local_time_type_reified/2, 2386
local_time_type_reified/3, 2386
local_time_type_reified/4, 2385
local_time_type_with_resolution/3, 2378
local_time_type_with_resolution/4, 2377
local_time_type_with_resolution/5, 2377
local_to_utc/3, 466
local_to_utc_tz/3, 478
local_to_utc_tz_with_resolution/4, 478
locate_directory/2, 572
locate_file/5, 573
locate_library/2, 572
locked_pairs/2, 1878
lof_anomaly_detector, 1456
log/3, 516
log_event/2, 1464
log_file/2, 1463
log_file_/2, 1459, 1461
log_point_/3, 502
logger, 1458
logging, 1460
logging/1, 1464
logging/3, 516
logging_to_file_/2, 1459, 1461
loggingp, 1462
logistic/3, 1863
logistic_regression_classifier, 1466
lognormal/3, 1859
logseries/2, 1860
logtalk, 386
logtalk_packs/0, 1732
logtalk_packs/1, 1732
long_flags/1, 374
longest_common_increasing_subsequence/3, 2154
longest_common_subsequence/3, 2127, 2152
longest_common_subsequence_length/3, 2126
longest_common_substring/3, 2127
longest_decreasing_subsequence/2, 2153
longest_increasing_subsequence/2, 2153
longest_repeating_subsequence/2, 2154
lookup/2, 658
lookup/3, 658
lookup/4, 658
lookup_in/3, 1625
loop, 1468
loopp, 1469
lovins_stemmer(Representation), 2107
low_surrogate/1, 223
lower_triangular_part/2, 1428
lower_triangular_part/3, 1428
lower_upper/2, 2269
lpop/3, 1970
lpush/4, 1969
lrange/5, 1970
lrem/5, 1971
ltrim/5, 1972
lucas/2, 2311

M

magic, 2464
magic/2, 2465
magic_expansion(Mode), 2466
magicise/4, 2465
make/1, 1323
make_directory/1, 1671
make_directory_path/1, 1672
make_set/3, 2403
man/1, 1015
manhattan_distance/3, 2327
manhattan_norm/2, 2326
map/2, 531, 662, 749, 1516, 1648, 1833, 2433
map/3, 531, 663, 1516, 1834, 2337, 2420
map/4, 1516, 2420
map/5, 1517
map/6, 1517
map/7, 1518
map/8, 1518
map_both/3, 753, 2435
map_catching/2, 753, 2434
map_element/2, 1197
map_id/2, 190
map_impl/2, 677
map_impl/3, 677
map_invalid/2, 2435
map_member/3, 2495
map_or_else/3, 754, 1653, 2434
map_reduce/5, 1518
map_store/4, 2495
map_unexpected/2, 752
mapped_single_byte_character_set, 240
mapping/2, 242
mask32/1, 953
mask64/1, 959
massey_ranker, 1475
master_channel_frame_count/2, 144
materialize/0, 455
matrix_column/3, 1422
matrix_column_means/2, 1417
matrix_diagonal/2, 1423
matrix_diagonal/3, 1423
matrix_entry/4, 1881
matrix_matrix_product/3, 1416
matrix_rank/2, 1432
matrix_rank/3, 1432
matrix_row/3, 1421

matrix_row_means/2, 1417
matrix_trace/2, 1426
matrix_value/4, 1422
matrix_vector_product/3, 1415
max/2, 2094, 2292, 2323
max/3, 661
max_clauses/1, 1527
max_inv_preds/1, 1527
max_path/5, 893
max_probability/3, 282
max_size/1, 64
max_tree/3, 929
maxheap, 1003
maximal_cliques/2, 901
maximum_cliques/2, 902
maximum_linear_system_residual/4, 2006
maybe, 1637
maybe/0, 1854
maybe/1, 1855
maybe/2, 1855
maybe_call/1, 1855
maybe_call/2, 1856
mcp_prompt_protocol, 1476
mcp_resource_protocol, 1479
mcp_server, 1481
mcp_tool_protocol, 1488
md5, 967
mean_center/2, 829
mean_deviation/2, 2098
mean_squared_error/3, 2106
measurement/4, 1890
median/2, 2097, 2325
median_deviation/2, 2098
meets/2, 1119
member/2, 1127, 1850, 2042, 2292
memberchk/2, 2043, 2293, 2356
memcached, 1492
merge/3, 999
merge_options/2, 1661, 2485
mermaid_graph_language, 635
message_body/2, 21
message_cache_/1, 1301, 1302, 1369, 1370, 1372, 1374
message_delivery_tag/2, 22
message_diagram_description/1, 576
message_exchange/2, 22
message_hook/4, 391
message_pack(StringRepresentation), 1505
message_prefix_file/6, 391
message_prefix_stream/4, 390
message_properties/2, 21
message_property/3, 21
message_routing_key/2, 22
message_tokens//2, 390
met_by/2, 1119
meta, 1507
meta/1, 375
meta_compiler, 1519
meta_type/3, 2349
metagol, 1521
metagol_example_protocol, 1529
metap, 1509
metaphone/2, 2130
metaphone_match/2, 2130
metarule/6, 1525
metarule_next_id/1, 1527
method/2, 1235
method_/7, 1296
method_not_found/2, 1231
mget/3, 1500, 1961
mi_metric, 347
mi_metric(Stroud), 348
mibenum/1, 227
midpoint/3, 826
milliseconds/2, 205
mime_types, 1531
min/2, 2094, 2293, 2323
min/3, 660
min_clauses/1, 1526
min_distances/3, 893
min_max/3, 2094, 2323
min_max_normalization/2, 2105
min_max_normalize/4, 2089
min_path/5, 892
min_path_bellman_ford/5, 924
min_paths/3, 917
min_predecessors/3, 893
min_tree/3, 929
mine/2, 1781
mine/3, 1781
minheap, 1004
minimal_output, 1356
minimum_enclosing_circle/3, 829
minkowski_distance/4, 2328
missing_one_hot_encode/3, 1996
missing_predicate_directive_/3, 2487
mixed_feature_distance/5, 271
modes/2, 2097, 2325
modified_z_score_anomaly_detector, 1539
module_predicate_called_/3, 2486
module_property/2, 638
modules_diagram_support, 637
monitor, 727
monitor/1, 724
monitor/4, 725
monitor_activated/0, 730
monitored/1, 725
monitoring, 403

monitorp, 728
monitors/1, 724
month/2, 203
month_weekday_date/5, 467
mset/3, 1962
msort/2, 2293
msort/3, 2294
mul32/3, 955
mul64/3, 960
multifile_directive_/3, 2488
multinomial/2, 2315
multiset/3, 1544
multiset/4, 1545
multiset_index/4, 1547
multiset_index/5, 1548
multisets, 1541
multisets/3, 1544
multisets/4, 1544
multisets_protocol, 1542
murmurhash3_x64_128, 969
murmurhash3_x86_128, 970
murmurhash3_x86_32, 972
mutation/2, 1586
mutation/3, 1607, 1609
mutation/4, 1610
mutation_testing, 1563
mutation_testing_messages, 1575
mutations, 1606
mutations_store, 1608
mutator_common, 1577
mutator_protocol, 1585

