

---

# **Fuzzy DL OWL 2**

***Release 1.0.9***

**Giuseppe Filippone**

**Jul 09, 2025**

## CONTENTS:

<b>I</b>	<b>Contents:</b>	<b>2</b>
<b>1</b>	<b>Fuzzy DL OWL 2</b>	<b>3</b>
1.1	Directory dl-examples . . . . .	3
1.2	Project Structure . . . . .	3
1.3	Test . . . . .	7
1.4	License . . . . .	7
<b>2</b>	<b>Funding</b>	<b>8</b>
<b>3</b>	<b>Installation and Configuration</b>	<b>9</b>
3.1	Configuration of the MILP solver . . . . .	9
3.2	MILP Provider Usage and Configuration . . . . .	10
3.2.1	GUROBI . . . . .	10
3.2.2	MIP . . . . .	10
3.2.3	GLPK . . . . .	10
3.2.4	CBC . . . . .	10
3.2.5	CPLEX . . . . .	10
3.2.6	HiGHS . . . . .	11
<b>4</b>	<b>Fuzzy Description Logic Grammars</b>	<b>12</b>
4.1	String and Numbers . . . . .	12
4.2	Define the semantics of the knowledge base . . . . .	12
4.3	Define truth constants . . . . .	12
4.4	Define modifiers . . . . .	12
4.5	Define concrete fuzzy concepts . . . . .	12
4.6	Define fuzzy numbers . . . . .	13
4.6.1	Fuzzy Number Operations . . . . .	13
4.6.2	Basic Fuzzy Number Representation . . . . .	13
4.6.3	Real Number Conversion . . . . .	13
4.6.4	Addition of Fuzzy Numbers . . . . .	13
4.6.5	Subtraction of Fuzzy Numbers . . . . .	13
4.6.6	Product of Fuzzy Numbers . . . . .	14
4.6.7	Division of Fuzzy Numbers . . . . .	14
4.7	Define Features, i.e., functional datatypes . . . . .	14
4.7.1	Feature Range Specifications . . . . .	14
4.7.2	Feature Definition . . . . .	14
4.7.3	Integer Range Features . . . . .	14
4.7.4	Real Number Range Features . . . . .	14
4.7.5	String Features . . . . .	14
4.7.6	Boolean Features . . . . .	14
4.8	Datatype/feature restrictions . . . . .	14
4.8.1	Datatype Restriction Operations . . . . .	15
4.8.2	Greater Than or Equal Restrictions with Variables . . . . .	15
4.8.3	Less Than or Equal Restrictions with Variables . . . . .	15

4.8.4	Equality Restrictions with Variables	15
4.8.5	Greater Than or Equal Restrictions with Fuzzy Numbers	15
4.8.6	Less Than or Equal Restrictions with Fuzzy Numbers	15
4.8.7	Equality Restrictions with Fuzzy Numbers	15
4.8.8	Function-Based Restrictions	15
4.8.9	Variable and Value Constraints	15
4.8.10	MILP Solver Constraints	16
4.9	Constraints	16
4.10	Show statements	16
4.10.1	Show Statement Operations	17
4.10.2	Concrete Filler Display	17
4.10.3	Individual-Specific Concrete Fillers	17
4.10.4	Concept Instance Degrees	17
4.10.5	Abstract Filler Analysis	17
4.10.6	Individual Abstract Fillers	17
4.10.7	Concept Membership Display	17
4.10.8	Concept Instance Values	17
4.10.9	Variable Value Display	17
4.10.10	Knowledge Base Language Display	17
4.11	Crisp declarations	17
4.12	Fuzzy relations	18
4.13	Concept expressions	18
4.13.1	Basic Concept Expressions	19
4.13.2	Top Concept	19
4.13.3	Bottom Concept	19
4.13.4	Atomic Concepts	19
4.13.5	Concrete Fuzzy Concepts	19
4.13.6	Datatype Restrictions	19
4.13.7	Logical Connectives	19
4.13.8	Standard Conjunction	19
4.13.9	Goedel Conjunction	19
4.13.10	Lukasiewicz Conjunction	19
4.13.11	Standard Disjunction	20
4.13.12	Goedel Disjunction	20
4.13.13	Lukasiewicz Disjunction	20
4.13.14	Negation	20
4.13.15	Implication Operations	20
4.13.16	Standard Implication	20
4.13.17	Goedel Implication	20
4.13.18	Lukasiewicz Implication	20
4.13.19	Kleene-Dienes Implication	20
4.13.20	Role Restrictions	20
4.13.21	Universal Role Restriction	20
4.13.22	Existential Role Restriction	20
4.13.23	Individual Value Restriction	20
4.13.24	Approximation Concepts	20
4.13.25	Upper Approximation	21
4.13.26	Loose Upper Approximation	21
4.13.27	Tight Upper Approximation	21
4.13.28	Lower Approximation	21
4.13.29	Loose Lower Approximation	21
4.13.30	Tight Lower Approximation	21
4.13.31	Special Concept Constructs	21
4.13.32	Self Reflexivity	21
4.13.33	Modified Concepts	21
4.13.34	Fuzzy Numbers in Concepts	21
4.13.35	Threshold Concepts	21
4.13.36	Weighted Concepts	21

4.13.37	Basic Weighted Concept	22
4.13.38	Weighted Sum	22
4.13.39	Weighted Maximum	22
4.13.40	Weighted Minimum	22
4.13.41	Weighted Sum Zero	22
4.13.42	Aggregation Operators	22
4.13.43	OWA Aggregation	22
4.13.44	Quantifier-Guided OWA	22
4.13.45	Choquet Integral	22
4.13.46	Sugeno Integral	22
4.13.47	Quasi-Sugeno Integral	22
4.13.48	Sigma-Count Concept	22
4.13.49	Important Constraints and Notes	22
4.14	Axioms	23
4.14.1	Assertion Axioms	24
4.14.2	Concept Assertions	24
4.14.3	Role Assertions	24
4.14.4	General Concept Inclusion Axioms	24
4.14.5	Standard General Concept Inclusion	24
4.14.6	Goedel General Concept Inclusion	24
4.14.7	Kleene-Dienes General Concept Inclusion	24
4.14.8	Lukasiewicz General Concept Inclusion	24
4.14.9	Zadeh General Concept Inclusion	24
4.14.10	Concept Definition Axioms	24
4.14.11	Complete Concept Definition	24
4.14.12	Primitive Concept Definition	24
4.14.13	Concept Equivalence	25
4.14.14	Concept Disjointness and Union	25
4.14.15	Concept Disjointness	25
4.14.16	Disjoint Union	25
4.14.17	Role Restriction Axioms	25
4.14.18	Role Range Restriction	25
4.14.19	Role Domain Restriction	25
4.14.20	Role Property Axioms	25
4.14.21	Functional Roles	25
4.14.22	Inverse Functional Roles	25
4.14.23	Reflexive Roles	25
4.14.24	Symmetric Roles	25
4.14.25	Transitive Roles	26
4.14.26	Role Relationship Axioms	26
4.14.27	Role Implication	26
4.14.28	Inverse Roles	26
4.14.29	Important Constraints	26
4.15	Queries	26
4.15.1	Knowledge Base Consistency Queries	27
4.15.2	Satisfiability Check	27
4.15.3	Instance Membership Queries	27
4.15.4	Maximum Instance Membership	27
4.15.5	Minimum Instance Membership	27
4.15.6	All Instance Memberships	27
4.15.7	Role Relationship Queries	27
4.15.8	Maximum Role Relationship	27
4.15.9	Minimum Role Relationship	27
4.15.10	Concept Subsumption Queries	27
4.15.11	Maximum Standard Subsumption	27
4.15.12	Minimum Standard Subsumption	27
4.15.13	Maximum Goedel Subsumption	28
4.15.14	Minimum Goedel Subsumption	28

4.15.15	Maximum Lukasiewicz Subsumption	28
4.15.16	Minimum Lukasiewicz Subsumption	28
4.15.17	Maximum Kleene-Dienes Subsumption	28
4.15.18	Minimum Kleene-Dienes Subsumption	28
4.15.19	Concept Satisfiability Queries	28
4.15.20	Maximum Concept Satisfiability	28
4.15.21	Minimum Concept Satisfiability	28
4.15.22	Variable Optimization Queries	28
4.15.23	Maximum Variable Value	28
4.15.24	Minimum Variable Value	28
4.15.25	Defuzzification Queries	29
4.15.26	Largest of Maxima Defuzzification	29
4.15.27	Middle of Maxima Defuzzification	29
4.15.28	Smallest of Maxima Defuzzification	29
4.15.29	Fuzzy Number Performance Analysis	29
4.15.30	Best Non-Fuzzy Performance	29
4.15.31	Important Notes on Defuzzification	29
<b>5</b>	<b>Fuzzy Concepts</b>	<b>30</b>
5.1	Base Concepts	30
5.1.1	Concept	30
5.1.2	AtomicConcept	30
5.1.3	TruthConcept	30
5.2	Logical Operators	30
5.2.1	OperatorConcept	30
5.2.2	AllSomeConcept	30
5.2.3	ImpliesConcept	30
5.3	Fuzzy Integrals	30
5.3.1	ChoquetIntegral	30
5.3.2	SugenoIntegral	31
5.3.3	QsugenoIntegral	31
5.4	OWA Concepts	31
5.4.1	OwaConcept	31
5.4.2	QowaConcept	31
5.5	Approximation and Threshold Concepts	31
5.5.1	ApproximationConcept	31
5.5.2	ThresholdConcept	31
5.5.3	ExtendedThresholdConcept	31
5.6	Value-Based Concepts	31
5.6.1	HasValueConcept	31
5.6.2	ValueConcept	31
5.6.3	SelfConcept	31
5.6.4	NegatedNominal	31
5.7	Weighted Concepts	32
5.7.1	WeightedConcept	32
5.7.2	WeightedMinConcept	32
5.7.3	WeightedMaxConcept	32
5.7.4	WeightedSumConcept	32
5.7.5	WeightedSumZeroConcept	32
5.8	Counting and Modification Concepts	32
5.8.1	SigmaConcept	32
5.8.2	ModifiedConcept	32
5.8.3	LinearlyModifiedConcept	32
5.8.4	TriangularlyModifiedConcept	32
5.9	Concrete Fuzzy Concepts	32
5.9.1	FuzzyConcreteConcept	32
5.9.2	CrispConcreteConcept	32
5.9.3	ModifiedConcreteConcept	32

5.9.4	LinearConcreteConcept . . . . .	33
5.9.5	LeftConcreteConcept . . . . .	33
5.9.6	RightConcreteConcept . . . . .	33
5.9.7	TriangularConcreteConcept . . . . .	33
5.9.8	TrapezoidalConcreteConcept . . . . .	33
5.9.9	TriangularFuzzyNumber . . . . .	33
<b>6</b>	<b>Usage</b>	<b>34</b>
6.1	Reasoning . . . . .	34
6.1.1	Knowledge base in example.fdl . . . . .	34
6.1.2	Python code . . . . .	34
6.2	Fuzzy OWL 2 . . . . .	34
6.2.1	From *.fdl to *.owl . . . . .	34
6.2.2	From *.owl to *.fdl . . . . .	34
<b>7</b>	<b>API Reference</b>	<b>35</b>
7.1	fuzzy_dl_owl2 . . . . .	35
7.1.1	Submodules . . . . .	35
7.1.1.1	fuzzy_dl_owl2.fuzzydl . . . . .	35
7.1.1.1.1	Submodules . . . . .	35
7.1.1.2	fuzzy_dl_owl2.fuzzyowl2 . . . . .	170
7.1.1.2.1	Submodules . . . . .	170
<b>8</b>	<b>Indices and tables</b>	<b>197</b>

Welcome to fuzzy\_dl\_owl2's documentation!

## **Part I**

### **Contents:**



## FUZZY DL OWL 2

A python porting of the [Fuzzy Description Language](#) and the [Fuzzy OWL 2](#) framework.

A lightweight Python porting of the Fuzzy Description Language (FuzzyDL) and the Fuzzy OWL 2 framework, designed for representing fuzzy logic within description logic and for mapping an knowledge base represented in FuzzyDL to a Fuzzy OWL 2 construct in RDF/XML format.

Features:

- Object-oriented representation of Fuzzy Description Logic elements
- Object-oriented representation of Fuzzy OWL 2 elements
- Mapping from FuzzyDL to Fuzzy OWL 2
- Mapping from Fuzzy OWL 2 to FuzzyDL
- Reasoning in FuzzyDL

### 1.1 Directory dl-examples

The directory `dl-examples` contains a few examples of Knowledge Bases written using the Fuzzy Description Logic language.

### 1.2 Project Structure

```
fuzzy_dl_owl2
├── __init__.py
├── fuzzydl
│   ├── __init__.py
│   ├── assertion
│   │   ├── __init__.py
│   │   ├── assertion.py
│   │   └── atomic_assertion.py
│   ├── classification_node.py
│   ├── concept
│   │   ├── __init__.py
│   │   ├── all_some_concept.py
│   │   ├── approximation_concept.py
│   │   ├── atomic_concept.py
│   │   ├── choquet_integral.py
│   │   ├── concept.py
│   │   ├── concrete
│   │   │   ├── __init__.py
│   │   │   ├── __pycache__
│   │   │   ├── crisp_concrete_concept.py
│   │   │   └── fuzzy_concrete_concept.py
```

(continues on next page)

(continued from previous page)

- fuzzy\_number
    - \_\_init\_\_.py
    - triangular\_fuzzy\_number.py
  - left\_concrete\_concept.py
  - linear\_concrete\_concept.py
  - modified\_concrete\_concept.py
  - right\_concrete\_concept.py
  - trapezoidal\_concrete\_concept.py
  - triangular\_concrete\_concept.py
- ext\_threshold\_concept.py
- has\_value\_concept.py
- implies\_concept.py
- interface
  - \_\_init\_\_.py
  - \_\_pycache\_\_
  - has\_concept\_interface.py
  - has\_concepts\_interface.py
  - has\_role\_concept\_interface.py
  - has\_role\_interface.py
  - has\_value\_interface.py
  - has\_weighted\_concepts\_interface.py
- modified
  - \_\_init\_\_.py
  - linearly\_modified\_concept.py
  - modified\_concept.py
  - triangularly\_modified\_concept.py
- negated\_nominal.py
- operator\_concept.py
- owa\_concept.py
- qowa\_concept.py
- quasi\_sugeno\_integral.py
- self\_concept.py
- sigma\_concept.py
- sigma\_count.py
- string\_concept.py
- sugeno\_integral.py
- threshold\_concept.py
- truth\_concept.py
- value\_concept.py
- weighted\_concept.py
- weighted\_max\_concept.py
- weighted\_min\_concept.py
- weighted\_sum\_concept.py
- weighted\_sum\_zero\_concept.py

- concept\_equivalence.py
- concrete\_feature.py
- degree
  - \_\_init\_\_.py
  - degree\_expression.py
  - degree\_numeric.py
  - degree\_variable.py
  - degree.py
- domain\_axiom.py
- exception
  - \_\_init\_\_.py
  - fuzzy\_ontology\_exception.py

(continues on next page)

(continued from previous page)

```

├── inconsistent_ontology_exception.py
├── feature_function.py
├── fuzzydl_to_owl2.py
├── general_concept_inclusion.py
├── individual
│   ├── __init__.py
│   ├── created_individual.py
│   ├── individual.py
│   └── representative_individual.py
├── knowledge_base.py
├── label.py
├── milp
│   ├── __init__.py
│   ├── expression.py
│   ├── inequation.py
│   ├── milp_helper.py
│   ├── show_variables_helper.py
│   ├── solution.py
│   ├── term.py
│   └── variable.py
├── modifier
│   ├── __init__.py
│   ├── linear_modifier.py
│   ├── modifier.py
│   └── triangular_modifier.py
├── parser
│   ├── __init__.py
│   └── dl_parser.py
├── primitive_concept_definition.py
├── query
│   ├── __init__.py
│   ├── all_instances_query.py
│   ├── bnp_query.py
│   ├── classification_query.py
│   ├── defuzzify
│   │   ├── __init__.py
│   │   ├── defuzzify_query.py
│   │   ├── lom_defuzzify_query.py
│   │   ├── mom_defuzzify_query.py
│   │   └── som_defuzzify_query.py
│   ├── instance_query.py
│   ├── kb_satisfiable_query.py
│   ├── max
│   │   ├── __init__.py
│   │   ├── max_instance_query.py
│   │   ├── max_query.py
│   │   ├── max_related_query.py
│   │   ├── max_satisfiable_query.py
│   │   └── max_subsumes_query.py
│   ├── min
│   │   ├── __init__.py
│   │   ├── min_instance_query.py
│   │   ├── min_query.py
│   │   ├── min_related_query.py
│   │   ├── min_satisfiable_query.py
│   │   └── min_subsumes_query.py

```

(continues on next page)