N

nack/3, 2118
naive_bayes_classifier, 1611
name/1, 48, 226, 373, 1686, 1754
name_of_day/3, 462
name_of_month/3, 463
nan_bits/1, 1081
nan_kind/2, 1090
nan_payload/2, 1090
nan_representation/1, 1097
nanoid, 1612
nanoid(Representation,Size,Alphabet), 1614
nanoid_protocol, 1615
natural, 2309
natural//1, 867
navltree, 1619
nbintree, 1621
nearest_centroid_classifier, 1617
nearest_coordinate/5, 829
nearest_point_on_polyline/4, 838
nearest_point_on_segment/4, 838
neighbor_state/2, 2057
neighbor_state/3, 2058
neighbors/2, 798
neighbors/3, 889
neighbors_bits/3, 804
neighbors_int/3, 802
nested_dictionary_protocol, 1622
network/1, 1106
new/1, 885, 1187, 1380, 1623, 2341
new/2, 885, 1107, 1186, 1380, 2403
new/3, 886, 1117
new_line//0, 859
new_lines//0, 859
new_matrix/4, 1414
new_vector/3, 1410
new_vector_like/2, 1410
next/2, 2508
next/3, 2509
next/4, 660
next_arrangement/3, 75
next_cartesian_tuple/3, 118
next_combination/3, 371
next_derangement/3, 543
next_impl/5, 678
next_multiset/3, 1551
next_occurrence/1, 1581
next_partition/3, 1774
next_partition/4, 1774
next_permutation/2, 1799
nextto/3, 2294, 2357
nmea, 1628
nmea_protocol, 1630
nmf_projection, 1634
noc_metric, 350
node/6, 569
node/7, 619
node2_from_node4/2, 678
node_/6, 576
node_counter_/1, 551
node_id_/2, 551
node_path_/2, 576
nodebug/0, 510
nodes/2, 1107
nolog/3, 517
nologall/0, 517
non_blank//1, 860
non_blanks//1, 861
non_standard_predicate_call_/2, 2487
nonempty_subsequence/2, 2146
nonempty_subsequences/2, 2146
nor_metric, 351
normal/3, 1859
normal_element/2, 1068
normalize/2, 1127, 2408
normalize_continuous/4, 295, 1995

normalize_coordinate/2, 821
normalize_date_time/2, 469
normalize_polygon_orientation/3, 833
normalize_range/2, 2328
normalize_range/4, 2329
normalize_scalar/2, 2330
normalize_strengths/6, 1901
normalize_unit/2, 2329
normalize_vector/2, 1437
normalize_vector/3, 1437
nospy/1, 513
nospy/3, 514
nospy/4, 515
nospyall/0, 515
not64/2, 962
not_excluded_file/3, 331
not_excluded_file/4, 571
note/1, 1323
note/2, 1756
note/3, 1688
notification/2, 1229
notification/3, 1228
notrace/0, 511
now/3, 474
nrbtree, 1627
nth0/3, 2295, 2357
nth0/4, 2295, 2357
nth1/3, 2295, 2358
nth1/4, 2296, 2358
nth_arrangement/4, 71
nth_arrangement/5, 71
nth_cartesian_tuple/3, 114
nth_cartesian_tuple/4, 115
nth_combination/4, 366
nth_combination/5, 367
nth_derangement/3, 538
nth_derangement/4, 539
nth_distinct_arrangement/4, 73
nth_distinct_cartesian_tuple/3, 116
nth_distinct_combination/4, 368
nth_distinct_derangement/3, 541
nth_distinct_multiset/4, 1549
nth_distinct_partition/3, 1768
nth_distinct_partition/4, 1769
nth_distinct_partition/5, 1769
nth_distinct_permutation/3, 1801
nth_multiset/4, 1547
nth_multiset/5, 1547
nth_partition/3, 1765
nth_partition/4, 1766
nth_partition/5, 1766
nth_permutation/3, 1799
nth_permutation/4, 1800
null/1, 1192
null_device_path/1, 1674
null_space/2, 1436
null_space/3, 1436
number, 2316
number//1, 868
number_grammars(Format), 864
number_of_edges/2, 891
number_of_tests/1, 1311
number_of_vertices/2, 891
number_string/2, 2133
numberlist, 2319
numberlistp, 2321
numbervars/1, 2345
numbervars/3, 2344

O

object_file_/2, 1299
object_predicate_called_/3, 2486
object_wrapper_hook, 1051
object_wrapper_hook(Name,Relations), 1054
object_wrapper_hook(Protocol), 1053
observer, 521
occurrences/2, 2297
occurrences/3, 2297
occurs/2, 2341
ocf/2, 131, 135
ocf_flag/2, 144
of/2, 1641
of_expected/2, 740
of_invalid/2, 2423
of_invalids/2, 2424
of_unexpected/2, 740
of_valid/2, 2423
offset/2, 2373
offset/3, 2199, 2373
offset/4, 2373
omit_path_prefix/3, 575
one_hot_encode/4, 1996
one_or_more//0, 872
one_or_more//1, 872
one_or_more//2, 871
operating_system_machine/1, 1682
operating_system_name/1, 1682
operating_system_release/1, 1683
operating_system_type/1, 1682
optics_clusterer, 1635
optimize_rule_body/2, 449
optimize_strengths/8, 1887, 1898
option/2, 1660
option/3, 1660
optional, 1640
optional(Optional), 1645
options, 1655
options_protocol, 1656

or/2, 755, 1649, 2436
or64/3, 961
or_else/2, 750, 1650, 2437
or_else_call/2, 751, 1651, 2438
or_else_fail/1, 752, 1652, 2438
or_else_get/2, 750, 1651, 2437
or_else_throw/1, 751, 2438
or_else_throw/2, 755, 1652, 2439
order/1, 1091, 1097
order_bytes/3, 1082
ordered_scores/3, 1936
originator/1, 51
orphaned/0, 1698
orphaned/2, 1698
os, 1662
os_types, 1664
osp, 1666
out/1, 1388
out/2, 1388
out_degree/3, 878
outdated/0, 1697
outdated/1, 1697
outdated/2, 1696
outdated/4, 1695
outdated/5, 1694
outer_product/3, 1418
output_edges/1, 570
output_externals/1, 567
output_file/4, 568
output_file_name/2, 617
output_file_path/4, 572
output_files/2, 568
output_library/3, 567
output_missing_externals/1, 571
output_node/6, 569
output_rdirectory/3, 567
output_rlibrary/3, 566
output_schema/2, 1491
output_sub_diagrams/1, 568
overlapped_by/2, 1120
overlaps/2, 1120