(continued from previous page)


```

    |   |   |   |
    |   |   |   |— query.py
    |   |   |   |— related_query.py
    |   |   |   |— satisfiable_query.py
    |   |   |   |— subsumption_query.py
    |   |   |— range_axiom.py
    |   |   |— relation.py
    |   |   |— restriction
    |   |   |   |   |   |
    |   |   |   |   |   |— __init__.py
    |   |   |   |   |   |— has_value_restriction.py
    |   |   |   |   |   |— restriction.py
    |   |   |— role_parent_with_degree.py
    |   |   |— util
    |   |   |   |   |   |
    |   |   |   |   |   |— __init__.py
    |   |   |   |   |   |— config_reader.py
    |   |   |   |   |   |— constants.py
    |   |   |   |   |   |— util.py
    |   |   |   |   |   |— utils.py
    |— fuzzyowl2
    |   |   |
    |   |   |— __init__.py
    |   |   |— fuzzyowl2_to_fuzzydl.py
    |   |   |— fuzzyowl2.py
    |   |   |— owl_types
    |   |   |   |   |   |
    |   |   |   |   |   |— __init__.py
    |   |   |   |   |   |— choquet_concept.py
    |   |   |   |   |   |— concept_definition.py
    |   |   |   |   |   |— fuzzy_datatype.py
    |   |   |   |   |   |— fuzzy_modifier.py
    |   |   |   |   |   |— fuzzy_nominal_concept.py
    |   |   |   |   |   |— fuzzy_property.py
    |   |   |   |   |   |— left_shoulder_function.py
    |   |   |   |   |   |— linear_function.py
    |   |   |   |   |   |— linear_modifier.py
    |   |   |   |   |   |— modified_concept.py
    |   |   |   |   |   |— modified_function.py
    |   |   |   |   |   |— modified_property.py
    |   |   |   |   |   |— owa_concept.py
    |   |   |   |   |   |— property_definition.py
    |   |   |   |   |   |— qowa_concept.py
    |   |   |   |   |   |— quasi_sugeno_concept.py
    |   |   |   |   |   |— right_shoulder_function.py
    |   |   |   |   |   |— sugeno_concept.py
    |   |   |   |   |   |— trapezoidal_function.py
    |   |   |   |   |   |— triangular_function.py
    |   |   |   |   |   |— triangular_modifier.py
    |   |   |   |   |   |— weighted_concept.py
    |   |   |   |   |   |— weighted_max_concept.py
    |   |   |   |   |   |— weighted_min_concept.py
    |   |   |   |   |   |— weighted_sum_concept.py
    |   |   |   |   |   |— weighted_sum_zero_concept.py
    |   |   |— parser
    |   |   |   |   |   |
    |   |   |   |   |   |— __init__.py
    |   |   |   |   |   |— owl2_parser.py
    |   |   |   |   |   |— owl2_xml_parser.py
    |   |   |— util
    |   |   |   |   |   |
    |   |   |   |   |   |— __init__.py
    |   |   |   |   |   |— constants.py

```

(continues on next page)

(continued from previous page)

 `fuzzy_xml.py`

## 1.3 Test

The directory `test` contains the `unittest` files. In particular, the file `test_suite.py` contains all the test suite. The directory `examples/TestSuite` contains all the knowledge bases used for the tests.

## 1.4 License

This project is licensed under the Creative Commons Attribution-ShareAlike 4.0 International.

## FUNDING

We acknowledge financial support from the **Sustainability Decision Framework (SDF)** Research Project – CUP **B79J23000540005** – Grant Assignment Decree No. **5486** adopted on **2023-08-04**.

## INSTALLATION AND CONFIGURATION

Check the [repository](#).

```
pip install fuzzy-dl-owl2
```

Examples of supported Fuzzy Description Logic Constructs

Python Class	Description
AtomicConcept	Define an atomic concept
ChoquetIntegral	Define a choquet integral concept
ApproximationConcept	Define a tight/upper/* lower/upper approximation concept

### 3.1 Configuration of the MILP solver

Since version 1.0.1 uses Gurobi Optimizer (see [gurobipy](#) for the Fuzzy DL reasoning, please create a GUROBI license to use this library.

For the configuration, create a `CONFIG.ini` file in the same directory used for the execution of the library. Example of your execution directory:

```
your_directory
├─ CONFIG.ini
└─ your_file.py
```

The file `CONFIG.ini` is structured as follows:

```
[DEFAULT]
debugPrint = False
epsilon = 0.001
maxIndividuals = -1
owlAnnotationLabel = fuzzyLabel
milpProvider = mip
```

Configuration Variable	Description
debugPrint	Enable/disable debugging
epsilon	Define the precision of the solution. For instance, <code>epsilon = 0.001</code> means that the solution will be calculated with an accuracy to the third decimal place
maxIndividuals	Define the maximal number of individuals to handle. The value <code>-1</code> indicates that there is no maximum
owlAnnotationLabel	Define the Annotation label used to build the Fuzzy OWL 2 RDF/XML ontology
milpProvider	Define the MILP provider used by the reasoner. The supported providers are listed below.

Supported MILP Providers:

Provider	milpProvider
Gurobi	gurobi
CPLEX	pulp_cplex
CBC	pulp
GLPK	pulp_glpk
HiGHS	pulp_highs
MIP	mip

## 3.2 MILP Provider Usage and Configuration

### 3.2.1 GUROBI

- Install [gurobipy](#):

```
pip install gurobipy==12.0.0
```

- Download the GUROBI license from their [website](#).
- Add Gurobi to the PATH

### 3.2.2 MIP

- Install Python [MIP](#):

```
pip install mip==1.16rc0
```

### 3.2.3 GLPK

- Install [GLPK](#) v5.0 and [GMP](#) v6.3.0
- Install Python [pulp](#):

```
pip install pulp==3.2.1
```

- Add GLPK to the PATH

### 3.2.4 CBC

- Install [CBC](#)
- Install Python [pulp](#):

```
pip install pulp==3.2.1
```

- Add CBC to the PATH

### 3.2.5 CPLEX

- Install [CPLEX](#) v22.11
- Install Python [pulp](#):

```
pip install pulp==3.2.1
```

- Add CPLEX to the PATH



### 3.2.6 HiGHS

- Install [HiGHS](#) v1.10.0
- Install python [pulp](#):

```
pip install pulp==3.2.1
```

- Add HiGHS to the PATH

## FUZZY DESCRIPTION LOGIC GRAMMATICS

### 4.1 String and Numbers

```
name      := ["][a-zA-Z_][a-zA-Z0-9_]*["]
numbers   := [+-]?[0-9]+(\.[0-9]+)
```

### 4.2 Define the semantics of the knowledge base

```
logic      := 'lukasiewicz' | 'zadeh' | 'classical'
define_logic := '(' 'define-fuzzy-logic' ["] logic ["] ')'
```

### 4.3 Define truth constants

```
constant := '(' 'define-truth-constant' name numbers ')'
```

- Example: (**define-truth-constant V 5.3**) defines the truth constant named *V* with value 5.3.

### 4.4 Define modifiers

Modifiers change the membership function of a fuzzy concept.

```
modifier := (
    '(' 'define-modifier' name 'linear-modifier' '(' numbers ')' ')'
    # linear hedge with c > 0
    | '(' 'define-modifier' name 'triangular-modifier' '(' numbers ',' numbers ','
    numbers ')' ')' # triangular function
)
```

### 4.5 Define concrete fuzzy concepts

```
concept_type := (
    'crisp' '(' numbers ',' numbers ',' numbers ',' numbers ')'
    # crisp interval
    | 'left-shoulder' '(' numbers ',' numbers ',' numbers ',' numbers ')'
    # left-shoulder function
    | 'right-shoulder' '(' numbers ',' numbers ',' numbers ',' numbers ')'
    # right-shoulder function
    | 'triangular' '(' numbers ',' numbers ',' numbers ',' numbers ')'
    # triangular function
    | 'trapezoidal' '(' numbers ',' numbers ',' numbers ',' numbers ',' numbers ')'
```

(continues on next page)

(continued from previous page)

```

→ '      # trapezoidal function
      | 'linear' '(' numbers ',' numbers ',' numbers ',' numbers ')'
# linear function
      | 'modified' '(' name ',' name ')'
# modified datatype
      )
fuzzy_concept    := '(' 'define-fuzzy-concept' name concept_type ')'

```

- Note: the fuzzy concept **modified** applies only to modifiers and datatype restrictions. Example: (**define-fuzzy-concept** **CONCEPT** **modified**(**MOD**, **F**)), where **CONCEPT** is the name of the created concrete fuzzy concept, **MOD** is the name of an already defined modifier, and **F** is the name of an already defined datatype restriction.

## 4.6 Define fuzzy numbers

```

fuzzy_number_range    := '(' 'define-fuzzy-number-range' numbers numbers ')'      # if
fuzzy numbers are used, then define the range [k1, k2]
fuzzy_number_expression := (
    name
    | numbers                                     # if fuzzy number is a real number 'n',
    then it is considered as (n, n, n)
    | '(' numbers ',' numbers ',' numbers ')'
    | '(' 'f+' fuzzy_number_expression+ ')'      # addition of
fuzzy numbers
    | '(' 'f-' fuzzy_number_expression fuzzy_number_expression ')'      # subtraction
of fuzzy numbers
    | '(' 'f*' fuzzy_number_expression+ ')'      # product of
fuzzy numbers
    | '(' 'f/' fuzzy_number_expression fuzzy_number_expression ')'      # division of
fuzzy numbers
)
fuzzy_number          := '(' 'define-fuzzy-number' name fuzzy_number_expression ')'

```

### 4.6.1 Fuzzy Number Operations

The fuzzy number system supports several types of expressions, each with specific mathematical definitions:

#### 4.6.2 Basic Fuzzy Number Representation

(**a**, **b**, **c**) represents a fuzzy number with the mathematical definition  $(a, b, c)$ .

#### 4.6.3 Real Number Conversion

**n** represents a real number that is automatically converted to the fuzzy number  $(n, n, n)$ .

#### 4.6.4 Addition of Fuzzy Numbers

$(f+ f_1 f_2 \dots f_n)$  performs addition of multiple fuzzy numbers, defined mathematically as  $(\sum_{i=0}^n a_i, \sum_{i=0}^n b_i, \sum_{i=0}^n c_i)$ .

#### 4.6.5 Subtraction of Fuzzy Numbers

$(f- f_1 f_2)$  performs subtraction between two fuzzy numbers, defined mathematically as  $(a_1 - c_2, b_1 - b_2, c_1 - a_2)$ .

### 4.6.6 Product of Fuzzy Numbers

$(f * f_1 f_2 \dots f_n)$  performs multiplication of multiple fuzzy numbers, defined mathematically as  $(\prod_{i=0}^n a_i, \prod_{i=0}^n b_i, \prod_{i=0}^n c_i)$ .

### 4.6.7 Division of Fuzzy Numbers

$(f / f_1 f_2)$  performs division between two fuzzy numbers, defined mathematically as  $(\frac{a_1}{c_2}, \frac{b_1}{b_2}, \frac{c_1}{a_2})$ .

## 4.7 Define Features, i.e., functional datatypes

```
feature          := '(' 'functional' name ')'          # first, define the feature
feature_range    := (
  '(' 'range' name '*integer*' numbers numbers ')'
  | '(' 'range' name '*real*' numbers numbers ')'
  | '(' 'range' name '*string*' ')'
  | '(' 'range' name '*boolean*' ')'
)
```

### 4.7.1 Feature Range Specifications

Features can be defined with different data types and ranges, each serving specific purposes in the fuzzy description logic system:

### 4.7.2 Feature Definition

**(functional F)** defines the feature  $F$  as a functional datatype that can be used throughout the knowledge base.

### 4.7.3 Integer Range Features

**(range F \*integer\*  $k_1$   $k_2$ )** specifies that the range of feature  $F$  consists of integer numbers within the interval  $[k_1, k_2]$ .

### 4.7.4 Real Number Range Features

**(range F \*real\*  $k_1$   $k_2$ )** specifies that the range of feature  $F$  consists of rational numbers within the interval  $[k_1, k_2]$ .

### 4.7.5 String Features

**(range F \*string\*)** specifies that the range of feature  $F$  consists of string values, allowing textual data to be handled within the fuzzy logic framework.

### 4.7.6 Boolean Features

**(range F \*boolean\*)** specifies that the range of feature  $F$  consists of boolean values (true/false), enabling logical propositions within the system.

## 4.8 Datatype/feature restrictions

```
# (>= ...) = at least datatype restriction
# (<= ...) = at most datatype restriction
# (= ...) = exact datatype restriction

restriction_function := (
  numbers
```

(continues on next page)

(continued from previous page)

```

| name
| numbers [*]? restriction_function
| restriction_function [-] restriction_function
| (restriction_function [+]) restriction_function
)
restriction          := '(' ('>=' | '<=' | '=') name (name | restriction_function |
fuzzy_number) ')'

```

### 4.8.1 Datatype Restriction Operations

The system supports various types of datatype restrictions that allow precise control over feature values and relationships:

### 4.8.2 Greater Than or Equal Restrictions with Variables

( $\geq F$  **variable**) defines a restriction where feature  $F$  must be greater than or equal to a variable, mathematically expressed as  $\sup_{b \in \Delta_D} [F^{\mathcal{I}}(x, b) \otimes (b \geq \text{variable})]$ .

### 4.8.3 Less Than or Equal Restrictions with Variables

( $\leq F$  **variable**) defines a restriction where feature  $F$  must be less than or equal to a variable, mathematically expressed as  $\sup_{b \in \Delta_D} [F^{\mathcal{I}}(x, b) \otimes (b \leq \text{variable})]$ .

### 4.8.4 Equality Restrictions with Variables

( $= F$  **variable**) defines an exact equality restriction for feature  $F$ , mathematically expressed as  $\sup_{b \in \Delta_D} [F^{\mathcal{I}}(x, b) \otimes (b = \text{variable})]$ .

### 4.8.5 Greater Than or Equal Restrictions with Fuzzy Numbers

( $\geq F$  **fuzzy\_number**) extends the comparison to fuzzy numbers, mathematically defined as  $\sup_{b', b \in \Delta_D} [F^{\mathcal{I}}(x, b) \otimes (b \geq b') \otimes \text{fuzzy\_number}(b')^{\mathcal{I}}]$ .

### 4.8.6 Less Than or Equal Restrictions with Fuzzy Numbers

( $\leq F$  **fuzzy\_number**) provides fuzzy number comparison for upper bounds, mathematically defined as  $\sup_{b', b \in \Delta_D} [F^{\mathcal{I}}(x, b) \otimes (b \leq b') \otimes \text{fuzzy\_number}(b')^{\mathcal{I}}]$ .

### 4.8.7 Equality Restrictions with Fuzzy Numbers

( $= F$  **fuzzy\_number**) establishes equality constraints with fuzzy numbers, mathematically expressed as  $\sup_{b', b \in \Delta_D} [F^{\mathcal{I}}(x, b) \otimes (b = b') \otimes \text{fuzzy\_number}(b')^{\mathcal{I}}]$ .

### 4.8.8 Function-Based Restrictions

( $\geq F$  **function**( $F_1, \dots, F_n$ )), ( $\leq F$  **function**( $F_1, \dots, F_n$ )), and ( $= F$  **function**( $F_1, \dots, F_n$ )) allow restrictions based on complex functions of multiple features, with mathematical definitions following the pattern  $\sup_{b \in \Delta_D} [F^{\mathcal{I}}(x, b) \otimes (b \text{ op } \text{function}(F_1, \dots, F_n)^{\mathcal{I}})]$  where **op** represents the comparison operator.

### 4.8.9 Variable and Value Constraints

In datatype restrictions, the variable **variable** must be declared as a (**free variable**) before its use in a datatype restriction, utilizing the **constraints** defined below. The value for  $b$  must be within the range  $[k_1, k_2]$  subset or equivalent to  $[-k_{\infty}, k_{\infty}]$  of the feature  $F$ , and the values for **variable**, **function**( $F_1, \dots, F_n$ ) and the range of **fuzzy\_number** must be in  $[-k_{\infty}, k_{\infty}]$ , where  $k_{\infty}$  is the maximal representable integer.

In datatype restrictions, the variable **variable** may be replaced with a value, such as an integer, a real number, a string, or a boolean constant (true, false), depending on the range of the feature  $F$ .

### 4.8.10 MILP Solver Constraints

The value of  $k_\infty$  varies depending on the MILP solver used, reflecting computational limitations and precision requirements:

MILP Solver	$k_\infty$
Gurobi	$1000 \cdot ((1 \ll 31) - 1)$
PULP CBC	$(1 \ll 31) - 1$
MIP	$(1 \ll 31) - 1$
PULP GLPK	$(1 \ll 28) - 1$
PULP HiGHS	$(1 \ll 28) - 1$
PULP CPLEX	$(1 \ll 28) - 1$

- Note: The value of  $k_\infty$  is different for some MILP solvers for computational issues. In particular, higher values lead to the accumulation of errors, which can distort the results. The values currently given give the same results for the test files provided.

## 4.9 Constraints

```

operator      := '>=' | '<=' | '='
term          := numbers | name | numbers [*] term | name [*] term
expression    := (term [+]) + term
inequation_constraint := expression operator numbers
constraints    := '(' 'constraints' (
  inequality_constraint
  | 'binary' name           # binary variable in {0, 1}
  | 'free' name             # continuous variable in (-inf, +inf)
) ')'
```

### 4.10 Show statements

```

statements = (
  '(' 'show-concrete-fillers' name+ ')'           # show value of the fillers
  | '(' 'show-concrete-fillers-for' name{2, } ')' # show value of the fillers
  for an individual
  | '(' 'show-concrete-instance-for' name{3, } ')' # show degrees of being the
  filler of individual instance of a concept
  | '(' 'show-abstract-fillers' name+ ')'          # show fillers and membership
  to any concept
  | '(' 'show-abstract-fillers-for' name{2, } ')'  # show fillers for an
  individuals and membership to any concept
  | '(' 'show-concepts' name+ ')'                 # show membership of
  individuals to any concept
  | '(' 'show-instances' name+ ')'                # show value of the instances
  of the listed concepts
  | '(' 'show-variables' name+ ')'                # show value of the listed
  variables
  | '(' 'show-language' ')'                       # show language of the KB,
  from ALC to SHIF(D)
)
```

### 4.10.1 Show Statement Operations

The system provides various show statements for debugging and analyzing the knowledge base contents:

### 4.10.2 Concrete Filler Display

(**show-concrete-fillers**  $F_1 \dots F_n$ ) displays the values of the fillers of the features  $F_i$ , providing concrete instantiations of feature relationships.

### 4.10.3 Individual-Specific Concrete Fillers

(**show-concrete-fillers-for ind**  $F_1 \dots F_n$ ) shows the values of the fillers of features  $F_i$  specifically for the individual **ind**, allowing targeted analysis of individual properties.

### 4.10.4 Concept Instance Degrees

(**show-concrete-instance-for ind**  $F C_1 \dots C_n$ ) displays the degrees of being the  $F$  filler of the individual **ind** instance of concepts  $C_i$ , revealing membership strengths in fuzzy concepts.

### 4.10.5 Abstract Filler Analysis

(**show-abstract-fillers**  $R_1 \dots R_n$ ) shows fillers of roles  $R_i$  and their membership to any concept, providing comprehensive relationship analysis.

### 4.10.6 Individual Abstract Fillers

(**show-abstract-fillers-for ind**  $R_1 \dots R_n$ ) displays fillers of roles  $R_i$  for the individual **ind** and their membership to any concept, focusing on specific individual relationships.

### 4.10.7 Concept Membership Display

(**show-concepts**  $a_1 \dots a_n$ ) shows membership of the individuals  $a_i$  to any concept, revealing the concept classification of individuals.

### 4.10.8 Concept Instance Values

(**show-instances**  $C_1 \dots C_n$ ) displays the values of the instances of the concepts  $C_i$ , showing which individuals belong to specific concepts.

### 4.10.9 Variable Value Display

(**show-variables**  $x_1 \dots x_n$ ) shows the values of the variables  $x_i$ , useful for debugging constraint satisfaction problems.

### 4.10.10 Knowledge Base Language Display

(**show-language**) shows the language of the knowledge base, indicating the description logic expressivity from  $\mathcal{ALC}$  to  $\mathcal{SHIF}(\mathbf{D})$ .

## 4.11 Crisp declarations

```
crisp_concepts := '(' 'crisp-concept' name+ ')' # the listed concepts are crisp
crisp_roles   := '(' 'crisp-role' name+ ')' # the listed roles are crisp
```

## 4.12 Fuzzy relations

```
fuzzy_similarity    := '(' 'define-fuzzy-similarity'  name ')' # fuzzy similarity
relation
fuzzy_equivalence  := '(' 'define-fuzzy-equivalence'  name ')' # fuzzy equivalence
relation
```

## 4.13 Concept expressions

```
concept := (
  '*top*'                                # top concept
  | '*bottom*'                            # bottom concept
  | name                                  # atomic concept or
  concrete fuzzy concept
  | restriction                           # datatype restriction
  | '(' 'and' concept concept ')'         # concept conjunction
  | '(' 'g-and' concept concept ')'       # Goedel conjunction
  | '(' 'l-and' concept concept ')'       # Lukasiewicz
  conjunction
  | '(' 'or' concept concept ')'          # concept disjunction
  | '(' 'g-or' concept concept ')'        # Goedel disjunction
  | '(' 'l-or' concept concept ')'        # Lukasiewicz
  disjunction
  | '(' 'not' concept ')'                 # concept negation
  | '(' 'implies' concept concept ')'     # concept implication
  | '(' 'g-implies' concept concept ')'   # Goedel implication
  | '(' 'l-implies' concept concept ')'   # Lukasiewicz
  implication
  | '(' 'kd-implies' concept concept ')'  # Kleene-Dienes
  implication
  | '(' 'all' name concept ')'            # universal role
  restriction
  | '(' 'some' name concept ')'           # existential role
  restriction
  | '(' 'some' name name ')'              # individual value
  restriction
  | '(' 'ua' name concept ')'             # upper approximation
  | '(' 'lua' name concept ')'            # loose upper
  approximation
  | '(' 'tua' name concept ')'            # tight upper
  approximation
  | '(' 'la' name concept ')'             # lower approximation
  | '(' 'lla' name concept ')'            # loose lower
  approximation
  | '(' 'tla' name concept ')'            # tight lower
  approximation
  | '(' 'self' concept ')'                # local reflexivity
  concept
  | '(' name concept ')'                  # modifier applied to
  concept
  | '(' fuzzy_number ')'                  # fuzzy number
  | '(' '[' ('>=' | '<=') name ']' concept ')' # threshold concept
  | '(' numbers concept ')'               # weighted concept
  | '(' 'w-sum' '(' numbers concept ')' '+' ')' # weighted sum concept
  | '(' 'w-max' '(' numbers concept ')' '+' ')' # weighted max concept
  | '(' 'w-min' '(' numbers concept ')' '+' ')' # weighted min concept
```

(continues on next page)



(continued from previous page)

```

| '(' 'w-sum-zero' '('(' numbers concept ')')+ ')'          # weighted sum zero
concept
| '(' 'owa' numbers+ concept+ ')'                          # OWA aggregation
operator
| '(' 'q-owa' name concept+ ')'                             # quantifier-guided
OWA
| '(' 'choquet' numbers+ concept+ ')'                      # Choquet integral
| '(' 'sugeno' numbers+ concept+ ')'                      # Sugeno integral
| '(' 'q-sugeno' numbers+ concept+ ')'                    # Quasi-Sugeno
integral
| '(' 'sigma-count' name concept '{' name+ '}' name ')'    # Sigma-count concept
)

```

### 4.13.1 Basic Concept Expressions

The fuzzy description logic provides fundamental concept building blocks:

### 4.13.2 Top Concept

**\*top\*** represents the universal concept that always evaluates to true, mathematically defined as  $\top = 1$ .

### 4.13.3 Bottom Concept

**\*bottom\*** represents the empty concept that always evaluates to false, mathematically defined as  $\perp = 0$ .

### 4.13.4 Atomic Concepts

**A** represents an atomic concept  $A$ , evaluated as  $A^{\mathcal{I}}(x)$  in the interpretation.

### 4.13.5 Concrete Fuzzy Concepts

**CFC** represents a concrete fuzzy concept (such as crisp, left-shoulder, and so on), evaluated as  $\text{CFC}^{\mathcal{I}}(x)$ .

### 4.13.6 Datatype Restrictions

**DR** represents a datatype restriction, evaluated as  $\text{DR}^{\mathcal{I}}(x)$ .

### 4.13.7 Logical Connectives

The system supports various logical operations with different semantics:

### 4.13.8 Standard Conjunction

(**and**  $C_1$   $C_2$ ) performs concept conjunction of  $C_1$  and  $C_2$ , defined as  $C_1^{\mathcal{I}}(x) \otimes C_2^{\mathcal{I}}(x)$ .

### 4.13.9 Goedel Conjunction

(**g-and**  $C_1$   $C_2$ ) performs Goedel conjunction of  $C_1$  and  $C_2$ , defined as  $C_1^{\mathcal{I}}(x) \otimes_G C_2^{\mathcal{I}}(x)$ .

### 4.13.10 Lukasiewicz Conjunction

(**l-and**  $C_1$   $C_2$ ) performs Lukasiewicz conjunction of  $C_1$  and  $C_2$ , defined as  $C_1^{\mathcal{I}}(x) \otimes_L C_2^{\mathcal{I}}(x)$ .

#### 4.13.11 Standard Disjunction

(**or**  $C_1$   $C_2$ ) performs concept disjunction of  $C_1$  and  $C_2$ , defined as  $C_1^{\mathcal{I}}(x) \oplus C_2^{\mathcal{I}}(x)$ .

#### 4.13.12 Goedel Disjunction

(**g-or**  $C_1$   $C_2$ ) performs Goedel disjunction of  $C_1$  and  $C_2$ , defined as  $C_1^{\mathcal{I}}(x) \oplus_G C_2^{\mathcal{I}}(x)$ .

#### 4.13.13 Lukasiewicz Disjunction

(**l-or**  $C_1$   $C_2$ ) performs Lukasiewicz disjunction of  $C_1$  and  $C_2$ , defined as  $C_1^{\mathcal{I}}(x) \oplus_L C_2^{\mathcal{I}}(x)$ .

#### 4.13.14 Negation

(**not**  $C$ ) performs concept  $C$  negation, defined as  $\ominus_L C^{\mathcal{I}}(x)$ .

#### 4.13.15 Implication Operations

Various forms of implication are supported for different logical semantics:

#### 4.13.16 Standard Implication

(**implies**  $C_1$   $C_2$ ) establishes concept implication between  $C_1$  and  $C_2$ , defined as  $C_1^{\mathcal{I}}(x) \Rightarrow C_2^{\mathcal{I}}(x)$ .

#### 4.13.17 Goedel Implication

(**g-implies**  $C_1$   $C_2$ ) establishes Goedel implication between  $C_1$  and  $C_2$ , defined as  $C_1^{\mathcal{I}}(x) \Rightarrow_G C_2^{\mathcal{I}}(x)$ .

#### 4.13.18 Lukasiewicz Implication

(**l-implies**  $C_1$   $C_2$ ) establishes Lukasiewicz implication between  $C_1$  and  $C_2$ , defined as  $C_1^{\mathcal{I}}(x) \Rightarrow_L C_2^{\mathcal{I}}(x)$ .

#### 4.13.19 Kleene-Dienes Implication

(**kd-implies**  $C_1$   $C_2$ ) establishes Kleene-Dienes implication between  $C_1$  and  $C_2$ , defined as  $C_1^{\mathcal{I}}(x) \Rightarrow_{\text{KD}} C_2^{\mathcal{I}}(x)$ .

#### 4.13.20 Role Restrictions

Role-based concept formation allows complex relationship modeling:

#### 4.13.21 Universal Role Restriction

(**all**  $R$   $C$ ) creates a universal role  $R$  restriction for concept  $C$ , defined as  $\inf_{y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x, y) \Rightarrow C^{\mathcal{I}}(y)$ .

#### 4.13.22 Existential Role Restriction

(**some**  $R$   $C$ ) creates an existential role  $R$  restriction for concept  $C$ , defined as  $\sup_{y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x, y) \otimes C^{\mathcal{I}}(y)$ .

#### 4.13.23 Individual Value Restriction

(**some**  $R$   $a$ ) creates an individual value restriction for role  $R$  and individual  $a$ , defined as  $R^{\mathcal{I}}(x, a)$ .

#### 4.13.24 Approximation Concepts

Rough set theory concepts for handling uncertainty:

#### 4.13.25 Upper Approximation

**(ua  $s$   $C$ )** defines upper approximation for a fuzzy relation  $s$  and concept  $C$ , calculated as  $\sup_{y \in \Delta^{\mathcal{I}}} s^{\mathcal{I}}(x, y) \otimes C^{\mathcal{I}}(y)$ .

#### 4.13.26 Loose Upper Approximation

**(lua  $s$   $C$ )** defines loose upper approximation for a fuzzy relation  $s$  and concept  $C$ , calculated as  $\sup_{z \in X} s^{\mathcal{I}}(x, z) \otimes \sup_{y \in \Delta^{\mathcal{I}}} s^{\mathcal{I}}(y, z) \otimes C^{\mathcal{I}}(x)$ .

#### 4.13.27 Tight Upper Approximation

**(tua  $s$   $C$ )** defines tight upper approximation for a fuzzy relation  $s$  and concept  $C$ , calculated as  $\inf_{z \in X} s^{\mathcal{I}}(x, z) \Rightarrow \sup_{y \in \Delta^{\mathcal{I}}} s^{\mathcal{I}}(y, z) \otimes C^{\mathcal{I}}(x)$ .

#### 4.13.28 Lower Approximation

**(la  $s$   $C$ )** defines lower approximation for a fuzzy relation  $s$  and concept  $C$ , calculated as  $\inf_{y \in \Delta^{\mathcal{I}}} s^{\mathcal{I}}(x, y) \Rightarrow C^{\mathcal{I}}(y)$ .

#### 4.13.29 Loose Lower Approximation

**(lla  $s$   $C$ )** defines loose lower approximation for a fuzzy relation  $s$  and concept  $C$ , calculated as  $\sup_{z \in X} s^{\mathcal{I}}(x, z) \otimes \inf_{y \in \Delta^{\mathcal{I}}} s^{\mathcal{I}}(y, z) \otimes C^{\mathcal{I}}(x)$ .

#### 4.13.30 Tight Lower Approximation

**(tla  $s$   $C$ )** defines tight lower approximation for a fuzzy relation  $s$  and concept  $C$ , calculated as  $\inf_{z \in X} s^{\mathcal{I}}(x, z) \Rightarrow \inf_{y \in \Delta^{\mathcal{I}}} s^{\mathcal{I}}(y, z) \otimes C^{\mathcal{I}}(x)$ .

#### 4.13.31 Special Concept Constructs

Advanced concept formation techniques:

#### 4.13.32 Self Reflexivity

**(self  $C$ )** defines a local reflexivity concept, evaluated as  $C^{\mathcal{I}}(x)(x, x)$ .

#### 4.13.33 Modified Concepts

**(MOD  $C$ )** applies modifier MOD to concept  $C$ , evaluated as  $f_m(C^{\mathcal{I}}(x))$ , where  $f_m$  is the modifier function associated to MOD.

#### 4.13.34 Fuzzy Numbers in Concepts

**(FN)** represents fuzzy number FN as a concept, evaluated as  $FM^{\mathcal{I}}(x)$ .

#### 4.13.35 Threshold Concepts

**([>= var ]  $C$ )** creates a threshold concept that returns  $C^{\mathcal{I}}(x)$  if  $C^{\mathcal{I}}(x) \geq \text{var}$ , otherwise returns 0.

**([<= var ]  $C$ )** creates a threshold concept that returns  $C^{\mathcal{I}}(x)$  if  $C^{\mathcal{I}}(x) \leq \text{var}$ , otherwise returns 0.

#### 4.13.36 Weighted Concepts

Concepts with numerical weights for aggregation:

#### 4.13.37 Basic Weighted Concept

(**n C**) creates a weighted concept  $C$  with weight  $n$ , evaluated as  $nC^{\mathcal{I}}(x)$ .

#### 4.13.38 Weighted Sum

(**w-sum**  $(n_1 C_1) \dots (n_k C_k)$ ) performs weighted sum of concepts, calculated as  $\sum_{i=1}^k n_i C_i^{\mathcal{I}}(x)$ .

#### 4.13.39 Weighted Maximum

(**w-max**  $(v_1 C_1) \dots (v_k C_k)$ ) performs weighted max of concepts, calculated as  $\max_{i=1}^k \min v_i, x_i$ .

#### 4.13.40 Weighted Minimum

(**w-min**  $(v_1 C_1) \dots (v_k C_k)$ ) performs weighted min of concepts, calculated as  $\min_{i=1}^k \max 1 - v_i, x_i$ .

#### 4.13.41 Weighted Sum Zero

(**w-sum-zero**  $(n_1 C_1) \dots (n_k C_k)$ ) performs weighted sum with zero-constraint, returning 0 if any  $C_i^{\mathcal{I}}(x) = 0$  for some  $i \in 1, \dots, k$ , otherwise  $\sum_{i=1}^k n_i C_i^{\mathcal{I}}(x)$ .

#### 4.13.42 Aggregation Operators

Advanced aggregation techniques for concept combination:

#### 4.13.43 OWA Aggregation

(**owa**  $(w_1, \dots, w_n) (C_1, \dots, C_n)$ ) performs OWA aggregation, calculated as  $\sum_{i=1}^n w_i y_i$ .

#### 4.13.44 Quantifier-Guided OWA

(**q-owa**  $Q (C_1, \dots, C_n)$ ) performs quantifier-guided OWA with quantifier  $Q$  (where  $Q$  is a right-shoulder or linear function), calculated as  $\sum_{i=1}^n w_i y_i$ , where  $w_i = Q(\frac{i}{n}) - Q(\frac{i-1}{n})$ .

#### 4.13.45 Choquet Integral

(**choquet**  $(w_1, \dots, w_n) (C_1, \dots, C_n)$ ) computes the Choquet integral, calculated as  $y_1 w_1 + \sum_{i=2}^n (y_i - y_{i-1}) w_i$ .