P

pack_dependency/6, 1705
pack_metadata/4, 1701
pack_object/3, 1702
pack_property/4, 1701
pack_protocol, 1684
package/1, 50
packet_order_flag/2, 146
packetize_aos_frames/7, 177
packetize_aos_frames/8, 178
packetize_aos_packets/6, 176
packetize_aos_packets/7, 176
packetize_tm_frames/7, 176
packetize_tm_frames/8, 177
packetize_tm_packets/6, 175
packetize_tm_packets/7, 175
packets/2, 163
packs, 1689
packs_common, 1725
packs_messages, 1737
packs_specs_hook, 1738
pad_md/4, 957
pairing_heap(Order), 1006
pairing_heap_max, 1007
pairing_heap_min, 1008
pairs, 2332
pairs_to_edges/2, 882
pairwise_dataset_connected_components/2, 1915
pairwise_dataset_declared_items/2, 1914
pairwise_dataset_items/2, 1914
pairwise_dataset_matchups/2, 1916
pairwise_dataset_preferences/2, 1914
pairwise_dataset_summary/2, 1915
pairwise_dataset_win_totals/2, 1915
pairwise_measurement_dataset_connected_components/2, 1917
pairwise_measurement_dataset_declared_items/2, 1916
pairwise_measurement_dataset_items/2, 1917
pairwise_measurement_dataset_measurements/2, 1917
pairwise_measurement_dataset_protocol, 1889
pairwise_measurement_dataset_summary/2, 1918
pairwise_ranking_dataset_protocol, 1891
pairwise_strength_ranker_common, 1893
params/2, 1235
parent/1, 1035
parent/2, 799
parent_stack_/1, 550
parenthesis/2, 2268
parents/1, 1035
parse/2, 87, 89, 91, 93, 95, 97, 121, 127, 181, 210, 811, 1094, 1204, 1210, 1217, 1222, 1244, 1506, 1631, 1823, 2206, 2215, 2230, 2407, 2475, 2492, 2501
parse/3, 87, 218, 1631, 1823, 2492
parse/4, 378
parse/5, 378
parse_all/2, 2501
parse_domain/2, 1788
parse_domain/3, 1788
parse_error/1, 1230
parse_fragment/2, 1223
parse_frame/3, 150
parse_frames/3, 150
parse_lines/4, 2206

parse_problem/2, 1789
parse_problem/3, 1789
partial_/1, 1365, 1367
partial_map/4, 666
partition/2, 1761
partition/3, 737, 1761, 2419
partition/4, 1512, 1762, 2036
partition/5, 2298
partition/6, 1512
partition_body_literals/5, 450
partition_index/3, 1766
partition_index/4, 1767
partition_index/5, 1767
partition_number/2, 2314
partitions, 1757
partitions/2, 1760
partitions/3, 1761
partitions/4, 1762
partitions_protocol, 1758
pass_info/1, 715
pass_info/2, 715
passed_/3, 1353
path/3, 892
path_concat/3, 1670
path_consistency/2, 1114
pattern_length/2, 2030
pattern_miner_common, 1776
pattern_miner_diagnostics/6, 1777
pattern_miner_diagnostics_data/2, 1777
pattern_miner_export_template/4, 1778
pattern_miner_options/2, 1783
pattern_miner_protocol, 1780
pca_projection, 1785
pcdata_7bit//1, 2497
pddl, 1787
peek_back/2, 530
peek_front/2, 530
pending_data/2, 161
pending_fragments/2, 153, 162, 194
pending_packets/2, 174
percentile/3, 2101
period/1, 1938
permutation/2, 1795, 1852, 2298, 2359
permutation/3, 1796
permutation_index/3, 1800
permutation_index/4, 1800
permutations, 1792
permutations/2, 1795
permutations/3, 1795
permutations_protocol, 1793
persist/3, 1965
pid/1, 1668
pin/0, 1728
pin/1, 1728
pinned/1, 1729
plackett_luce_common, 1899
plackett_luce_last_ranker, 1804
plackett_luce_ranker, 1806
plantuml_graph_language, 639
pls_projection, 1808
plus/3, 2281
point_in_polygon/2, 830
point_to_polyline_distance/3, 838
poisson/2, 1863
polygon_area/2, 831
polygon_bounding_box/2, 831
polygon_centroid/2, 831
polygon_covering/4, 805
polygon_orientation/2, 832
polygon_perimeter/2, 840
polygon_perimeter/3, 841
polygons_intersect/2, 841
polyline_covering/4, 806
polyline_length/2, 839
polyline_length/3, 839
polyline_resample/3, 840
polyline_simplify/3, 839
polyline_split_at_distance/4, 840
pool_config/5, 28
pop_back/3, 529
pop_front/3, 529
population, 2082
port/5, 1813
port_/5, 1815
porter_stemmer(Representation), 2109
portray_clause/1, 1455
ports_profiler, 1810
possible/4, 1108
postorder/2, 651
power/2, 1864
power_sequence/4, 2282
power_set/2, 2147
powerset/2, 2043
pp/1, 2493
pp_string/1, 2496
pprint/1, 1525
pprint_clause/1, 1528
pprint_clauses/1, 1528
pq_insert/3, 921
precision/1, 1091, 1096
precision/3, 797
precision_spec/5, 1085
pred_info/3, 714
predicate/1, 547
predicate/2, 489, 547, 1567
predicate/3, 490, 1568
predicate_breakpoint_/3, 503
predicate_called_but_not_defined_/2, 2486

predicate_coverage_/7, 1304
predicate_directive_suppression(Entity,Predicate,Directive), 1587
predicate_entity_/4, 1278
predicate_info_pair_score_hook/4, 340
predicate_info_score_hook/3, 339
predicate_mode_score_hook/3, 339
predicate_mode_score_hook/5, 339
predicate_mutants/3, 1571
predicate_mutants/4, 1572
predicate_stratum/3, 458
predicate_stratum_/3, 445
predicates/2, 488
predicates/3, 489
predicates_in_stratum/2, 448
predict/3, 41, 274, 2009
predict/4, 42, 1260, 1618
predict_distribution/3, 100, 783
predict_from_probabilities/3, 282
predict_probabilities/3, 284
predict_probabilities/4, 1260, 1618
predict_scores/3, 1273, 1826
preference/3, 1892
preferred_mime_name/1, 226
prefix/0, 1733
prefix/1, 1733
prefix/2, 2298, 2359
prefix/3, 2299
prefix_data/2, 163
prefix_span_pattern_miner, 1816
preorder/2, 651
prepend/3, 1497
preprocess_groups/10, 1901
preprocess_matchups/6, 1896
preprocessing_diagnostics/3, 689
previous/2, 2509
previous/3, 2509
previous/4, 659
previous_arrangement/3, 75
previous_cartesian_tuple/3, 118
previous_combination/3, 371
previous_derangement/3, 543
previous_impl/5, 678
previous_multiset/3, 1551
previous_partition/3, 1775
previous_partition/4, 1775
previous_permutation/2, 1799
primary_office_bic/1, 1176
print_anomaly_detector/1, 43
print_anomaly_detector_template/1, 37
print_classifier/1, 277
print_classifier_template/1, 265
print_clusterer/1, 302
print_dimension_reducer/1, 701
print_dimension_reducer_details/3, 695
print_dimension_reducer_proportions/1, 685
print_flags/0, 765, 770
print_flags/1, 765
print_goal_hook, 1056
print_message/3, 389
print_message_token/4, 390
print_message_tokens/3, 389
print_mutation/3, 1578
print_pattern_miner/1, 1785
print_ranker/1, 1911
print_ranker_template/1, 1905
print_readme_file_path/1, 1734
print_regressor/1, 2012
print_regressor_template/1, 1984
probabilistic_classifier_common, 281
probabilistic_classifier_protocol, 283
probabilistic_pca_projection, 1818
probe_mutation_happened_/0, 1574
probing_/0, 1574
process, 1819
process_all/1, 322
process_directory/2, 321
process_entity/2, 320
process_file/2, 320
process_library/2, 321
process_rdirectory/2, 321
process_rlibrary/2, 322
product/2, 2093, 2324
product/3, 2043
program_to_clauses/2, 1525
progress/5, 2060
project_components/4, 694
prolog_module_hook(Module), 1057
prompt_get/3, 1478
prompts/1, 1478
propagate/2, 1111
propagate/3, 1111, 2207
propagate/4, 2207
propagate/5, 2208
propagate_state/4, 2208
propagate_state/5, 2208
propagation_triple/2, 1113
propagation_triples/2, 1114
proper_prefix/2, 2299
proper_prefix/3, 2300
proper_subsequence/2, 2155
proper_suffix/2, 2307
proper_suffix/3, 2307
property/1, 56
proto_hierarchy, 1032
proto_hierarchyp, 1033
protobuf, 1822
prove/2, 2463

prove/3, 2463
provides/2, 1750
prune/3, 106
prune/5, 106
pseudo_inverse/2, 1434
pseudo_inverse/3, 1435
pseudo_random_protocol, 1841
push_back/3, 529
push_front/3, 529