#### 4.13.46 Sugeno Integral

(**sugeno**  $(v_1, \dots, v_n) (C_1, \dots, C_n)$ ) computes the Sugeno integral, calculated as  $\max_{i=1}^n \min y_i, m u_i$ .

#### 4.13.47 Quasi-Sugeno Integral

(**q-sugeno**  $(v_1, \dots, v_n) (C_1, \dots, C_n)$ ) computes the Quasi-Sugeno integral, calculated as  $\max_{i=1}^n y_i \otimes_L m u_i$ .

#### 4.13.48 Sigma-Count Concept

(**sigma-count**  $R C a_1 \dots a_k F_C$ ) creates a Sigma-Count concept with role  $R$  and associated to the concept  $C$ , the individuals  $a_i$  and the fuzzy concrete concept  $F_C$ .

#### 4.13.49 Important Constraints and Notes

Several constraints apply to the proper use of these concept expressions:

- $n_1, \dots, n_k \in [0, 1]$ , with  $\sum_{i=1}^k n_i \leq 1$
- $w_1, \dots, w_n \in [0, 1]$ , with  $\sum_{i=1}^n w_i = 1$
- $v_1, \dots, v_n \in [0, 1]$ , with  $\max_{i=1}^n v_i = 1$

- $y_i$  is the  $i$ -largest of the  $C_i^{\mathcal{I}}(x)$
- $ow_i$  is the weight  $v_i$  of the  $i$ -largest of the  $C_i^{\mathcal{I}}(x)$
- $mu_i$  is defined as follows:  $mu_1 = ow_1$ , and  $mu_i = ow_i \oplus mu_{i-1}$  for  $i \in 2, \dots, n$
- Fuzzy numbers can only appear in existential, universal and datatype restrictions
- In threshold concepts **var** may be replaced with  $w \in [0, 1]$
- Fuzzy relations  $s$  should be previously defined as fuzzy similarity relation or a fuzzy equivalence relation as **(define-fuzzy-similarity s)** or **(define-fuzzy-equivalence s)**, respectively
- Fuzzy concrete concept  $F_C$  in **sigma-count** concept has to be previously defined as **left-shoulder**, **right-shoulder** or **triangular** concept with **define-fuzzy-concept**

## 4.14 Axioms

```

degree := (
  numbers           # a rational number
  | expression      # a linear expression
  | name            # variable or an already defined truth constant
)
axioms := (
  '(' 'instance' name concept degree? ')'          # concept assertion
  | '(' 'related' name name name degree? ')'       # role assertion
  | '(' 'implies' concept concept numbers? ')'     # General Concept Inclusion
  (GCI) with degree 'numbers'
  | '(' 'g-implies' concept concept numbers? ')'   # Goedel GCI with degree
  ↪ 'numbers'
  | '(' 'kd-implies' concept concept numbers? ')'  # Kleene-Dienes GCI with
  degree 'numbers'
  | '(' 'l-implies' concept concept numbers? ')'   # Lukasiewicz GCI with degree
  ↪ 'numbers'
  | '(' 'z-implies' concept concept numbers? ')'   # Zadeh's set GCI with degree
  ↪ 'numbers'
  | '(' 'define-concept' name concept ')'          # concept definition
  | '(' 'define-primitive-concept' name concept ')' # concept subsumption
  | '(' 'equivalent-concepts' concept concept ')'  # equivalent concept
  definition
  | '(' 'disjoint' concept+ ')'                    # concept disjointness
  | '(' 'disjoint-union' concept+ ')'              # disjoint union of concepts
  | '(' 'range' name concept ')'                   # range restriction of a
  concept
  | '(' 'domain' name concept ')'                  # domain restriction of a
  concept
  | '(' 'functional' name ')'                      # functional role
  | '(' 'inverse-functional' name ')'              # inverse functional role
  | '(' 'reflexive' name ')'                       # reflexive role
  | '(' 'symmetric' name ')'                      # symmetric role
  | '(' 'transitive' name ')'                     # transitive role
  | '(' 'implies-role' name name numbers? ')'     # Role Implication Axiom (RIA)
  | '(' 'inverse' name name ')'                   # inverse role
)

```

#### 4.14.1 Assertion Axioms

Basic statements about individuals and their relationships:

#### 4.14.2 Concept Assertions

(**instance**  $a \ C \ d$ ) asserts that individual  $a$  is an instance of concept  $C$  with degree  $d$ , formally defined as  $C^{\mathcal{I}}(a^{\mathcal{I}}) \geq d$ .

#### 4.14.3 Role Assertions

(**related**  $a \ b \ R \ d$ ) asserts that individuals  $a$  and  $b$  are related by role  $R$  with degree  $d$ , formally defined as  $R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}}) \geq d$ .

#### 4.14.4 General Concept Inclusion Axioms

Various forms of concept subsumption with different logical semantics:

#### 4.14.5 Standard General Concept Inclusion

(**implies**  $C_1 \ C_2 \ d$ ) establishes that concept  $C_1$  implies concept  $C_2$  with degree  $d$ , formally defined as  $\inf_{x \in \Delta^{\mathcal{I}}} C_1^{\mathcal{I}}(x) \Rightarrow C_2^{\mathcal{I}}(x) \geq d$ .

#### 4.14.6 Goedel General Concept Inclusion

(**g-implies**  $C_1 \ C_2 \ d$ ) establishes Goedel implication between concepts with degree  $d$ , formally defined as  $\inf_{x \in \Delta^{\mathcal{I}}} C_1^{\mathcal{I}}(x) \Rightarrow_G C_2^{\mathcal{I}}(x) \geq d$ .

#### 4.14.7 Kleene-Dienes General Concept Inclusion

(**kd-implies**  $C_1 \ C_2 \ d$ ) establishes Kleene-Dienes implication between concepts with degree  $d$ , formally defined as  $\inf_{x \in \Delta^{\mathcal{I}}} C_1^{\mathcal{I}}(x) \Rightarrow_{\text{KD}} C_2^{\mathcal{I}}(x) \geq d$ .

#### 4.14.8 Lukasiewicz General Concept Inclusion

(**l-implies**  $C_1 \ C_2 \ d$ ) establishes Lukasiewicz implication between concepts with degree  $d$ , formally defined as  $\inf_{x \in \Delta^{\mathcal{I}}} C_1^{\mathcal{I}}(x) \Rightarrow_L C_2^{\mathcal{I}}(x) \geq d$ .

#### 4.14.9 Zadeh General Concept Inclusion

(**z-implies**  $C_1 \ C_2 \ d$ ) establishes Zadeh set implication between concepts with degree  $d$ , formally defined as  $\inf_{x \in \Delta^{\mathcal{I}}} C_1^{\mathcal{I}}(x) \Rightarrow_Z C_2^{\mathcal{I}}(x) \geq d$ .

#### 4.14.10 Concept Definition Axioms

Formal concept definitions and relationships:

#### 4.14.11 Complete Concept Definition

(**define-concept**  $A \ C$ ) provides a complete definition of atomic concept  $A$  as equivalent to complex concept  $C$ , formally defined as  $\forall_{x \in \Delta^{\mathcal{I}}} A^{\mathcal{I}}(x) = C^{\mathcal{I}}(x)$ .

#### 4.14.12 Primitive Concept Definition

(**define-primitive-concept**  $A \ C$ ) establishes that atomic concept  $A$  is subsumed by complex concept  $C$ , formally defined as  $\inf_{x \in \Delta^{\mathcal{I}}} A^{\mathcal{I}}(x) \leq C^{\mathcal{I}}(x)$ .

#### 4.14.13 Concept Equivalence

(**equivalent-concepts**  $C_1 \ C_2$ ) establishes that two concepts are equivalent, formally defined as  $\forall_{x \in \Delta^{\mathcal{I}}} C_1^{\mathcal{I}}(x) = C_2^{\mathcal{I}}(x)$ .

#### 4.14.14 Concept Disjointness and Union

Axioms for controlling concept overlap:

#### 4.14.15 Concept Disjointness

(**disjoint**  $C_1 \dots C_k$ ) declares that the listed concepts are mutually disjoint, equivalent to (implies (g-and  $C_i \ C_j$ ) \*bottom\*)) for all pairs, formally  $\forall_{i,j \in 1, \dots, k, i < j} (C_i^{\mathcal{I}}(x) \otimes_G C_j^{\mathcal{I}}(x)) \Rightarrow \perp$ .

#### 4.14.16 Disjoint Union

(**disjoint-union**  $C_1 \dots C_k$ ) establishes that the first concept is the disjoint union of the others, formally  $C_1 = \bigoplus_{i=2}^k C_i$  and  $\forall_{i,j \in 1, \dots, k, i < j} (C_i^{\mathcal{I}}(x) \otimes_G C_j^{\mathcal{I}}(x)) \Rightarrow \perp$ .

#### 4.14.17 Role Restriction Axioms

Domain and range constraints for roles:

#### 4.14.18 Role Range Restriction

(**range**  $R \ C$ ) restricts the range of role  $R$  to concept  $C$ , equivalent to (implies \*top\* (all  $R \ C$ )), formally  $\top \Rightarrow \inf_{y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x, y) \Rightarrow C^{\mathcal{I}}(y)$ .

#### 4.14.19 Role Domain Restriction

(**domain**  $R \ C$ ) restricts the domain of role  $R$  to concept  $C$ , equivalent to (implies (some  $R \ \text{*top*}$ )  $C$ ), formally  $\sup_{y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x, y) \otimes \top \Rightarrow C^{\mathcal{I}}(x)$ .

#### 4.14.20 Role Property Axioms

Fundamental properties that roles can possess:

#### 4.14.21 Functional Roles

(**functional**  $R$ ) declares role  $R$  as functional, meaning each individual can be related to at most one other individual, formally  $R^{\mathcal{I}}(a, b) = R^{\mathcal{I}}(a, c) \rightarrow b = c$ .

#### 4.14.22 Inverse Functional Roles

(**inverse-functional**  $R$ ) declares role  $R$  as inverse functional, meaning each individual can be the target of at most one relationship, formally  $R^{\mathcal{I}}(b, a) = R^{\mathcal{I}}(c, a) \rightarrow b = c$ .

#### 4.14.23 Reflexive Roles

(**reflexive**  $R$ ) declares role  $R$  as reflexive, meaning every individual is related to itself, formally  $\forall_{a \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(a, a) = 1$ .

#### 4.14.24 Symmetric Roles

(**symmetric**  $R$ ) declares role  $R$  as symmetric, meaning if  $a$  is related to  $b$ , then  $b$  is related to  $a$ , formally  $\forall_{a,b \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(a, b) = R^{\mathcal{I}}(b, a)$ .

#### 4.14.25 Transitive Roles

(**transitive**  $R$ ) declares role  $R$  as transitive, meaning the relationship chains together, formally  $\forall_{a,b \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(a,b) \geq \sup_{c \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(a,c) \otimes R^{\mathcal{I}}(c,b)$ .

#### 4.14.26 Role Relationship Axioms

Axioms governing relationships between roles:

#### 4.14.27 Role Implication

(**implies-role**  $R_1 R_2 d$ ) establishes that role  $R_1$  implies role  $R_2$  with degree  $d$ , formally  $\inf_{x,y \in \Delta^{\mathcal{I}}} R_1^{\mathcal{I}}(x,y) \Rightarrow_L R_2^{\mathcal{I}}(x,y) \geq d$ .

#### 4.14.28 Inverse Roles

(**inverse**  $R_1 R_2$ ) declares that role  $R_1$  is the inverse of role  $R_2$ , formally  $R_1^{\mathcal{I}} \equiv (R_2^{\mathcal{I}})^{-1}$ .

#### 4.14.29 Important Constraints

Several important constraints must be observed when using axioms:

- Transitive roles cannot be functional
- In Zadeh logic,  $\Rightarrow$  is Zadeh's set inclusion

### 4.15 Queries

```
queries := (
  '(' 'sat?' ')' # is Knowledge base consistent?
  | '(' 'max-instance?' name concept ')'
  | '(' 'min-instance?' name concept ')'
  | '(' 'all-instances?' concept ')'
  | '(' 'max-related?' name name name ')'
  | '(' 'min-related?' name name name ')'
  | '(' 'max-subs?' concept concept ')'
  | '(' 'min-subs?' concept concept ')'
  | '(' 'max-g-subs?' concept concept ')'
  | '(' 'min-g-subs?' concept concept ')'
  | '(' 'max-l-subs?' concept concept ')'
  | '(' 'min-l-subs?' concept concept ')'
  | '(' 'max-kd-subs?' concept concept ')'
  | '(' 'min-kd-subs?' concept concept ')'
  | '(' 'max-sat?' concept name? ')'
  | '(' 'min-sat?' concept name? ')'
  | '(' 'max-var?' name ')'
  | '(' 'min-var?' name ')'
  | '(' 'defuzzify-lom?' concept name name ')' # Defuzzify using the largest of
the maxima
  | '(' 'defuzzify-mom?' concept name name ')' # Defuzzify using the middle of
the maxima
  | '(' 'defuzzify-som?' concept name name ')' # Defuzzify using the smallest of
the maxima
  | '(' 'bnp?' name ')' # Computes the Best Non-Fuzzy
Performance (BNP) of a fuzzy number
)
```



### 4.15.1 Knowledge Base Consistency Queries

Fundamental queries for knowledge base validation:

### 4.15.2 Satisfiability Check

**(sat?)** checks if the knowledge base  $\mathcal{K}$  is consistent, determining whether there exists a valid interpretation that satisfies all axioms.

### 4.15.3 Instance Membership Queries

Queries for determining individual concept membership:

### 4.15.4 Maximum Instance Membership

**(max-instance?  $a$   $C$ )** computes the maximum degree to which individual  $a$  belongs to concept  $C$ , formally  $\sup n \mid \mathcal{K} \models (\text{instance } a \text{ } C \text{ } n)$ .

### 4.15.5 Minimum Instance Membership

**(min-instance?  $a$   $C$ )** computes the minimum degree to which individual  $a$  belongs to concept  $C$ , formally  $\inf n \mid \mathcal{K} \models (\text{instance } a \text{ } C \text{ } n)$ .

### 4.15.6 All Instance Memberships

**(all-instances?  $C$ )** computes **(min-instance?  $a$   $C$ )** for every individual  $a$  in the knowledge base, providing a comprehensive view of concept membership.

### 4.15.7 Role Relationship Queries

Queries for analyzing relationships between individuals:

### 4.15.8 Maximum Role Relationship

**(max-related?  $a$   $b$   $R$ )** computes the maximum degree to which individuals  $a$  and  $b$  are related by role  $R$ , formally  $\sup n \mid \mathcal{K} \models (\text{related } a \text{ } b \text{ } R \text{ } n)$ .

### 4.15.9 Minimum Role Relationship

**(min-related?  $a$   $b$   $R$ )** computes the minimum degree to which individuals  $a$  and  $b$  are related by role  $R$ , formally  $\inf n \mid \mathcal{K} \models (\text{related } a \text{ } b \text{ } R \text{ } n)$ .

### 4.15.10 Concept Subsumption Queries

Queries for analyzing hierarchical relationships between concepts:

### 4.15.11 Maximum Standard Subsumption

**(max-subs?  $C$   $D$ )** computes the maximum degree to which concept  $D$  subsumes concept  $C$ , formally  $\sup n \mid \mathcal{K} \models (\text{implies } D \text{ } C \text{ } n)$ .

### 4.15.12 Minimum Standard Subsumption

**(min-subs?  $C$   $D$ )** computes the minimum degree to which concept  $D$  subsumes concept  $C$ , formally  $\inf n \mid \mathcal{K} \models (\text{implies } D \text{ } C \text{ } n)$ .

#### 4.15.13 Maximum Goedel Subsumption

**(max-g-subs?  $C D$ )** computes the maximum degree of Goedel subsumption between concepts, formally  $\sup n \mid \mathcal{K} \models (\text{g-implies } D C n)$ .

#### 4.15.14 Minimum Goedel Subsumption

**(min-g-subs?  $C D$ )** computes the minimum degree of Goedel subsumption between concepts, formally  $\inf n \mid \mathcal{K} \models (\text{g-implies } D C n)$ .

#### 4.15.15 Maximum Lukasiewicz Subsumption

**(max-l-subs?  $C D$ )** computes the maximum degree of Lukasiewicz subsumption between concepts, formally  $\sup n \mid \mathcal{K} \models (\text{l-implies } D C n)$ .

#### 4.15.16 Minimum Lukasiewicz Subsumption

**(min-l-subs?  $C D$ )** computes the minimum degree of Lukasiewicz subsumption between concepts, formally  $\inf n \mid \mathcal{K} \models (\text{l-implies } D C n)$ .

#### 4.15.17 Maximum Kleene-Dienes Subsumption

**(max-kd-subs?  $C D$ )** computes the maximum degree of Kleene-Dienes subsumption between concepts, formally  $\sup n \mid \mathcal{K} \models (\text{kd-implies } D C n)$ .

#### 4.15.18 Minimum Kleene-Dienes Subsumption

**(min-kd-subs?  $C D$ )** computes the minimum degree of Kleene-Dienes subsumption between concepts, formally  $\inf n \mid \mathcal{K} \models (\text{kd-implies } D C n)$ .