Q

qda_classifier, 1825
qr_decomposition/3, 1431
quartiles/4, 2102
quasi_skipping_/0, 499
query/1, 456
query/2, 456
question_hook/6, 392
question_prompt_stream/4, 392
queue, 1827
queue_bind/4, 12
queue_declare/3, 11
queue_delete/3, 12
queue_purge/2, 13
queue_unbind/4, 13
queuep, 1828
quick_check/1, 1315
quick_check/2, 1314
quick_check/3, 1314

R

random, 1843
random(Algorithm), 1844
random/1, 1849
random/3, 1853
random_arrangement/3, 74
random_cartesian_tuple/2, 116
random_combination/3, 369
random_derangement/2, 541
random_distinct_arrangement/3, 74
random_distinct_cartesian_tuple/2, 117
random_distinct_combination/3, 370
random_distinct_derangement/2, 542
random_distinct_multiset/3, 1550
random_distinct_partition/2, 1772
random_distinct_partition/3, 1773
random_distinct_permutation/2, 1803
random_forest_classifier, 1870
random_forest_regression, 1871
random_multiset/3, 1549
random_node/1, 2416
random_partition/2, 1771
random_partition/3, 1771
random_permutation/2, 1802
random_projection, 1873
random_protocol, 1848
random_subsequence/2, 2160
random_tree/1, 2444
randomize/1, 1840, 1846
randseq/4, 1853
randset/4, 1854
range/2, 2095
range_query/5, 307
rank/3, 1908
rank_by_scores/3, 1930
rank_centrality, 1875
rank_correlation/3, 2103
ranked_pairs, 1876
ranker_common, 1902
ranker_diagnostics_data/2, 1904
ranker_export_template/4, 1904
ranker_options/2, 1909
ranker_protocol, 1906
ranker_scores_data/2, 1903
ranker_term_template/2, 1904
ranking_dataset_common, 1912
ranking_dataset_protocol, 1927
rbtree, 664
rd/1, 1391
rd/2, 1391
rd_list/2, 1393
rd_list/3, 1392
rd_noblock/1, 1392
rd_noblock/2, 1392
rdirectories/1, 1287
rdirectories/2, 1287
rdirectory/1, 315, 486, 561, 585, 1288, 2480
rdirectory/2, 314, 486, 561, 585, 1288, 2480
rdirectory/3, 561, 585
rdirectory_score/2, 319
reachable/3, 890
read_file, 1790
read_file/2, 427, 1791, 2245
read_file/3, 426, 2244
read_file_by_line/2, 430, 2248
read_file_by_line/3, 429, 2247
read_framed_message/2, 1238
read_from_atom/2, 2168
read_from_chars/2, 2169
read_from_codes/2, 2170
read_message/2, 1238
read_mime_types/2, 1537
read_only_device_path/1, 1675
read_stream/2, 428, 2246
read_stream/3, 426, 2244
read_stream_by_line/2, 431, 2249
read_stream_by_line/3, 430, 2248
read_term_from_atom/3, 2168

read_term_from_chars/3, 2168
read_term_from_chars/4, 2169
read_term_from_codes/3, 2170
read_term_from_codes/4, 2170
reader, 1939
readme/1, 1731
readme/2, 1731
readme_file_path/2, 1734
reassemble_aos_frames/5, 171
reassemble_aos_frames/6, 172
reassemble_aos_frames/7, 172
reassemble_aos_packets/5, 170
reassemble_aos_packets/6, 170
reassemble_aos_packets/7, 171
reassemble_frames/6, 157
reassemble_frames/7, 157
reassemble_frames/8, 157
reassemble_packet_zone/5, 166
reassemble_packets/6, 155
reassemble_packets/7, 156
reassemble_packets/8, 156
reassemble_tc_frame/4, 196
reassemble_tc_frame/5, 196
reassemble_tc_frame/6, 196
reassemble_tc_frame_with_provenance/5, 198
reassemble_tc_frame_with_provenance/6, 198
reassemble_tc_frames/4, 197
reassemble_tc_frames/5, 197
reassemble_tc_frames/6, 198
reassemble_tc_frames_with_provenance/5, 199
reassemble_tc_frames_with_provenance/6, 199
reassemble_tm_frames/5, 169
reassemble_tm_frames/6, 169
reassemble_tm_frames/7, 170
reassemble_tm_packets/5, 168
reassemble_tm_packets/6, 168
reassemble_tm_packets/7, 168
receive/3, 18, 2117
record_/3, 1954
recorda/2, 1951
recorda/3, 1950
recorded/2, 1953
recorded/3, 1952
recorded_database, 1948
recorded_database_core, 1949
recorded_linter_warning_/7, 1352
recorded_warning_/4, 1452
recordz/2, 1952
recordz/3, 1951
redis, 1955
reference_/1, 1954
referenced_entity_/2, 605
referenced_logtalk_directory_/1, 595
referenced_logtalk_file_/1, 612
referenced_logtalk_library_/2, 630
referenced_module_/2, 605
referenced_predicate_/1, 647
referenced_prolog_directory_/1, 595
referenced_prolog_file_/1, 612
referenced_prolog_library_/2, 631
refine/5, 1110
refine_propagate/3, 1113
refine_propagate/4, 1113
refine_propagate/5, 1112
refine_propagate/6, 1112
registries, 1740
registry_loader_hook, 1752
registry_protocol, 1753
regression_dataset_protocol, 1976
regression_tree, 2013
regressor_common, 1979
regressor_diagnostics_data/2, 1982
regressor_export_template/4, 1982
regressor_options/2, 2011
regressor_protocol, 2007
regressor_term_template/2, 1983
regularize_ridge_matrix/4, 2003
regularized_bradley_terry_ranker, 2015
relation/1, 1102
relation/3, 1118
relation/4, 1107
relation_set/1, 1126
relational_operator_replacement(Entity,Predicate,ClauseIndex,Occu
1589
relative_standard_deviation/2, 2099
relax_neighbors/7, 921
release/1, 26
release_date/1, 52
relevance/3, 1928
remember_included_directory/1, 594
remember_included_file/1, 611
remember_included_library/2, 629
remember_referenced_logtalk_directory/1, 594
remember_referenced_logtalk_file/1, 611
remember_referenced_logtalk_library/2, 629
remember_referenced_prolog_directory/1, 594
remember_referenced_prolog_file/1, 611
remember_referenced_prolog_library/2, 630
remove_candidate/3, 296
remove_directive_/2, 2489
remove_duplicates/2, 2300, 2359
remove_rule/1, 453
removeDependent/1, 525
rename/4, 1966
rename_file/2, 1679
replace/3, 2512
replace/5, 1496
replace_sub_atom/4, 2256

report_directory/3, 1569
report_entity/3, 1568
report_library/3, 1569
report_predicate/4, 1568
repository/1, 53
repository_branch/1, 53
repository_commit/1, 54
repository_commit_abbreviated/1, 54
repository_commit_author/1, 55
repository_commit_date/1, 54
repository_commit_message/1, 55
request/3, 1228
request/4, 1228
require_strong_connectivity/4, 1888, 1897
rescale/3, 2330
reset/0, 509, 569, 1450, 1538, 1586, 1727, 1812, 2451
reset/1, 1813
reset_counter/1, 1378
reset_counters/0, 1378
reset_flags/0, 763
reset_flags/1, 764
reset_genint/0, 786
reset_genint/1, 786
reset_gensym/0, 790
reset_gensym/1, 790
reset_monitor/0, 730
reset_seed/0, 1840, 1846
residual_sum_of_squares/4, 1998
residuals/2, 1042
resize/2, 28
resource_read/3, 1480
resources/1, 1480
response/3, 1229
restore/1, 1720
restore/2, 1719
restore_edb_facts/1, 446
restore_idb_facts/1, 446
restore_predicate_strata/1, 447
restore_snapshot/6, 445
restore_support_counts/1, 446
restore_support_edges/1, 447
result/2, 1235
retract_fact/1, 455
reverse/2, 2300, 2360
rewind/2, 2510
rewind/3, 2510
rfc_metric, 352
rhumb_bearing/3, 823
rhumb_destination_point/4, 824
rhumb_distance/3, 823
rhumb_midpoint/3, 824
ridge_feature_statistics/3, 1999
ridge_regression, 2016
rising_factorial/3, 2312
rlibraries/1, 1284
rlibraries/2, 1284
rlibrary/1, 316, 488, 559, 583, 1285
rlibrary/2, 316, 487, 558, 582, 1284
rlibrary_score/2, 318
rol32/3, 955
rol64/3, 960
rollback/0, 454
root_mean_squared_error/3, 2106
ror32/3, 955
route_distance/2, 842
route_distance/3, 842
route_distance/4, 843
rows_to_design_matrix/3, 1998
rpop/3, 1970
rpush/4, 1969
rule/2, 2453
rule/3, 2453
rule/4, 2453
rule_/3, 443
rule_expansion(Mode), 2467
rules/1, 457
run/0, 1309
run/1, 1310
run/2, 1310, 2054
run/3, 2054
run/4, 2055
run_quick_check_tests/5, 1321
run_test_set/0, 1321
run_test_sets/1, 1311
run_tests/0, 1321
run_tests/1, 1321
running_test_sets_/0, 1350
runtime_type_/3, 1539

S

sadd/4, 1972
same_instant/2, 470
same_length/2, 2301, 2360
same_length/3, 2301
sample, 2083
sample_arrangements/4, 74
sample_cartesian_tuples/3, 117
sample_combinations/4, 369
sample_derangements/3, 542
sample_distinct_arrangements/4, 75
sample_distinct_cartesian_tuples/3, 117
sample_distinct_combinations/4, 370
sample_distinct_derangements/3, 542
sample_distinct_multisets/4, 1551
sample_distinct_partitions/3, 1773
sample_distinct_partitions/4, 1774
sample_distinct_permutations/3, 1803
sample_multisets/4, 1550

sample_partitions/3, 1772
sample_partitions/4, 1772
sample_permutations/3, 1802
sampling_protocol, 1857
sarif, 2018
save/0, 2484
save/1, 1718, 2369, 2484
save/2, 1717, 2369
save_dut1_entries/1, 2185, 2194
save_edge/5, 570
save_leap_seconds_entries/1, 2183, 2192
sbom, 2021
scalar_product/3, 2328
scale_matrix/3, 1419
scale_vector/3, 1412
scaled_row_difference/4, 2006
scan_left/4, 1513
scan_left_1/3, 1514
scan_right/4, 1515
scan_right_1/3, 1515
scard/3, 1973
schulze_ranker, 2024
scope_directive_replacement(Entity,Predicate,DirectiveIndex,Occurrence,PrintMutation), 1590
score/3, 42, 1178
score_all/3, 42, 1178
score_ranker_common, 1929
score_ranker_data/4, 1934
score_ranker_label/1, 1932
score_ranker_model/1, 1932
score_ranker_model_common, 1931
score_ranker_term/4, 1933
scores/2, 1908
sdbm_32, 973
sdbm_64, 974
search/1, 1705
search_index_cell_size/2, 307
search_indexing, 305
secondary_header/2, 147, 186
secondary_header_flag/2, 145, 184
secondary_header_time/3, 186
section_started_/1, 550
seed_/2, 1840, 1847
seen_/1, 1585
segment_data/2, 193
segment_header/2, 141
segment_header_suffix/2, 192
segment_header_suffixes/2, 193
segment_length_identifier/2, 146
select/3, 1850, 2044, 2301, 2360
select/4, 679, 1851, 2302
select_frequent_item_supports/3, 777, 2032
select_metric_pivot/4, 308
select_pivot_row/3, 2004
selectchk/3, 2044, 2302
selectchk/4, 2302
selected_test_/1, 1351
selection_exposures/4, 1901
send/3, 1959
send/4, 2116
send_heartbeat/1, 20, 2119
sentence_type/2, 1632
sequence/2, 738, 1639, 2028, 2420
sequence/3, 2281
sequence/4, 1852, 2278, 2282
sequence/5, 2278
sequence_count/2, 185
sequence_dataset_protocol, 2027
sequence_flags/2, 184, 190
sequence_grammars, 870
sequence_number/2, 141
sequential_occurrences/2, 2296
sequential_occurrences/3, 2296
sequential_pattern_mining_common, 2029
serve/3, 1833
server_accept/4, 2079
server_accept/5, 2079
server_close/1, 2079
server_name_/1, 1486
server_open/2, 2077
server_open/3, 2077
server_open/4, 2076
server_running_/1, 1400
server_shutdown_/0, 1400
server_socket_/1, 1399
server_title_/1, 1486
server_version_/1, 1486
set, 2033
set(Type), 2035
set/1, 2450
set/3, 1495
set/4, 1853, 1960
set/5, 1495
set_binary_input/1, 1330
set_binary_input/2, 1330
set_binary_input/3, 1329
set_binary_output/1, 1340
set_binary_output/2, 1339
set_binary_output/3, 1339
set_element/2, 1197
set_field/2, 1186
set_flag_value/2, 763
set_flag_value/3, 763
set_matrix_entry/5, 1881
set_monitor/4, 726
set_seed/1, 64, 1842
set_spy_point/4, 731
set_text_input/1, 1326