#### 4.15.19 Concept Satisfiability Queries

Queries for analyzing concept satisfiability across all models:

#### 4.15.20 Maximum Concept Satisfiability

**(max-sat?  $C a$ )** computes the maximum satisfiability of concept  $C$  across all interpretations, formally  $\sup_{\mathcal{I}} \sup_{a \in \Delta^{\mathcal{I}}} C^{\mathcal{I}}(a)$ .

#### 4.15.21 Minimum Concept Satisfiability

**(min-sat?  $C a$ )** computes the minimum satisfiability of concept  $C$  across all interpretations, formally  $\inf_{\mathcal{I}} \inf_{a \in \Delta^{\mathcal{I}}} C^{\mathcal{I}}(a)$ .

#### 4.15.22 Variable Optimization Queries

Queries for optimizing variable values within constraints:

#### 4.15.23 Maximum Variable Value

**(max-var?  $\text{var}$ )** computes the maximum value that variable **var** can take while maintaining knowledge base consistency, formally  $\sup \text{var} \mid \mathcal{K}$  is consistent.

#### 4.15.24 Minimum Variable Value

**(min-var?  $\text{var}$ )** computes the minimum value that variable **var** can take while maintaining knowledge base consistency, formally  $\inf \text{var} \mid \mathcal{K}$  is consistent.

#### 4.15.25 Defuzzification Queries

Queries for converting fuzzy values to crisp values using different strategies:

#### 4.15.26 Largest of Maxima Defuzzification

**(defuzzify-lom?  $C$   $a$   $F$ )** defuzzifies the value of feature  $F$  using the largest of the maxima strategy, where concept  $C$  represents Mamdani/Rules IF-THEN fuzzy rules for determining the feature value.

#### 4.15.27 Middle of Maxima Defuzzification

**(defuzzify-mom?  $C$   $a$   $F$ )** defuzzifies the value of feature  $F$  using the middle of the maxima strategy, providing a balanced approach to crisp value selection.

#### 4.15.28 Smallest of Maxima Defuzzification

**(defuzzify-som?  $C$   $a$   $F$ )** defuzzifies the value of feature  $F$  using the smallest of the maxima strategy, offering a conservative approach to defuzzification.

#### 4.15.29 Fuzzy Number Performance Analysis

#### 4.15.30 Best Non-Fuzzy Performance

**(bnp?  $f$ )** computes the Best Non-Fuzzy Performance (BNP) of a fuzzy number  $f$ , providing a measure of how well the fuzzy number can be represented as a crisp value.

#### 4.15.31 Important Notes on Defuzzification

In defuzzify queries, the concept  $C$  represents several Mamdani/Rules IF-THEN fuzzy rules expressing how to obtain the value of the concrete feature  $F$ . These rules form the basis for the fuzzy inference process that determines the appropriate crisp output value.

## FUZZY CONCEPTS

The fuzzy concept currently implemented are as follows.

### 5.1 Base Concepts

#### 5.1.1 Concept

Defines a base class of any concept. This is the fundamental class from which all other concept implementations derive.

#### 5.1.2 AtomicConcept

Defines an atomic concept. This represents the smallest, indivisible unit in the fuzzy logic system.

#### 5.1.3 TruthConcept

Defines the top  $\top$  and bottom  $\perp$  concepts. These represent absolute truth and absolute falsehood respectively in the logical system.

### 5.2 Logical Operators

#### 5.2.1 OperatorConcept

It is the class to handle a logic (Zadeh, Lukasiewicz, and product) connectives (AND, OR, NOT) between fuzzy concepts. This class implements the fundamental logical operations in fuzzy logic systems.

#### 5.2.2 AllSomeConcept

Defines a universal ( $\forall$ ) and existential ( $\exists$ ) restrictions on fuzzy concepts. These allow for expressing quantifications in the fuzzy domain.

#### 5.2.3 ImpliesConcept

Defines a zadeh and goedel implies concept. This handles implication operations in fuzzy logic.

### 5.3 Fuzzy Integrals

#### 5.3.1 ChoquetIntegral

Defines a Choquet integral of fuzzy concept. This integral is used for aggregating information when measures are non-additive.

### 5.3.2 SugenoIntegral

Defines a Sugeno integral of fuzzy concept. This is a particular type of non-linear integral with respect to fuzzy measures.

### 5.3.3 QsugenoIntegral

Defines a Quasi-Sugeno integral of fuzzy concept. This extends the Sugeno integral with additional flexibility.

## 5.4 OWA Concepts

### 5.4.1 OwaConcept

Defines a OWA concept. OWA (Ordered Weighted Averaging) operators provide a family of aggregation operators.

### 5.4.2 QowaConcept

Defines a quantified-guided OWA concept. This extends the OWA concept with quantifier-guided behavior.

## 5.5 Approximation and Threshold Concepts

### 5.5.1 ApproximationConcept

Defines uppers and lowers approximation concept. This handles rough set approximations in the fuzzy context.

### 5.5.2 ThresholdConcept

Defines a positives and negatives threshold concept. This implements threshold-based classification in fuzzy systems.

### 5.5.3 ExtendedThresholdConcept

Defines a extended positives and negatives threshold concept. This provides enhanced threshold functionality beyond the basic threshold concept.

## 5.6 Value-Based Concepts

### 5.6.1 HasValueConcept

Defines a concept associated with a value. This links concepts to specific values in the domain.

### 5.6.2 ValueConcept

Defines a datatype restriction (at most, at least, and exact) concept. This handles numerical constraints and restrictions.

### 5.6.3 SelfConcept

Defines a self reflexivity concept. This implements self-referential properties in fuzzy logic.

### 5.6.4 NegatedNominal

Defines a negated nominal concept. This handles the negation of nominal (named) concepts.

## 5.7 Weighted Concepts

### 5.7.1 WeightedConcept

Defines a weighted concept. This provides basic weighting functionality for concepts.

### 5.7.2 WeightedMinConcept

Defines a weighted min concept. This implements weighted minimum operations.

### 5.7.3 WeightedMaxConcept

Defines a weighted max concept. This implements weighted maximum operations.

### 5.7.4 WeightedSumConcept

Defines a weighted sum concept. This implements weighted summation operations.

### 5.7.5 WeightedSumZeroConcept

Defines a weighted sum-zero concept. This implements weighted sum operations with zero-centering.

## 5.8 Counting and Modification Concepts

### 5.8.1 SigmaConcept

Defines a sigma-count concept. This handles counting operations in fuzzy contexts.

### 5.8.2 ModifiedConcept

Defines the base class for modified (linear and triangular) concepts. This is the parent class for concepts that can be modified through various functions.

### 5.8.3 LinearlyModifiedConcept

Define a linearly modified concept. This applies linear modifications to base concepts.

### 5.8.4 TriangularlyModifiedConcept

Define a triangularly modified concept. This applies triangular function modifications to base concepts.

## 5.9 Concrete Fuzzy Concepts

### 5.9.1 FuzzyConcreteConcept

Defines the base class for concrete (crisp, left-shoulder, right-shoulder, and so on) concepts. This is the parent class for all concrete implementations of fuzzy concepts.

### 5.9.2 CrispConcreteConcept

Define a crisp concept. This represents classical, non-fuzzy (crisp) concepts within the fuzzy framework.

### 5.9.3 ModifiedConcreteConcept

It is a modified datatype concept. This applies modifications to concrete datatype concepts.

#### **5.9.4 LinearConcreteConcept**

It is a concept defined by a linear function. This implements concepts using linear mathematical functions.

#### **5.9.5 LeftConcreteConcept**

It is a concept defined by a left-shoulder function. This implements concepts using left-shoulder membership functions.

#### **5.9.6 RightConcreteConcept**

It is a concept defined by a right-shoulder function. This implements concepts using right-shoulder membership functions.

#### **5.9.7 TriangularConcreteConcept**

It is a concept defined by a triangular function. This implements concepts using triangular membership functions.

#### **5.9.8 TrapezoidalConcreteConcept**

It is a concept defined by a trapezoidal function. This implements concepts using trapezoidal membership functions.

#### **5.9.9 TriangularFuzzyNumber**

It is a sub-class of the `TriangularConcreteConcept` and represents a fuzzy triangular number. This provides a specialized implementation for triangular fuzzy numbers.

## 6.1 Reasoning

### 6.1.1 Knowledge base in example.fdl

```
(define-fuzzy-logic lukasiewicz)
(define-modifier very linear-modifier(0.8))
(define-fuzzy-concept eq243 crisp(0, 400, 243, 243))
(define-fuzzy-concept geq300 crisp(0, 400, 300, 400))
(define-fuzzy-concept High right-shoulder(0, 400, 180, 250))
(define-concept SportCar (and Car (some speed (very High))))
(instance ferrari (and Car (some speed geq300)) 1)
(instance audi (and Car (some speed eq243)) 1)

(min-instance? audi SportCar)
```

### 6.1.2 Python code

```
from fuzzy_dl_owl2 import DLParser

DLParser.main("./example.fdl") # "Is audi instance of SportCar ? >= 0.92"
```

## 6.2 Fuzzy OWL 2

### 6.2.1 From \*.fdl to \*.owl

```
from fuzzy_dl_owl2 import FuzzydlToOwl2

fdl = FuzzydlToOwl2("./example.fdl", "example.owl")
fdl.run() # save example.owl in the subdirectory "./results"
```

### 6.2.2 From \*.owl to \*.fdl

```
from fuzzy_dl_owl2 import FuzzyOwl2ToFuzzyDL

fdl = FuzzyOwl2ToFuzzyDL("./results/example.owl", "example.fdl")
fdl.translate_owl2ontology() # save example.fdl in the subdirectory "./results"
```



## API REFERENCE

This page contains auto-generated API reference documentation<sup>1</sup>.

### 7.1 fuzzy\_dl\_owl2

#### 7.1.1 Submodules

##### 7.1.1.1 fuzzy\_dl\_owl2.fuzzydl

##### 7.1.1.1.1 Submodules

##### fuzzy\_dl\_owl2.fuzzydl.assertion

##### Submodules

##### fuzzy\_dl\_owl2.fuzzydl.assertion.assertion

##### Classes

---

*Assertion*

---

#### Module Contents

```
class Assertion(  
    ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,  
    c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,  
    d: fuzzy_dl_owl2.fuzzydl.degree.degree.Degree,  
)  
  
    __eq__(value: Self) → bool  
    __ne__(value: Self) → bool  
    __repr__() → str  
    __str__() → str  
    clone() → Self  
    equals(ass: Self) → bool  
    get_concept() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept  
    get_individual() → fuzzy_dl_owl2.fuzzydl.individual.individual.Individual
```

---

<sup>1</sup> Created with sphinx-autoapi

```
get_lower_limit() → fuzzy_dl_owl2.fuzzydl.degree.degree.Degree
get_name_without_degree() → str
get_type() → fuzzy_dl_owl2.fuzzydl.util.constants.ConceptType
set_individual(ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual) → None
set_lower_limit(deg: fuzzy_dl_owl2.fuzzydl.degree.degree.Degree) → None
concept:  fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
degree:   fuzzy_dl_owl2.fuzzydl.degree.degree.Degree
individual: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual
```

## fuzzy\_dl\_owl2.fuzzydl.assertion.atomic\_assertion

### Classes

---

*AtomicAssertion*

---

### Module Contents

```
class AtomicAssertion(c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept, degree:
                      fuzzy_dl_owl2.fuzzydl.degree.degree.Degree)

    __str__() → str
    get_concept_name() → str
    get_degree() → fuzzy_dl_owl2.fuzzydl.degree.degree.Degree
    c:  fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
    degree:  fuzzy_dl_owl2.fuzzydl.degree.degree.Degree
```

## fuzzy\_dl\_owl2.fuzzydl.classification\_node

### Classes

---

*ClassificationNode*

---

### Module Contents

```
class ClassificationNode(name: str)

    __hash__() → int
    __repr__() → str
    __str__() → str
    add_input_edge(node: Self, n: float) → None
    add_label(c: str) → None
```

```

add_output_edge(node: Self, n: float) → None
get_full_name() → str
get_immediate_predecessors() → set[Self]
get_immediate_successors() → set[Self]
get_output_edges() → dict[Self, float]
has_name(name: str) → bool
is_nothing() → bool
is_thing() → bool
remove_input_edge(node: Self, n: float) → None
remove_output_edge(node: Self, n: float) → None
EQUIVALENT_NAMES: set[str]
INPUT_EDGES: dict[Self, float]
OUTPUT_EDGES: dict[Self, float]

```

## fuzzy\_dl\_owl2.fuzzydl.concept

### Submodules

## fuzzy\_dl\_owl2.fuzzydl.concept.all\_some\_concept

### Attributes

*All*

*Some*

### Classes

*AllSomeConcept*

Helper class that provides a standard way to create an ABC using

### Module Contents

```

class AllSomeConcept(
    role: str,
    c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    c_type: fuzzy_dl_owl2.fuzzydl.util.constants.ConceptType,
)
    Bases: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept, fuzzy_dl_owl2.fuzzydl.concept.interface.has_role_concept_interface.HasRoleConceptInterface
    Helper class that provides a standard way to create an ABC using inheritance.
    __hash__() → int
    __neg__() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept

```

```
static all(role: str, concept: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → Self
clone() → Self
compute_atomic_concepts() → set[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept]
compute_name() → str
get_atoms() → list[Self]
get_roles() → set[str]
is_complemented_atomic() → bool

static new(
    c_type: fuzzy_dl_owl2.fuzzydl.util.constants.ConceptType,
    role: str,
    concept: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
) → Self

replace(
    a: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
) → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept

static some(role: str, concept: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → Self
```

All

Some

**fuzzy\_dl\_owl2.fuzzydl.concept.approximation\_concept**

### Attributes

<i>LooseLowerApprox</i>
<i>LooseUpperApprox</i>
<i>LowerApprox</i>
<i>TightLowerApprox</i>
<i>TightUpperApprox</i>
<i>UpperApprox</i>

### Classes

<i>ApproximationConcept</i>	Helper class that provides a standard way to create an ABC using
-----------------------------	--

### Module Contents

```
class ApproximationConcept(
    c_type: fuzzy_dl_owl2.fuzzydl.util.constants.ConceptType,
    role: str,
    c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
)
```

Bases: *fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*, *fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_role\_concept\_interface.HasRoleConceptInterface*

Helper class that provides a standard way to create an ABC using inheritance.

`__and__(value: Self) → Self`

`__hash__() → int`

`__neg__() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept`

`__or__(value: Self) → Self`

`clone() → Self`

`compute_atomic_concepts() → set[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept]`

`compute_name() → str | None`

`get_roles() → set[str]`

`static loose_lower_approx(role: str, c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → Self`

`static loose_upper_approx(role: str, c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → Self`

`static lower_approx(role: str, c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → Self`

`replace(  
    a: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,  
    c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,  
) → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept`

`static tight_lower_approx(role: str, c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → Self`

`static tight_upper_approx(role: str, c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → Self`

`to_all_some_concept() → fuzzy_dl_owl2.fuzzydl.concept.all_some_concept.AllSomeConcept`

`static upper_approx(role: str, c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → Self`

`INVERSE_APPROXIMATION: dict[fuzzy_dl_owl2.fuzzydl.util.constants.ConceptType,  
fuzzy_dl_owl2.fuzzydl.util.constants.ConceptType]`

`name`

**LooseLowerApprox**

**LooseUpperApprox**

**LowerApprox**

**TightLowerApprox**

**TightUpperApprox**

**UpperApprox**

**fuzzy\_dl\_owl2.fuzzydl.concept.atomic\_concept**

**Classes**

***AtomicConcept***

Helper class that provides a standard way to create an ABC using

**Module Contents****class AtomicConcept**(*name: str*)Bases: *fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*

Helper class that provides a standard way to create an ABC using inheritance.

**\_\_and\_\_**(*value: Self*) → Self**\_\_eq\_\_**(*value: Self*) → bool**\_\_hash\_\_**() → int**\_\_invert\_\_**() → Self**\_\_ne\_\_**(*value: Self*) → bool**\_\_neg\_\_**() → Self**\_\_or\_\_**(*value: Self*) → Self**\_\_repr\_\_**() → str**\_\_rshift\_\_**(*value: Self*) → Self**clone**() → Self**compute\_atomic\_concepts**() → set[Self]**compute\_name**() → str**get\_atomic\_concepts**() → set[Self]**get\_atoms**() → list[Self]**get\_clauses**(*is\_type: Callable*) → set[Self]**get\_roles**() → set[str]**is\_atomic**() → bool**is\_complemented\_atomic**() → bool**is\_concrete**() → bool**static new\_atomic\_concept**() → Self**reduce\_idempotency**(*is\_type: Callable*) → Self**replace**(*a: Self, c: Self*) → Self | None**fuzzy\_dl\_owl2.fuzzydl.concept.choquet\_integral****Classes*****ChoquetIntegral***

Choquet integral concept.

## Module Contents

```

class ChoquetIntegral(weights: list[float], concepts:
    list[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept])

Bases: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept, fuzzy_dl_owl2.fuzzydl.concept.
interface.has_weighted_concepts_interface.HasWeightedConceptsInterface

Choquet integral concept.

__and__(value: Self) → Self

__hash__() → int

__neg__() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept

__or__(value: Self) → Self

clone() → Self

compute_atomic_concepts() → set[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept]

compute_name() → str

get_roles() → set[str]

replace(
    a: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
) → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept

```

## fuzzy\_dl\_owl2.fuzzydl.concept.concept

### Classes

<i>Concept</i>	Helper class that provides a standard way to create an ABC using
<i>Thing</i>	Helper class that provides a standard way to create an ABC using

## Module Contents

```

class Concept(c_type: fuzzy_dl_owl2.fuzzydl.util.constants.ConceptType = ConceptType.ATOMIC, name:
    str = "")

Bases: Thing

Helper class that provides a standard way to create an ABC using inheritance.

__and__(value: Self) → Self

__eq__(value: Self) → bool

__iand__(value: Self) → Self

__ior__(value: Self) → Self

__irshift__(value: Self) → Self

__ne__(value: Self) → bool

__or__(value: Self) → Self

```

```
__rshift__(value: Self) → Self
__str__() → str
is_atomic() → bool
is_complemented_atomic() → bool
DEFAULT_NAME = 'Concept@'
SPECIAL_STRING = '@'
property name: str
num_new_concepts = 1
property type: fuzzy_dl_owl2.fuzzycl.util.constants.ConceptType
```

**class Thing**

Bases: `abc.ABC`

Helper class that provides a standard way to create an ABC using inheritance.

```
abstractmethod __eq__(value: Self) → bool
__ge__(value: Self) → Self
__gt__(value: Self) → Self
__invert__() → Self
__le__(value: Self) → Self
__lt__(value: Self) → Self
__ne__(value: Self) → bool
abstractmethod __neg__() → Self
__repr__() → str
classic_cnf() → Self
classic_dnf() → Self
abstractmethod clone() → Self
abstractmethod compute_atomic_concepts() → set[Self]
abstractmethod compute_name() → str | None
static contains_negated_subconcept(v: list[Self], cj: Self) → int
static contains_subconcept(v: list[Self], cj: Self) → bool
de_morgan() → Self
distribute(c_type: fuzzy_dl_owl2.fuzzycl.util.constants.ConceptType) → Self
get_atomic_concepts() → set[Self]
get_atomic_concepts_names() → set[str]
get_atoms() → list[Self]
get_clauses(is_type: Callable) → list[Self]
```



**abstractmethod** `get_roles()` → set[str]

**goedel\_cnf()** → Self

**goedel\_dnf()** → Self

**has\_nominals()** → bool

**is\_concrete()** → bool

**is\_simplified()** → bool

**This function check if current formula is simplified, i.e., if:**

- The only negated elements are literal of kind ( $\sim A$ ), where A is an AtomicProposition
- **The OR operator is between:**
  - Two literals  $\Rightarrow A \mid B$
  - One literal and a AND  $\Rightarrow A \mid (B \ \& \ C) - (A \ \& \ B) \mid C$
  - Two (or more) OR  $\Rightarrow (A \ \& \ B) \mid (C \ \& \ D) \mid (E \ \& \ F)$
- **The AND operator is between:**
  - Two literals  $\Rightarrow A \ \& \ B$
  - One literal and a OR  $\Rightarrow A \ \& \ (B \mid C) - (A \mid B) \ \& \ C$
  - Two (or more) AND  $\Rightarrow (A \mid B) \ \& \ (C \mid D) \ \& \ (E \mid F)$
- The only operators are AND, OR and NOT

**lukasiewicz\_cnf()** → Self

**lukasiewicz\_dnf()** → Self

**normal\_form(is\_type: Callable)** → Self

**reduce\_double\_negation()** → Self

**reduce\_idempotency(is\_type: Callable)** → Self

**reduce\_quantifiers()** → Self

**reduce\_truth\_values()** → Self

**static remove\_element(v: list[Self], i: int)** → None

**abstractmethod replace(a: Self, c: Self)** → Self | None

**fuzzy\_dl\_owl2.fuzzydl.concept.concrete**

**Submodules**

**fuzzy\_dl\_owl2.fuzzydl.concept.concrete.crisp\_concrete\_concept**

**Classes**

*CrispConcreteConcept*

Concrete concept defined with a crisp interval.

## Module Contents

```
class CrispConcreteConcept(name: str, k1: float, k2: float, a: float, b: float)
    Bases: fuzzy\_dl\_owl2.fuzzydl.concept.concrete.fuzzy\_concrete\_concept.FuzzyConcreteConcept
    Concrete concept defined with a crisp interval.
    __and__(value: Self) → Self
    __hash__() → int
    __neg__() → fuzzy\_dl\_owl2.fuzzydl.concept.concrete.fuzzy\_concrete\_concept.FuzzyConcreteConcept
    __or__(value: Self) → Self
    clone() → Self
    compute_name() → str
    get_membership_degree(x: float) → float
        Get membership degree for a value
    property a: float
    property b: float
    k1: float
    k2: float
```

[fuzzy\\_dl\\_owl2.fuzzydl.concept.concrete.fuzzy\\_concrete\\_concept](#)

## Classes

<a href="#">FuzzyConcreteConcept</a>	Fuzzy concrete concept defined with an explicit membership function.
--------------------------------------	--

## Module Contents

```
class FuzzyConcreteConcept(name: str)
    Bases: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept, abc.ABC
    Fuzzy concrete concept defined with an explicit membership function.
    compute_atomic_concepts() → set[Self]
    compute_name() → str
    abstractmethod get_membership_degree(value: float) → float
        Get membership degree for a value
    get_roles() → set[str]
    is_concrete() → bool
    replace(
        concept1: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
        concept2: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    ) → fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept
    property k1: float
```

property k2: float

name: str

fuzzy\_dl\_owl2.fuzzydl.concept.concrete.fuzzy\_number

Submodules

fuzzy\_dl\_owl2.fuzzydl.concept.concrete.fuzzy\_number.triangular\_fuzzy\_number

Classes

*TriangularFuzzyNumber*

Fuzzy number defined with a triangular function.

## Module Contents

**class** **TriangularFuzzyNumber**(name: str, a: float, b: float, c: float)

**class** **TriangularFuzzyNumber**(a: float, b: float, c: float)

Bases: *fuzzy\_dl\_owl2.fuzzydl.concept.concrete.triangular\_concrete\_concept.TriangularConcreteConcept*

Fuzzy number defined with a triangular function.

**\_\_add\_\_**(other: Self) → Self

**\_\_and\_\_**(value: Self) → Self

**\_\_eq\_\_**(other: Self) → bool

**\_\_hash\_\_**() → int

**\_\_mul\_\_**(other: Self) → Self

**\_\_ne\_\_**(other: Self) → bool

**\_\_neg\_\_**() → *TriangularFuzzyNumber*

**\_\_or\_\_**(value: Self) → Self

**\_\_sub\_\_**(other: Self) → Self

**\_\_truediv\_\_**(other: Self) → Self

**static add**(t1: Self, t2: Self) → Self

Adds two triangular fuzzy numbers.

**clone**() → Self

**compute\_name**() → str

**static divided\_by**(t1: Self, t2: Self) → Self

Divides two triangular fuzzy numbers.

**get\_best\_non\_fuzzy\_performance**() → float

Gets the Best Non fuzzy Performance (BNP) of the fuzzy number.

**static has\_defined\_range**() → bool

Checks if the range of the fuzzy numbers has been defined.

**is\_concrete**() → bool

**is\_number()** → bool

**static minus**(*t1: Self, t2: Self*) → Self

Subtracts two triangular fuzzy numbers.

**static set\_range**(*min\_range: float, max\_range: float*) → None

**static times**(*t1: Self, t2: Self*) → Self

Multiplies two triangular fuzzy numbers.

**K1:** float

**K2:** float

**fuzzy\_dl\_owl2.fuzzydl.concept.concrete.left\_concrete\_concept**

### Classes

<i>LeftConcreteConcept</i>	Fuzzy concrete concept defined with a left shoulder function
----------------------------	--

### Module Contents

**class LeftConcreteConcept**(*name: str, k1: float, k2: float, a: float, b: float*)

Bases: *fuzzy\_dl\_owl2.fuzzydl.concept.concrete.fuzzy\_concrete\_concept.FuzzyConcreteConcept*

Fuzzy concrete concept defined with a left shoulder function

**\_\_and\_\_**(*value: Self*) → Self

**\_\_hash\_\_**() → int

**\_\_neg\_\_**() → *fuzzy\_dl\_owl2.fuzzydl.concept.concrete.fuzzy\_concrete\_concept.FuzzyConcreteConcept*

**\_\_or\_\_**(*value: Self*) → Self

**clone**() → Self

**compute\_name**() → str

**get\_membership\_degree**(*value: float*) → float

Get membership degree for a value

**property a:** float

**property b:** float

**k1:** float

**k2:** float

**fuzzy\_dl\_owl2.fuzzydl.concept.concrete.linear\_concrete\_concept**

### Classes

<i>LinearConcreteConcept</i>	Fuzzy concrete concept defined with a left shoulder function
------------------------------	--

## Module Contents

```

class LinearConcreteConcept(name: str, k1: float, k2: float, a: float, b: float)
    Bases: fuzzy\_dl\_owl2.fuzzydl.concept.concrete.fuzzy\_concrete\_concept.FuzzyConcreteConcept
    Fuzzy concrete concept defined with a left shoulder function
    __and__(value: Self) → Self
    __hash__() → int
    __neg__() → fuzzy\_dl\_owl2.fuzzydl.concept.concrete.fuzzy\_concrete\_concept.FuzzyConcreteConcept
    __or__(value: Self) → Self
    clone() → Self
    compute_name() → str
    get_membership_degree(value: float) → float
        Get membership degree for a value
    property a: float
    property b: float
    k1: float
    k2: float

```

[fuzzy\\_dl\\_owl2.fuzzydl.concept.concrete.modified\\_concrete\\_concept](#)

## Classes

<a href="#">ModifiedConcreteConcept</a>	Modified concrete concept.
---	----------------------------

## Module Contents

```

class ModifiedConcreteConcept(
    name: str,
    modifier: fuzzy\_dl\_owl2.fuzzydl.modifier.modifier.Modifier,
    f: fuzzy\_dl\_owl2.fuzzydl.concept.concrete.fuzzy\_concrete\_concept.FuzzyConcreteConcept,
)
    Bases: fuzzy\_dl\_owl2.fuzzydl.concept.concrete.fuzzy\_concrete\_concept.FuzzyConcreteConcept
    Modified concrete concept.
    __and__(value: Self) → Self
    __hash__() → int
    __neg__() → fuzzy\_dl\_owl2.fuzzydl.concept.concrete.fuzzy\_concrete\_concept.FuzzyConcreteConcept
    __or__(value: Self) → Self
    clone() → Self
    compute_name() → str

```

```
get_membership_degree(x: float) → float
```

Get membership degree for a value

```
k1: float = 0.0
```

```
k2: float = 1.0
```

```
property modified: fuzzy_dl_owl2.fuzzydl.concept.concrete.fuzzy_concrete_ -  
concept.FuzzyConcreteConcept
```

```
property modifier: fuzzy_dl_owl2.fuzzydl.modifier.modifier.Modifier
```

`fuzzy_dl_owl2.fuzzydl.concept.concrete.right_concrete_concept`

## Classes

<code>RightConcreteConcept</code>	Fuzzy concrete concept defined with a right shoulder function.
-----------------------------------	--

## Module Contents

```
class RightConcreteConcept(name: str, k1: float, k2: float, a: float, b: float)
```

Bases: `fuzzy_dl_owl2.fuzzydl.concept.concrete.fuzzy_concrete_concept.FuzzyConcreteConcept`

Fuzzy concrete concept defined with a right shoulder function.

```
__and__(value: Self) → Self
```

```
__hash__() → int
```

```
__neg__() → fuzzy_dl_owl2.fuzzydl.concept.concrete.fuzzy_concrete_concept.FuzzyConcreteConcept
```

```
__or__(value: Self) → Self
```

```
clone() → Self
```

```
compute_name() → str
```

```
get_membership_degree(x: float) → float
```

Get membership degree for a value

```
property a: float
```

```
property b: float
```

```
k1: float
```

```
k2: float
```

`fuzzy_dl_owl2.fuzzydl.concept.concrete.trapezoidal_concrete_concept`

## Classes

<code>TrapezoidalConcreteConcept</code>	Fuzzy concrete concept defined with a trapezoidal function.
---	---

## Module Contents

```
class TrapezoidalConcreteConcept(name: str, k1: float, k2: float, a: float, b: float, c: float, d: float)
    Bases: fuzzy_dl_owl2.fuzzydl.concept.concrete.fuzzy_concrete_concept.FuzzyConcreteConcept
    Fuzzy concrete concept defined with a trapezoidal function.
    __and__(value: Self) → Self
    __hash__() → int
    __neg__() → fuzzy_dl_owl2.fuzzydl.concept.concrete.fuzzy_concrete_concept.FuzzyConcreteConcept
    __or__(value: Self) → Self
    clone() → Self
    compute_name() → str
    get_membership_degree(x: float) → float
        Get membership degree for a value
    property a: float
    property b: float
    property c: float
    property d: float
    k1: float
    k2: float
    name: str
```

*fuzzy\_dl\_owl2.fuzzydl.concept.concrete.triangular\_concrete\_concept*

## Classes

<i>TriangularConcreteConcept</i>	Fuzzy concrete concept defined with a triangular function.
----------------------------------	--

## Module Contents

```
class TriangularConcreteConcept(name: str, k1: float, k2: float, a: float, b: float, c: float)
    Bases: fuzzy_dl_owl2.fuzzydl.concept.concrete.fuzzy_concrete_concept.FuzzyConcreteConcept
    Fuzzy concrete concept defined with a triangular function.
    __and__(value: Self) → Self
    __hash__() → int
    __neg__() → fuzzy_dl_owl2.fuzzydl.concept.concrete.fuzzy_concrete_concept.FuzzyConcreteConcept
    __or__(value: Self) → Self
    clone() → Self
```

```
compute_name() → str

get_membership_degree(x: float) → float
    Get membership degree for a value

property a: float
property b: float
property c: float

k1: float
k2: float
```

**fuzzy\_dl\_owl2.fuzzydl.concept.ext\_threshold\_concept**

### Attributes

<i>ExtendedNegThreshold</i>
<i>ExtendedPosThreshold</i>

### Classes

<i>ExtThresholdConcept</i>	Helper class that provides a standard way to create an ABC using
----------------------------	--

### Module Contents

```
class ExtThresholdConcept(
    c_type: fuzzy_dl_owl2.fuzzydl.util.constants.ConceptType,
    c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    weight_variable: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
)

Bases: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept, fuzzy_dl_owl2.fuzzydl.concept.interface.has_concept_interface.HasConceptInterface

Helper class that provides a standard way to create an ABC using inheritance.

__and__(value: Self) → Self

__hash__() → int

__neg__() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept

__or__(value: Self) → Self

clone()

compute_atomic_concepts() → set[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept]

compute_name() → str | None

static extended_neg_threshold(v: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable, c: Self) → Self

static extended_pos_threshold(v: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable, c: Self) → Self
```



**get\_roles()** → set[str]

**replace()**

*a: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept,*  
*c: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept,*  
*) → fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*

**name:** str = '(>= Uninferable] Uninferable)'

**property weight\_variable:** *fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable*

**ExtendedNegThreshold**

**ExtendedPosThreshold**

**fuzzy\_dl\_owl2.fuzzydl.concept.has\_value\_concept**

## Classes

*HasValueConcept*

Helper class that provides a standard way to create an ABC using

## Module Contents

**class HasValueConcept**(*role: str, value: Any*)

Bases: *fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept, fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_value\_interface.HasValueInterface*

Helper class that provides a standard way to create an ABC using inheritance.

**\_\_and\_\_**(*value: Self*) → Self

**\_\_hash\_\_**() → int

**\_\_neg\_\_**() → *fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*

**\_\_or\_\_**(*value: Self*) → Self

**clone**() → Self

**compute\_atomic\_concepts**() → set[*fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*]

**compute\_name**() → str | None

**get\_roles**() → set[str]

**static has\_value**(*role: str, i: Any*) → Self

**replace()**

*a: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept,*  
*c: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept,*  
*) → fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*

**name:** str

**fuzzy\_dl\_owl2.fuzzydl.concept.implies\_concept**

## Attributes

*GoedelImplies**KleeneDienesImplies**LukasiewiczImplies**ZadehImplies*

---

## Classes

*ImpliesConcept*Helper class that provides a standard way to create an ABC using

---

## Module Contents

```
class ImpliesConcept(
    c_type: fuzzy_dl_owl2.fuzzydl.util.constants.ConceptType,
    concepts: list[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept],
)

Bases: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept, fuzzy_dl_owl2.fuzzydl.concept.interface.has_concepts_interface.HasConceptsInterface

Helper class that provides a standard way to create an ABC using inheritance.

__and__(value: Self) → Self

__eq__(value: Self) → bool

__hash__() → int

__neg__() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept

__or__(value: Self) → Self

clone() → Self

compute_atomic_concepts() → set[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept]

compute_name() → str | None

get_roles() → set[str]

static goedel_implies(
    c1: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    c2: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
) → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept

static kleene_dienes_implies(
    c1: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    c2: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
) → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept

static lukasiewicz_implies(
    c1: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    c2: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
) → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
```

```

replace(
    a: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
) → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept

static zadeh_implies(
    c1: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    c2: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
) → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept

name: str

```

**GoedelImplies**

**KleeneDienesImplies**

**LukasiewiczImplies**

**ZadehImplies**

**fuzzy\_dl\_owl2.fuzzydl.concept.interface**

**Submodules**

**fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_concept\_interface**

**Classes**

<i>HasConceptInterface</i>	Helper class that provides a standard way to create an ABC using
----------------------------	--

## Module Contents

**class HasConceptInterface**(concept: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept)

Bases: abc.ABC

Helper class that provides a standard way to create an ABC using inheritance.