set_text_input/2, 1326
set_text_input/3, 1325
set_text_output/1, 1334
set_text_output/2, 1333
set_text_output/3, 1333
set_write_max_depth/1, 518
setp, 2037
setrange/5, 1961
setup/0, 1322, 1727
sgd_classifier, 2048
sha1, 976
sha256, 977
sha256sum_command/1, 1735
sha3_224, 978
sha3_256, 980
sha3_384, 981
sha3_512, 982
shake128(OutputBytes), 984
shake256(OutputBytes), 985
shell, 2468
shell(Interpreters), 2470
shell/1, 1668
shell/2, 1668
shell_command/1, 709
shell_expansion(Mode), 2471
shift_matrix_diagonal/3, 1426
shl64/3, 962
short_flags/1, 374
shr64/3, 962
shrink/3, 62
shrink_sequence/3, 63
shrinker/1, 62
shutdown_server/1, 1387
sign//1, 868
signaling_field/2, 124
simulated_annealing(Problem), 2050
simulated_annealing(Problem,RandomAlgorithm), 2051
simulated_annealing_protocol, 2056
single_byte_character_set(MaxCode), 243
singleton/2, 1127
singletons/2, 2344
siphash_2_4, 986
siphash_2_4(Key), 988
sismember/4, 1973
size/2, 664, 999, 2042
size_metric, 354
skewness/2, 2099
skipped_/1, 1353
skipping_/0, 499
skipping_unleashed_/1, 499
sleep/1, 1684
sliding_window/3, 2159
smembers/3, 1973
snapshot_/6, 445
snowflakeid, 2061
snowflakeid(Representation,EpochMilliseconds,TimeUnitMilliseconds, 2062
snowflakeid_instagram, 2064
snowflakeid_instagram(Representation), 2066
snowflakeid_protocol, 2067
snowflakeid_sonyflake, 2069
snowflakeid_sonyflake(Representation), 2070
snowflakeid_twitter, 2071
snowflakeid_twitter(Representation), 2072
socket, 2074
softmax/2, 2331
softmax/3, 2331
software_heritage_identifier/1, 56
solve_cholesky/3, 1441
solve_linear_system/3, 1429
solve_linear_system/4, 2003
solve_linear_systems/3, 1429
solve_lower_triangular_matrix/3, 1442
solve_upper_triangular/3, 1442
solve_upper_triangular_matrix/3, 1442
sort/2, 2036, 2303
sort/3, 2303
sort/4, 2304
sort_item_supports/2, 2032
sort_patterns/2, 776, 2031
sorted_median/3, 2088
soundex/2, 2129
soundex_match/2, 2129
source_bits/2, 1079
source_file_extension/1, 638
space//0, 858
spacecraft_id/2, 130, 134
spaces//0, 858
spade_pattern_miner, 2081
spdx_license_schema_/1, 2024
splaytree, 666
split/3, 2256
split/4, 2304
split_aos_packet_zone/3, 166
split_packet_zone/4, 164
split_sorted_rows/5, 308
split_string/4, 2136
spy/1, 512
spy/3, 513
spy/4, 514
spy_point/4, 731
spy_point_/4, 728
spying/1, 512
spying/3, 514
spying/4, 515
squared_error_sum/6, 2090
squares_and_cubes/6, 2087

squares_and_hypers/6, 2087
squeeze/3, 948
srem/4, 1972
stabilize_vector_sign/2, 1438
stabilize_vector_sign/3, 1438
standard_cauchy/3, 1868
standard_deviation/2, 2098
standard_error/2, 2105
standard_exponential/1, 1868
standard_gamma/2, 1868
standard_normal/1, 1869
standard_t/2, 1867
start/0, 1811, 2469
start/2, 1483
start/3, 1484
start/4, 1484
start/5, 1484
start_redirect_to_file/2, 520
started_by/2, 1121
starts/2, 1120
state_energy/2, 2058
statistics, 2085
statisticsp, 2091
stats/1, 27
stats/2, 1504
stats/3, 1504
status_file_/1, 1595
stem/2, 2111
stemmer_protocol, 2110
stems/2, 2112
stirling_first/3, 2313
stirling_second/3, 2314
stomp, 2113
stop/0, 1811
stop_condition/3, 2059
stop_redirect_to_file/0, 520
strata/1, 458
strata_from_numbers/2, 447
stream_position/1, 1349
stream_to_bytes/2, 1945
stream_to_bytes/3, 1946
stream_to_chars/2, 1944
stream_to_chars/3, 1944
stream_to_codes/2, 1943
stream_to_codes/3, 1943
stream_to_terms/2, 1945
stream_to_terms/3, 1945
streamvars, 1379
strength_dictionary/2, 1888, 1898
strength_pairs/3, 1887, 1895
string(Representation), 2132
string_chars/2, 2134
string_codes/2, 2134
string_concat/3, 2135
string_distance(Representation), 2122
string_length/2, 2135
string_lower/2, 2136
string_upper/2, 2136
strlen/3, 1961
strongest_paths/2, 2026
strongly_connected/3, 1888, 1897
strongly_connected_components/2, 879
strongly_connected_components/4, 1889, 1898
sub_diagram_/1, 609, 616, 627, 634
sub_diagram_/2, 592, 599
sub_directory/2, 322
sub_library/2, 323
sub_string/5, 2135
subclass/1, 1024
subclasses/1, 1025
subdivision/4, 1156, 1157
subfactorial/2, 2313
subject, 522
sublist/2, 2304, 2361
submilliseconds/2, 206
subprocess_coverage_hook, 1592
subprocess_mutation_hook, 1594
subscribe/4, 2116
subsequence/2, 2144
subsequence/3, 2145, 2305
subsequence/4, 2305
subsequence_at_indices/3, 2156
subsequence_length/2, 2163
subsequences, 2140
subsequences/2, 2144
subsequences/3, 2145
subsequences_protocol, 2142
subsequences_with_min_span/3, 2160
subset/2, 1128, 2044
subslices/2, 2159
substitute/4, 2306
subsumes/2, 1115, 2342
subterm/2, 2342, 2493
subtract/3, 2045, 2306, 2361
subtract_duration/3, 465
subtract_matrices/3, 1419
subtract_vectors/3, 1411
subunit_v1_output, 1358
subunit_v1_report, 1359
subunit_v2_output, 1360
subunit_v2_report, 1362
succ/2, 2281
suffix/2, 2306, 2361
suffix/3, 2307
suffix_alias/2, 1538
suffix_data/2, 164
sum/2, 2093, 2324
sum_of_squares/2, 2104

sum_of_squares/4, 2089
summary/1, 1451
summary_/2, 1297
superclass/1, 1025
superclasses/1, 1025
supervised_dimension_reduction_dataset_protocol, 703
supplier/1, 51
support_count_/2, 444
support_edge_/3, 444
supported_archive/1, 1736
supported_editor_url_scheme_prefix/1, 574
supported_range/2, 2190
supported_url_archive/1, 1736
suppress_binary_output/0, 1325
suppress_goal_hook, 1059
suppress_text_output/0, 1324
suspend_monitor/0, 730
swap/1, 754, 2436
swap/2, 1851
swap_consecutive/2, 1851
symdiff/3, 2045
symmetric_closure/2, 877
symmetric_eigen/3, 1433
symmetric_eigen/4, 1433
symmetric_eigen/5, 1434
synchronization_flag/2, 145