**property curr\_concept:** *fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*

**fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_concepts\_interface**

**Classes**

<i>HasConceptsInterface</i>	Helper class that provides a standard way to create an ABC using
-----------------------------	--

## Module Contents

**class HasConceptsInterface**(concepts: Iterable[fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept])

Bases: abc.ABC

Helper class that provides a standard way to create an ABC using inheritance.

**property concepts:** list[*fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*]

**fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_role\_concept\_interface****Classes**

<i>HasRoleConceptInterface</i>	Helper class that provides a standard way to create an ABC using
--------------------------------	--

**Module Contents**

```
class HasRoleConceptInterface(role: str, concept: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept)
    Bases: fuzzy_dl_owl2.fuzzydl.concept.interface.has_role_interface.HasRoleInterface,
fuzzy_dl_owl2.fuzzydl.concept.interface.has_concept_interface.HasConceptInterface,
    abc.ABC
    Helper class that provides a standard way to create an ABC using inheritance.
```

**fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_role\_interface****Classes**

<i>HasRoleInterface</i>	Helper class that provides a standard way to create an ABC using
-------------------------	--

**Module Contents**

```
class HasRoleInterface(role: str)
    Bases: abc.ABC
    Helper class that provides a standard way to create an ABC using inheritance.
    property role: str
```

**fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_value\_interface****Classes**

<i>HasValueInterface</i>	Helper class that provides a standard way to create an ABC using
--------------------------	--

**Module Contents**

```
class HasValueInterface(role: str, value: Any)
    Bases: fuzzy_dl_owl2.fuzzydl.concept.interface.has_role_interface.HasRoleInterface,
    abc.ABC
    Helper class that provides a standard way to create an ABC using inheritance.
    property value: Any
```

**fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_weighted\_concepts\_interface****Classes**

*HasWeightedConceptsInterface*

Helper class that provides a standard way to create an ABC using

## Module Contents

**class HasWeightedConceptsInterface(***weights: Iterable[float] | None,**concepts: Iterable[fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept],***)**Bases: *fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_concepts\_interface.HasConceptsInterface*, *abc.ABC*

Helper class that provides a standard way to create an ABC using inheritance.

**property weights: list[float] | None****fuzzy\_dl\_owl2.fuzzydl.concept.modified**

## Submodules

**fuzzy\_dl\_owl2.fuzzydl.concept.modified.linearly\_modified\_concept**

## Classes

*LinearlyModifiedConcept*

Fuzzy concept modified with a linear modifier

## Module Contents

**class LinearlyModifiedConcept(c: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept, mod:**  
*fuzzy\_dl\_owl2.fuzzydl.modifier.modifier.Modifier)*Bases: *fuzzy\_dl\_owl2.fuzzydl.concept.modified.modified\_concept.ModifiedConcept*

Fuzzy concept modified with a linear modifier

**\_\_and\_\_(value: Self) → Self****\_\_hash\_\_() → int****\_\_neg\_\_() → fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept****\_\_or\_\_(value: Self) → Self****clone() → Self****replace(a: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept, c:**  
*fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept) → Self***fuzzy\_dl\_owl2.fuzzydl.concept.modified.modified\_concept**

## Classes

*ModifiedConcept*

Modified fuzzy concept.

## Module Contents

```
class ModifiedConcept(c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept, mod:
    fuzzy_dl_owl2.fuzzydl.modifier.modifier.Modifier)

    Bases: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept, fuzzy_dl_owl2.fuzzydl.concept.
interface.has_concept_interface.HasConceptInterface, abc.ABC

    Modified fuzzy concept.

    __and__() → Self

    __neg__() → Self

    __or__() → Self

    __repr__() → str

    __str__() → str

    compute_atomic_concepts() → set[Self]

    compute_name() → str | None

    get_roles() → set[str]

    is_concrete() → bool

    replace(
        concept1: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
        concept2: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    ) → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept

    property modifier: fuzzy_dl_owl2.fuzzydl.modifier.modifier.Modifier
```

**fuzzy\_dl\_owl2.fuzzydl.concept.modified.triangularly\_modified\_concept**

## Classes

<i>TriangularlyModifiedConcept</i>	Fuzzy concept modified with a triangular modifier.
------------------------------------	--

## Module Contents

```
class TriangularlyModifiedConcept(
    c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    mod: fuzzy_dl_owl2.fuzzydl.modifier.modifier.Modifier,
)

    Bases: fuzzy_dl_owl2.fuzzydl.concept.modified.modified_concept.ModifiedConcept

    Fuzzy concept modified with a triangular modifier.

    __and__(value: Self) → Self

    __hash__() → int

    __neg__() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept

    __or__(value: Self) → Self
```

**clone()** → Self

**replace**(  
     *a*: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept,  
     *c*: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept,  
 ) → fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept

**fuzzy\_dl\_owl2.fuzzydl.concept.negated\_nominal**

## Classes

<i>NegatedNominal</i>	Negated nominal concept. Only used in range restrictions for the moment.
-----------------------	--

## Module Contents

```
class NegatedNominal(ind_name: str)
    Bases: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
    Negated nominal concept. Only used in range restrictions for the moment.
    __and__(value: Self) → Self
    __hash__() → int
    __neg__() → Self
    __or__(value: Self) → Self
    clone() → Self
    compute_name() → str | None
    property ind_name: str
    name: str = '(not { Uninferable } )'
```

**fuzzy\_dl\_owl2.fuzzydl.concept.operator\_concept**

## Attributes

<i>And</i>
<i>GoedelAnd</i>
<i>GoedelOr</i>
<i>LukasiewiczAnd</i>
<i>LukasiewiczOr</i>
<i>Not</i>
<i>Or</i>

## Classes

<i>OperatorConcept</i>	Defines a logic operator concept defined as AND, OR or NOT of concepts.
------------------------	---

## Module Contents

```
class OperatorConcept(  
    c_type: fuzzy_dl_owl2.fuzzydl.util.constants.ConceptType,  
    concepts: Iterable[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept],  
)  
  
    Bases: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept, fuzzy_dl_owl2.fuzzydl.concept.  
interface.has_concepts_interface.HasConceptsInterface  
  
    Defines a logic operator concept defined as AND, OR or NOT of concepts.  
  
    __and__(value: Self) → Self  
  
    __eq__(value: Self) → bool  
  
    __hash__() → int  
  
    __ne__(value: Self) → bool  
  
    __neg__() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept  
  
    __or__(value: Self) → Self  
  
    static and_(*concepts: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) →  
        fuzzy_dl_owl2.fuzzydl.concept.concept.Concept  
  
    clone() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept  
  
    compute_atomic_concepts() → set[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept]  
  
    compute_name() → str | None  
  
    de_morgan() → Self  
  
    distribute(c_type: fuzzy_dl_owl2.fuzzydl.util.constants.ConceptType) → Self  
  
    get_atom() → Self | None  
  
    get_atoms() → list[Self]  
  
    get_clauses(is_type: Callable) → list[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept]  
  
    get_roles() → set[str]  
  
    static goedel_and(*concepts: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) →  
        fuzzy_dl_owl2.fuzzydl.concept.concept.Concept  
  
    static goedel_or(*concepts: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) →  
        fuzzy_dl_owl2.fuzzydl.concept.concept.Concept  
  
    static is_and(c_type: fuzzy_dl_owl2.fuzzydl.util.constants.ConceptType) → bool  
  
    is_atomic() → bool  
  
    is_complemented_atomic() → bool  
  
    is_concrete() → bool
```



```

static is_not_at_least_value(op: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → bool
static is_not_at_most_value(op: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → bool
static is_not_choquet(op: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → bool
static is_not_concrete(op: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → bool
static is_not_exact_value(op: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → bool
static is_not_ext_neg_threshold(op: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → bool
static is_not_ext_pos_threshold(op: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → bool
static is_not_fuzzy_number(op: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → bool
static is_not_goedel_implies(op: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → bool
static is_not_has_value(op: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → bool
static is_not_modified(op: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → bool
static is_not_neg_threshold(op: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → bool
static is_not_owa(op: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → bool
static is_not_pos_threshold(op: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → bool
static is_not_qowa(op: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → bool
static is_not_quasi_sugeno(op: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → bool
static is_not_self(op: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → bool
static is_not_sigma_concept(op: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → bool
static is_not_sugeno(op: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → bool
static is_not_type(op: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept, c_type:
    fuzzy_dl_owl2.fuzzydl.util.constants.ConceptType) → bool
static is_not_weighted(op: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → bool
static is_not_weighted_max(op: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → bool
static is_not_weighted_min(op: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → bool
static is_not_weighted_sum(op: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → bool
static is_not_weighted_sum_zero(op: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → bool
static is_not_zadeh_implies(op: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → bool
static is_or(c_type: fuzzy_dl_owl2.fuzzydl.util.constants.ConceptType) → bool
is_simplified() → bool

```

**This function check if current formula is simplified, i.e., if:**

- The only negated elements are literal of kind ( $\sim A$ ), where  $A$  is an AtomicProposition
- **The OR operator is between:**
  - Two literals  $\Rightarrow A \mid B$
  - One literal and a AND  $\Rightarrow A \mid (B \ \& \ C) - (A \ \& \ B) \mid C$
  - Two (or more) OR  $\Rightarrow (A \ \& \ B) \mid (C \ \& \ D) \mid (E \ \& \ F)$

- The AND operator is between:
  - Two literals  $\Rightarrow A \ \& \ B$
  - One literal and a OR  $\Rightarrow A \ \& \ (B \ | \ C) - (A \ | \ B) \ \& \ C$
  - Two (or more) AND  $\Rightarrow (A \ | \ B) \ \& \ (C \ | \ D) \ \& \ (E \ | \ F)$
- The only operators are AND, OR and NOT

```
static lukasiewicz_and(*concepts: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) →  
    fuzzy_dl_owl2.fuzzydl.concept.concept.Concept  
  
static lukasiewicz_or(*concepts: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) →  
    fuzzy_dl_owl2.fuzzydl.concept.concept.Concept  
  
normal_form(is_type: Callable) → Self  
  
static not_(concept: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) →  
    fuzzy_dl_owl2.fuzzydl.concept.concept.Concept  
  
static or_(*concepts: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) →  
    fuzzy_dl_owl2.fuzzydl.concept.concept.Concept  
  
reduce_double_negation() → Self  
  
reduce_idempotency(is_type: Callable) → Self  
  
reduce_quantifiers() → Self  
  
reduce_truth_values() → Self  
  
replace(  
    a: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,  
    c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,  
    ) → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept  
  
ABSORPTION_OPERATORS: list[fuzzy_dl_owl2.fuzzydl.util.constants.ConceptType]  
  
ALL_OPERATORS: list[fuzzy_dl_owl2.fuzzydl.util.constants.ConceptType]  
  
AND_OPERATORS: list[fuzzy_dl_owl2.fuzzydl.util.constants.ConceptType]  
  
BINARY_OPERATORS: list[fuzzy_dl_owl2.fuzzydl.util.constants.ConceptType]  
  
COMPLEMENT_LAW_OPERATORS: list[fuzzy_dl_owl2.fuzzydl.util.constants.ConceptType]  
  
DISTRIBUTIVE_OPERATORS: list[fuzzy_dl_owl2.fuzzydl.util.constants.ConceptType]  
  
OPERATORS: dict[fuzzy_dl_owl2.fuzzydl.util.constants.ConceptType,  
    fuzzy_dl_owl2.fuzzydl.util.constants.ConceptType]  
  
OR_OPERATORS: list[fuzzy_dl_owl2.fuzzydl.util.constants.ConceptType]  
  
property_concepts: list[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept]  
  
name = '(and )'  
  
type: fuzzy_dl_owl2.fuzzydl.util.constants.ConceptType
```

And

GoedelAnd

GoedelOr

LukasiewiczAnd

LukasiewiczOr

Not

Or

`fuzzy_dl_owl2.fuzzydl.concept.owa_concept`

## Classes

*OwaConcept*

Helper class that provides a standard way to create an ABC using

## Module Contents

**class** `OwaConcept`(*weights*: list[float], *concepts*: list[fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept])

Bases: `fuzzy_dl_owl2.fuzzydl.concept.concept.Concept`, `fuzzy_dl_owl2.fuzzydl.concept.interface.has_weighted_concepts_interface.HasWeightedConceptsInterface`

Helper class that provides a standard way to create an ABC using inheritance.

`__and__(value: Self) → Self`

`__hash__() → int`

`__neg__() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept`

`__or__(value: Self) → Self`

`clone() → Self`

`compute_atomic_concepts() → set[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept]`

`compute_name() → str`

`get_roles() → set[str]`

`replace(`  
     *a*: `fuzzy_dl_owl2.fuzzydl.concept.concept.Concept`,  
     *c*: `fuzzy_dl_owl2.fuzzydl.concept.concept.Concept`,  
`) → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept | None`

`fuzzy_dl_owl2.fuzzydl.concept.qowa_concept`

## Classes

*QowaConcept*

Quantified-guided OWA concept.

## Module Contents

**class** `QowaConcept`(

*quantifier*: `fuzzy_dl_owl2.fuzzydl.concept.concrete.fuzzy_concrete_concept.FuzzyConcreteConcept`,  
     *concepts*: list[fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept],

)

Bases: `fuzzy_dl_owl2.fuzzydl.concept.owa_concept.OwaConcept`

Quantified-guided OWA concept.

```
__and__(value: Self) → Self
__hash__() → int
__neg__() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
__or__(value: Self) → Self
clone() → Self
compute_name() → str
compute_weights(n: int) → None
replace(
    a: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
) → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept | None
name = '(q-owa Uninferable )'
property quantifier: fuzzy_dl_owl2.fuzzydl.concept.concrete.fuzzy_concrete_
concept.FuzzyConcreteConcept
type
```

**fuzzy\_dl\_owl2.fuzzydl.concept.quasi\_sugeno\_integral**

## Classes

<i>QsugenoIntegral</i>	Quasi Sugeno integral concept.
------------------------	--------------------------------

## Module Contents

```
class QsugenoIntegral(weights: list[float], concepts:
    list[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept])
    Bases: fuzzy_dl_owl2.fuzzydl.concept.sugeno_integral.SugenoIntegral
    Quasi Sugeno integral concept.
    __and__(value: Self) → Self
    __hash__() → int
    __neg__() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
    __or__(value: Self) → Self
    clone() → Self
    compute_name() → str
    replace(
        a: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
        c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    ) → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
    type
```

**fuzzy\_dl\_owl2.fuzzydl.concept.self\_concept****Classes**

<i>SelfConcept</i>	Helper class that provides a standard way to create an ABC using
--------------------	--

**Module Contents**

**class SelfConcept**(*role: str*)

Bases: *fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*, *fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_role\_interface.HasRoleInterface*

Helper class that provides a standard way to create an ABC using inheritance.