T

tab//0, 858
tabs//0, 859
tai_minus_utc_for_tai_unix/2, 2188
tail/2, 2149
tail1/2, 2151
tails/2, 2148
tails1/2, 2150
take/3, 2308
take/4, 2308
take_first_k/3, 296
talker/2, 1632
tap_output, 1363
tap_report, 1365
tar_command/1, 1736
target/1, 707, 1978
target_predicate/3, 1578
target_predicate_clause_index/4, 1579
target_predicate_directive/3, 1580
target_predicate_directive_index/4, 1580
target_scope_directive/3, 1579
target_scope_directive_index/4, 1580
target_supervised_dimension_reduction_dataset_protocol, 706
target_uses_directive/3, 1581
target_uses_directive_index/4, 1581
tcb_minus_tdb_approx/3, 2188
tcg_minus_tt_approx/3, 2187
tdb_minus_tt_approx/3, 2187
temporal_pairwise_dataset_connected_components/2, 1920
temporal_pairwise_dataset_declared_items/2, 1918
temporal_pairwise_dataset_games/2, 1919
temporal_pairwise_dataset_games/3, 1920
temporal_pairwise_dataset_items/2, 1918
temporal_pairwise_dataset_periods/2, 1919
temporal_pairwise_dataset_summary/2, 1920
temporal_pairwise_ranking_dataset_protocol, 1936
temporary_directory/1, 1674
temporary_file_/1, 2166
term, 2338
term/2, 2019
term/4, 2020
term_expansion/2, 384
term_io, 2164
term_io_protocol, 2166
termp, 2339
terms_to_array/2, 1194
test/1, 1311
test/2, 1350
test/3, 1350
test_/2, 1351
test_count_/1, 1365, 1367
text_file_assertion/3, 1346
text_file_assertion/4, 1346
text_input_assertion/2, 1328
text_input_assertion/3, 1328
text_output_assertion/2, 1337
text_output_assertion/3, 1336
text_output_assertion/4, 1336
text_output_contents/1, 1338
text_output_contents/2, 1338
text_output_contents/3, 1337
thurstone_mosteller_ranker, 2177
time, 471
time_of_day/4, 203
time_scales, 2179
time_scales_data, 2180
time_scales_protocol, 2189
time_stamp/1, 1680
time_string/3, 1140
time_type/2, 2372
time_type/3, 2372
time_type/4, 2371
timeout, 2201
timeout/1, 1527
timep, 472
timestamp/2, 2398

timestamp/8, 2398
timestamp_/6, 1295, 1299
tle_orbits, 2203
tle_orbits_protocol, 2205
to_deviation/2, 1884
to_expected/1, 2440
to_expected/2, 1654
to_mu/2, 1883
to_optional/1, 756, 2440
to_phi/2, 1884
to_rating/2, 1884
today/3, 462
tolerance_equal/4, 1319, 2318
toml, 2210
toml(ObjectRepresentation,PairRepresentation,StringRepresentation), 2213
toml(StringRepresentation), 2211
toml_protocol, 2214
tool/1, 1013
tool_call/3, 1490
tool_call/4, 1490
tool_diagnostics_common, 2217
tool_diagnostics_protocol, 2219
tools/0, 1013
tools/1, 1489
toon, 2225
toon(ObjectRepresentation,PairRepresentation,StringRepresentation), 2227
toon(StringRepresentation), 2226
toon_protocol, 2229
top/3, 1000
top_next/5, 1001
topological_sort/2, 877
topological_sort/3, 906
touch/3, 1501
toychrdb, 2231
trace/0, 511
trace_event/2, 393
tracing_/0, 498
train_linear_model/6, 1998
train_ridge_model/6, 1999
transaction/2, 779
transaction_dataset_protocol, 778
transform/3, 697
transform/4, 418
transitive_closure/2, 877
transitive_reduction/2, 906
transpose/2, 876, 2336
transpose_matrix/2, 1420
traverse/3, 738, 1639, 2421
treap_set, 2047
triangular/4, 1865
triangularize/2, 2004
triggered_breakpoint_/4, 504
triggered_breakpoint_enabled_/2, 505
trim/2, 2138
trim/3, 2138
trim_left/2, 2138
trim_left/3, 2139
trim_right/2, 2139
trim_right/3, 2139
trimmed_mean/3, 2104
true/1, 1191
truncated_svd_projection, 2236
truth_literal_flip(Entity,Predicate,ClauseIndex,Occurrence,PrintMut 1596
ts_findall_in_noblock/3, 1404
ts_findall_rd_noblock/3, 1404
ts_in/4, 1402
ts_in_list/4, 1403
ts_in_noblock/2, 1402
ts_out/1, 1402
ts_rd/4, 1403
ts_rd_list/4, 1404
ts_rd_noblock/2, 1403
tsv, 2238
tsv(Header), 2240
tsv(Header,Comments), 2241
tsv_protocol, 2242
tt_minus_tai/2, 2186
tuple/3, 1964
tuple_/1, 1401
tutor, 2252
tutor_explanations, 2253
two3tree, 668
tx_commit/1, 19
tx_rollback/1, 20
tx_select/1, 19
type, 2346
type/1, 374, 2349
type/2, 183
type/3, 1528, 1966
type_entity_/4, 1278
tzdb_version/1, 2392
tzif, 2363
tzif_protocol, 2365
tzif_zone_ids, 2391

U

ulid, 2393
ulid(Representation), 2394
ulid_protocol, 2396
ulid_types, 2399
undefined/2, 242
undirected_graph_common, 898
unexpected/1, 748
unexpecteds/2, 737
uniform/1, 1865

uniform/3, 1864
uninstall/0, 1716
uninstall/1, 1715
uninstall/2, 1714
union/3, 670, 905, 1128, 2046
union/4, 2046, 2403
union_all/3, 2404
union_find, 2400
union_find_protocol, 2402
union_impl/3, 670
union_impl_list/3, 671
universal/1, 1126
unix_seconds/2, 213
unix_to_date_time/2, 464
unknown_predicate_called_/2, 2487
unpin/0, 1729
unpin/1, 1729
unsafe_set_flag_value/2, 766
unsubscribe/3, 2117
unweighted_directed_graph, 902
unweighted_directed_graph(Dictionary), 904
unweighted_graph_common(Dictionary), 907
unweighted_graph_protocol, 908
unweighted_undirected_graph, 911
unweighted_undirected_graph(Dictionary), 912
unzip/2, 2508
update/0, 709, 1714, 1747
update/1, 522, 1713, 1746
update/2, 1712, 1745
update/3, 455, 657, 1711
update/4, 656
update/5, 656
update_fecf/2, 125, 128, 136, 141, 147
update_impl/3, 679
update_in/4, 1625
update_in/5, 1625
update_item_parameters/10, 1883
update_score/4, 1935
update_strengths/4, 1886, 1895
update_target_predicate_clause_index_/2, 1582
update_target_predicate_directive_index_/2, 1584
update_target_scope_directive_index_/2, 1583
update_target_uses_directive_index_/2, 1584
upn_metric, 356
upper_triangular_part/2, 1427
upper_triangular_part/3, 1427
url(Representation), 2406
us_ascii, 244
user, 405
user_data/2, 185
uses_diagram, 640
uses_diagram(Format), 642
uses_directive_resource_deletion(Entity,Predicate,DirectiveIndex,Oc 1598
utc_date_time_to_instant/2, 2194
utc_to_local/3, 465
utc_to_local_tz/3, 478
utf_16_character_set(Endian), 245
utf_16be, 246
utf_16le, 247
utf_32_character_set(Endian), 248
utf_32be, 250
utf_32le, 251
utf_8, 252
utf_8_character_set, 253
utm_crs/2, 418
utm_zone/2, 417
uuid, 2410
uuid(Representation), 2411
uuid_max/1, 2416
uuid_nil/1, 2416
uuid_null/1, 2415
uuid_protocol, 2412
uuid_v1/2, 2414
uuid_v3/3, 2414
uuid_v4/1, 2414
uuid_v5/3, 2415
uuid_v7/1, 2415

V

valid/0, 373
valid/1, 128, 133, 163, 212, 215, 1095, 1118, 2107, 2342, 2362, 2407, 2431
valid/2, 2350
valid/3, 463, 474
valid_anomaly_detector/1, 40
valid_attribute_declarations/1, 1985
valid_attribute_names/1, 265, 288, 1985
valid_baseline_class_values/1, 36
valid_bits/1, 1082
valid_channel_reassembly_state/1, 161
valid_class_values/1, 266
valid_classifier/1, 275
valid_classifier_metadata/2, 271
valid_classifier_metadata/3, 271
valid_clusterer/1, 300
valid_clusterer_metadata/3, 290
valid_continuous_encoders/1, 289
valid_conversion/3, 2196
valid_coordinate/1, 821
valid_coordinate/2, 415
valid_crs/1, 413
valid_date/3, 1143
valid_date_time/1, 469
valid_descriptor/1, 218
valid_diagnostic_choice/3, 291

valid_diagnostic_count/3, 290, 1987
valid_dimension_reducer/1, 698
valid_dimension_reducer_metadata/1, 695
valid_discontinuity_policy/1, 154, 162, 195
valid_discrete_encoders/1, 289
valid_discrete_values/1, 266, 1986
valid_encoded_rows/2, 1988
valid_feature_labels/1, 1991
valid_feature_types/2, 266
valid_geohash/1, 795
valid_instant/1, 2195
valid_item_value_pairs/2, 1905
valid_itemset_patterns/2, 777
valid_linear_encoders/1, 267, 694
valid_linear_model_convergence/1, 1995
valid_linear_model_diagnostics/1, 1988
valid_map_dispatch/1, 191
valid_mixed_encoders/1, 289
valid_mixed_vectors/2, 290
valid_option/1, 1658
valid_options/1, 1659
valid_pattern_miner/1, 1783
valid_pattern_miner_metadata/5, 1778
valid_profile/1, 150
valid_projection_components/2, 694
valid_ranker/1, 1910
valid_ranker_metadata/2, 1905
valid_reassembled_segment/1, 192
valid_reassembly_state/1, 152, 160, 193
valid_regression_encoders/1, 1986
valid_regression_tree/2, 1991
valid_regressor/1, 2010
valid_regressor_metadata/2, 1987
valid_regressor_options/1, 1986
valid_scale/1, 2190
valid_score/1, 1933
valid_score_ranker_diagnostics/3, 1934
valid_segment/1, 190
valid_sequence_patterns/2, 2032
valid_unicode_scalar/1, 222
valid_until_date/1, 52
validate/1, 770, 812, 2476
validate/2, 812, 1244, 2476
validate/3, 768, 1244
validate_grouped_dataset/1, 1923
validate_grouped_dataset/2, 1924
validate_pairwise_dataset/1, 1922
validate_pairwise_dataset/2, 1922
validate_pairwise_measurement_dataset/1, 1923
validate_pairwise_measurement_dataset/2, 1923
validate_temporal_pairwise_dataset/1, 1924
validate_temporal_pairwise_dataset/2, 1924
validate_type/1, 768
validated, 2417
validation, 2422
validation(Validation), 2428
valids/2, 2419
value/1, 2450
value/3, 2335
value_reference/2, 1191
values/2, 662, 2335
variables/2, 2343
variance/2, 2100
variance/6, 2088
variant/2, 1317, 2343
varlist, 2351
varlistp, 2353
varnumbers/2, 2345
varnumbers/3, 2345
vector_norm/3, 1413
vector_value/3, 1421
verify_commands_availability/0, 1727
verify_fecf/1, 125, 129, 136, 142, 148
version/1, 48
version/2, 130, 134, 183, 1503
version/6, 1687
versions/3, 1699
vertex_neighbors_to_edges/4, 882
vertices/2, 887
vincenty_distance/3, 822
virtual_channel_frame_count/2, 124, 145
virtual_channel_id/2, 130, 134
void/1, 1192
void_element/1, 1067
von_mises/3, 1866

W

wait/2, 1820
waiting_/3, 1401
wald/3, 1859
wall_time/1, 1681
warning/1, 1450
warning_sequence_/1, 1452
warnings/1, 1450
weakly_connected_components/2, 880
week_of_year_iso/2, 468
weekday/2, 468
weibull/3, 1864
weight_variances/2, 100
weighted_directed_graph, 915
weighted_directed_graph(Dictionary), 916
weighted_graph_common(Dictionary), 918
weighted_graph_protocol, 922
weighted_mean/3, 2096
weighted_undirected_graph, 925
weighted_undirected_graph(Dictionary), 927
welcome/0, 2469
wfind/3, 920

when/2, 409
whiledo/2, 1470
white_space//0, 857
white_spaces//0, 858
windows_1250, 254
windows_1251, 256
windows_1252, 257
windows_1253, 258
windows_1254, 259
windows_1257, 260
winsert_neighbor/4, 919
with_connection/1, 27
with_output_to/2, 2176
within_distance/4, 828
without//2, 873
wkt_wkb, 2473
wkt_wkb_protocol, 2474
wmc_metric, 357
wneighbors/3, 924
word32_hex/2, 953
word64_hex/2, 959
word_bytes/4, 223
working_directory/1, 1673
wpairs_to_edges/2, 882
wrapper, 2478
wremove_neighbor/4, 920
wremove_vertex_from_all/3, 920
write_file/3, 432, 2250
write_framed_message/2, 1238
write_max_depth/1, 517
write_max_depth_/1, 501
write_message/2, 1237
write_stream/3, 432, 2250
write_term_to_atom/3, 2171
write_term_to_chars/3, 2172
write_term_to_chars/4, 2172
write_term_to_codes/3, 2173
write_term_to_codes/4, 2173
write_to_atom/2, 2171
write_to_chars/2, 2172
write_to_codes/2, 2173
write_to_file_hook(File), 1060
write_to_file_hook(File,Options), 1062
write_to_stream_hook(Stream), 1063
write_to_stream_hook(Stream,Options), 1064
wvertex_neighbors_to_edges/4, 883

X

xhtml11, 1070
xml, 2490
xml_to_document/3, 2494
xor64/3, 961
xref_diagram, 643
xref_diagram(Format), 644
xunit_net_v2_output, 1368
xunit_net_v2_report, 1369
xunit_output, 1371
xunit_report, 1372

Y

yaml, 2498
yaml_protocol, 2500
year/2, 202

Z

z_normalization/2, 2100
z_score_anomaly_detector, 2504
zadd/5, 1974
zap_to_port_/1, 501
zcard/3, 1975
zero_bits/2, 1084
zero_from_sign/2, 1084
zero_matrix/2, 1881
zero_or_more//0, 872
zero_or_more//1, 871
zero_or_more//2, 871
zero_vector_from_values/3, 1997
zero_vector_like/2, 690
zip/2, 2507
zip/3, 755, 1653, 2439, 2507
zip_at_index/4, 2517
zipperp, 2505
zlist, 2516
zone/3, 2370
zone_id_kind/2, 2392
zones/1, 2371
zones/2, 2371
zrange/5, 1974
zrank/4, 1975
zrem/4, 1974
zscore/4, 1976