**\_\_and\_\_**(*value: Self*) → Self

**\_\_hash\_\_**() → int

**\_\_neg\_\_**() → *fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*

**\_\_or\_\_**(*value: Self*) → Self

**clone**()

**compute\_atomic\_concepts**() → set[*fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*]

**compute\_name**() → str | None

**get\_roles**() → set[str]

**replace**(  
*a: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*,  
*c: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*,  
) → *fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*

**static self**(*role: str*) → Self

**name** = '(self Uninferable)'

**fuzzy\_dl\_owl2.fuzzydl.concept.sigma\_concept****Classes**

<i>SigmaConcept</i>	Sigma-count concept.
---------------------	----------------------

**Module Contents**

**class SigmaConcept**(

*concept: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*,  
*role: str*,  
*individuals: list[fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual]*,  
*concrete\_concept: fuzzy\_dl\_owl2.fuzzydl.concept.concrete.fuzzy\_concrete\_concept.FuzzyConcreteConcept*,  
)

Bases: *fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*

Sigma-count concept.

```
__and__(value: Self) → Self
__hash__() → int
__neg__() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
__or__(value: Self) → Self
clone() → Self
compute_atomic_concepts() → set[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept]
compute_name() → str | None
get_concept() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
get_fuzzy_concept() → fuzzy_dl_owl2.fuzzydl.concept.concrete.fuzzy_concrete_
concept.FuzzyConcreteConcept
get_individuals() → list[fuzzy_dl_owl2.fuzzydl.individual.individual.Individual]
get_role() → str
get_roles() → set[str]
replace(
    a: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
) → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
concept: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
concrete_concept: fuzzy_dl_owl2.fuzzydl.concept.concrete.fuzzy_concrete_concept.
FuzzyConcreteConcept
individuals: list[fuzzy_dl_owl2.fuzzydl.individual.individual.Individual]
name: str = '(sigma-count Uninferable Uninferable {} Uninferable)'
role: str
```

**fuzzy\_dl\_owl2.fuzzydl.concept.sigma\_count**

## Classes

<i>SigmaCount</i>	Sigma-count pending tasks.
-------------------	----------------------------

## Module Contents

```
class SigmaCount(
    var: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,
    inds: list[fuzzy_dl_owl2.fuzzydl.individual.individual.Individual],
    role: str,
    concept: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
)
Sigma-count pending tasks.
__hash__() → int
```

```

__repr__() → str
__str__() → str
clone() → Self
get_concept() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
get_individual() → fuzzy_dl_owl2.fuzzydl.individual.individual.Individual
get_individuals() → list[fuzzy_dl_owl2.fuzzydl.individual.individual.Individual]
get_role() → str
get_variable() → fuzzy_dl_owl2.fuzzydl.milp.variable.Variable
concept: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
individual: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual
individuals: list[fuzzy_dl_owl2.fuzzydl.individual.individual.Individual]
role: str
variable: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable

```

**fuzzy\_dl\_owl2.fuzzydl.concept.string\_concept**

### Classes

<i>StringConcept</i>	Helper class that provides a standard way to create an ABC using
----------------------	--

### Module Contents

```

class StringConcept(name: str)
    Bases: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
    Helper class that provides a standard way to create an ABC using inheritance.
    __hash__() → int
    __neg__() → Self
    clone() → Self
    compute_atomic_concepts() → set[Self]
    compute_name() → str | None
    get_roles() → set[str]
    replace(a: Self, c: Self) → Self | None

```

**fuzzy\_dl\_owl2.fuzzydl.concept.sugeno\_integral**

### Classes

<i>SugenoIntegral</i>	Sugeno integral concept.
-----------------------	--------------------------

## Module Contents

**class SugenoIntegral**

**class SugenoIntegral**(weights: list[float] | None, concepts:  
list[fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept])

Bases: *fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept, fuzzy\_dl\_owl2.fuzzydl.concept.  
interface.has\_weighted\_concepts\_interface.HasWeightedConceptsInterface*

Sugeno integral concept.

**\_\_and\_\_**(value: Self) → Self

**\_\_hash\_\_**() → int

**\_\_neg\_\_**() → *fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*

**\_\_or\_\_**(value: Self) → Self

**clone**() → Self

**compute\_atomic\_concepts**() → set[fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept]

**compute\_name**() → str

**get\_roles**() → set[str]

**replace**(  
a: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept,  
c: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept,  
) → *fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*

**fuzzy\_dl\_owl2.fuzzydl.concept.threshold\_concept**

## Attributes

*NegThreshold*

*PosThreshold*

## Classes

*ThresholdConcept*

Helper class that provides a standard way to create an ABC using

## Module Contents

**class ThresholdConcept**(

c\_type: fuzzy\_dl\_owl2.fuzzydl.util.constants.ConceptType,  
c: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept,  
weight: float,

)

Bases: *fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept, fuzzy\_dl\_owl2.fuzzydl.concept.  
interface.has\_concept\_interface.HasConceptInterface*

Helper class that provides a standard way to create an ABC using inheritance.

**\_\_and\_\_**(value: Self) → Self



```

__hash__() → int
__neg__() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
__or__(value: Self) → Self
clone() → Self
compute_atomic_concepts() → set[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept]
compute_name() → str | None
get_roles() → set[str]
static neg_threshold(w: float, c: Self) → Self
static pos_threshold(w: float, c: Self) → Self
replace(
    a: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
) → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
name = '([>= Uninferable] Uninferable)'
property weight: float

```

NegThreshold

PosThreshold

*fuzzy\_dl\_owl2.fuzzydl.concept.truth\_concept*

Attributes

*BOTTOM*

*TOP*

Classes

*TruthConcept*

Helper class that provides a standard way to create an ABC using

Module Contents

**class TruthConcept**(c\_type: *fuzzy\_dl\_owl2.fuzzydl.util.constants.ConceptType*)

Bases: *fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*

Helper class that provides a standard way to create an ABC using inheritance.

**\_\_and\_\_**(value: Self) → Self

**\_\_eq\_\_**(value: Self) → bool

**\_\_hash\_\_**() → int

**\_\_ne\_\_**(value: Self) → bool

```
__neg__() → Self
__or__(value: Self) → Self
__rshift__(value: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) →
    fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
clone() → Self
compute_atomic_concepts() → set[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept]
compute_name() → str | None
get_atomic_concepts() → set[Self]
get_atoms() → list[Self]
static get_bottom()
get_roles() → set[str]
static get_top()
is_atomic() → bool
is_complemented_atomic() → bool
replace(
    a: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
) → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
name = '*top'
```

BOTTOM: [fuzzy\\_dl\\_owl2.fuzzydl.concept.concept.Concept](#)

TOP: [fuzzy\\_dl\\_owl2.fuzzydl.concept.concept.Concept](#)

[fuzzy\\_dl\\_owl2.fuzzydl.concept.value\\_concept](#)

## Classes

<a href="#">ValueConcept</a>	Helper class that provides a standard way to create an ABC using
------------------------------	--

## Module Contents

**class ValueConcept**(*c\_type*: [fuzzy\\_dl\\_owl2.fuzzydl.util.constants.ConceptType](#), *role*: *str*, *value*: *Any*)

Bases: [fuzzy\\_dl\\_owl2.fuzzydl.concept.concept.Concept](#), [fuzzy\\_dl\\_owl2.fuzzydl.concept.interface.has\\_value\\_interface.HasValueInterface](#)

Helper class that provides a standard way to create an ABC using inheritance.

```
__and__(value: Self) → Self
__hash__() → int
__neg__() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
__or__(value: Self) → Self
static at_least_value(role: str, o: Any) → Self
```

```

static at_most_value(role: str, o: Any) → Self
clone() → Self
compute_atomic_concepts() → set[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept]
compute_name() → str | None
static exact_value(role: str, o: Any) → Self
get_roles() → set[str]
replace(
    a: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
) → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
name

```

## fuzzy\_dl\_owl2.fuzzydl.concept.weighted\_concept

### Classes

<i>WeightedConcept</i>	Helper class that provides a standard way to create an ABC using
------------------------	--

### Module Contents

```

class WeightedConcept(weight: float, c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept)
    Bases: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept, fuzzy_dl_owl2.fuzzydl.concept.interface.has_concept_interface.HasConceptInterface
    Helper class that provides a standard way to create an ABC using inheritance.
    __and__(value: Self) → Self
    __hash__() → int
    __neg__() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
    __or__(value: Self) → Self
    clone() → Self
    compute_atomic_concepts() → set[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept]
    compute_name() → str | None
    get_roles() → set[str]
    replace(
        a: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
        c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    ) → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
    name = '(Uninferable Uninferable)'
    property weight: float

```

**fuzzy\_dl\_owl2.fuzzydl.concept.weighted\_max\_concept****Classes**

<i>WeightedMaxConcept</i>	Helper class that provides a standard way to create an ABC using
---------------------------	--

**Module Contents**

```
class WeightedMaxConcept(weights: list[float], concepts:
    list[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept])

Bases: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept, fuzzy_dl_owl2.fuzzydl.concept.
interface.has_weighted_concepts_interface.HasWeightedConceptsInterface

Helper class that provides a standard way to create an ABC using inheritance.

__and__(value: Self) → Self

__hash__() → int

__neg__() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept

__or__(value: Self) → Self

clone() → Self

compute_atomic_concepts() → set[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept]

compute_name() → str

get_roles() → set[str]

replace(
    a: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
) → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept

name = '(w-max )'
```

**fuzzy\_dl\_owl2.fuzzydl.concept.weighted\_min\_concept****Classes**

<i>WeightedMinConcept</i>	Helper class that provides a standard way to create an ABC using
---------------------------	--

**Module Contents**

```
class WeightedMinConcept(weights: list[float], concepts:
    list[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept])

Bases: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept, fuzzy_dl_owl2.fuzzydl.concept.
interface.has_weighted_concepts_interface.HasWeightedConceptsInterface

Helper class that provides a standard way to create an ABC using inheritance.

__and__(value: Self) → Self

__hash__() → int
```

```

__neg__() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
__or__(value: Self) → Self
clone() → Self
compute_atomic_concepts() → set[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept]
compute_name() → str
get_roles() → set[str]
replace(
    a: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
) → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
name = '(w-min )'

```

## fuzzy\_dl\_owl2.fuzzydl.concept.weighted\_sum\_concept

### Classes

<i>WeightedSumConcept</i>	Helper class that provides a standard way to create an ABC using
---------------------------	--

### Module Contents

```

class WeightedSumConcept(weights: list[float], concepts:
    list[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept])
    Bases: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept, fuzzy_dl_owl2.fuzzydl.concept.
    interface.has_weighted_concepts_interface.HasWeightedConceptsInterface
    Helper class that provides a standard way to create an ABC using inheritance.
    __and__(value: Self) → Self
    __hash__() → int
    __neg__() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
    __or__(value: Self) → Self
    clone() → Self
    compute_atomic_concepts() → set[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept]
    compute_name() → str
    get_roles() → set[str]
    replace(
        a: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
        c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    ) → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
    name = '(w-sum )'

```

## fuzzy\_dl\_owl2.fuzzydl.concept.weighted\_sum\_zero\_concept

## Classes

<i>WeightedSumZeroConcept</i>	Helper class that provides a standard way to create an ABC using
-------------------------------	--

## Module Contents

```

class WeightedSumZeroConcept(weights: list[float], concepts:
    list[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept])
    Bases: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept, fuzzy_dl_owl2.fuzzydl.concept.
    interface.has_weighted_concepts_interface.HasWeightedConceptsInterface
    Helper class that provides a standard way to create an ABC using inheritance.
    __and__(value: Self) → Self
    __hash__() → int
    __neg__() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
    __or__(value: Self) → Self
    clone() → Self
    compute_atomic_concepts() → set[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept]
    compute_name() → str
    get_roles() → set[str]
    replace(
        a: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
        c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    ) → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
    name = '(w-sum-zero )'

```

## fuzzy\_dl\_owl2.fuzzydl.concept\_equivalence

## Classes

<i>ConceptEquivalence</i>	Concept equivalence axiom
---------------------------	---------------------------

## Module Contents

```

class ConceptEquivalence(c1: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept, c2:
    fuzzy_dl_owl2.fuzzydl.concept.concept.Concept)
    Concept equivalence axiom
    clone() → Self
    get_c1() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
    get_c2() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
    c1: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
    c2: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept

```

## fuzzy\_dl\_owl2.fuzzydl.concrete\_feature

## Classes

<i>ConcreteFeature</i>
------------------------

## Module Contents

```

class ConcreteFeature(name: str)
class ConcreteFeature(name: str, is_boolean: bool)
class ConcreteFeature(name: str, k1: int, k2: int)
class ConcreteFeature(name: str, k1: float, k2: float)

    __repr__() → str
    __str__() → str
    clone() → Self
    get_k1() → float | int | None
    get_k2() → float | int | None
    get_name() → str
    get_type() → fuzzy_dl_owl2.fuzzydl.util.constants.ConcreteFeatureType
    set_range(k1: float | int | None, k2: float | int | None) → None
    set_type(new_type: fuzzy_dl_owl2.fuzzydl.util.constants.ConcreteFeatureType) → None

```

## fuzzy\_dl\_owl2.fuzzydl.degree

## Submodules

## fuzzy\_dl\_owl2.fuzzydl.degree.degree

## Classes

<i>Degree</i>	Helper class that provides a standard way to create an ABC using
---------------	--

## Module Contents

```

class Degree
    Bases: abc.ABC

    Helper class that provides a standard way to create an ABC using inheritance.

    abstractmethod __eq__(degree: Self) → bool
    __ne__(value: Self) → bool
    __repr__() → str

```

```
abstractmethod __str__() → str

abstractmethod add_to_expression(
    expression: fuzzy_dl_owl2.fuzzydl.milp.expression.Expression,
) → fuzzy_dl_owl2.fuzzydl.milp.expression.Expression

abstractmethod clone() → Self

abstractmethod create_inequality_with_degree_rhs(
    expression: fuzzy_dl_owl2.fuzzydl.milp.expression.Expression,
    inequation_type: fuzzy_dl_owl2.fuzzydl.util.constants.InequalityType,
) → fuzzy_dl_owl2.fuzzydl.milp.inequation.Inequation

static get_degree(value) → Self

    Abstractmethod

abstractmethod is_number_not_one() → bool

abstractmethod is_number_zero() → bool

abstractmethod is_numeric() → bool

abstractmethod multiply_constant(double: float) →
    fuzzy_dl_owl2.fuzzydl.milp.expression.Expression

abstractmethod subtract_from_expression(
    expression: fuzzy_dl_owl2.fuzzydl.milp.expression.Expression,
) → fuzzy_dl_owl2.fuzzydl.milp.expression.Expression
```

## fuzzy\_dl\_owl2.fuzzydl.degree.degree\_expression

### Classes

<i>DegreeExpression</i>	Helper class that provides a standard way to create an ABC using
-------------------------	--

### Module Contents

```
class DegreeExpression(expr: fuzzy_dl_owl2.fuzzydl.milp.expression.Expression)
    Bases: fuzzy_dl_owl2.fuzzydl.degree.degree.Degree
    Helper class that provides a standard way to create an ABC using inheritance.

    __eq__(d: fuzzy_dl_owl2.fuzzydl.degree.degree.Degree) → bool

    __str__() → str

    add_to_expression(expr: fuzzy_dl_owl2.fuzzydl.milp.expression.Expression) →
        fuzzy_dl_owl2.fuzzydl.milp.expression.Expression

    clone() → Self

    create_inequality_with_degree_rhs(
        expr: fuzzy_dl_owl2.fuzzydl.milp.expression.Expr,
        inequality_type: fuzzy_dl_owl2.fuzzydl.util.constants.InequalityType,
    ) → fuzzy_dl_owl2.fuzzydl.milp.inequation.Inequation

    static get_degree(value: fuzzy_dl_owl2.fuzzydl.milp.expression.Expression) → Self

    get_expression() → fuzzy_dl_owl2.fuzzydl.milp.expression.Expression
```



```

is_number_not_one() → bool

is_number_zero() → bool

is_numeric() → bool

multiply_constant(constant: float) → fuzzy_dl_owl2.fuzzydl.milp.expression.Expression

subtract_from_expression(
    expr: fuzzy_dl_owl2.fuzzydl.milp.expression.Expression,
) → fuzzy_dl_owl2.fuzzydl.milp.expression.Expression

expr: fuzzy_dl_owl2.fuzzydl.milp.expression.Expression

```

**fuzzy\_dl\_owl2.fuzzydl.degree.degree\_numeric**

## Classes

<i>DegreeNumeric</i>	Helper class that provides a standard way to create an ABC using
----------------------	--

## Module Contents

```

class DegreeNumeric(numeric: float)
    Bases: fuzzy_dl_owl2.fuzzydl.degree.degree.Degree
    Helper class that provides a standard way to create an ABC using inheritance.
    __eq__(d: fuzzy_dl_owl2.fuzzydl.degree.degree.Degree) → bool
    __str__() → str
    add_to_expression(expr: fuzzy_dl_owl2.fuzzydl.milp.expression.Expression) →
        fuzzy_dl_owl2.fuzzydl.milp.expression.Expression
    clone() → Self
    create_inequality_with_degree_rhs(
        expr: fuzzy_dl_owl2.fuzzydl.milp.expression.Expression,
        inequation_type: fuzzy_dl_owl2.fuzzydl.util.constants.InequalityType,
    ) → fuzzy_dl_owl2.fuzzydl.milp.inequation.Inequation
    static get_degree(value: float) → Self
    get_numerical_value() → float
    static get_one() → Self
    is_number_not_one() → bool
    is_number_zero() → bool
    is_numeric() → bool
    multiply_constant(constant: float) → fuzzy_dl_owl2.fuzzydl.milp.expression.Expression
    subtract_from_expression(
        expr: fuzzy_dl_owl2.fuzzydl.milp.expression.Expression,
    ) → fuzzy_dl_owl2.fuzzydl.milp.expression.Expression
    value: float

```

## fuzzy\_dl\_owl2.fuzzydl.degree.degree\_variable

## Classes

*DegreeVariable*

Helper class that provides a standard way to create an ABC using

## Module Contents

```

class DegreeVariable(variable: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable)
    Bases: fuzzy_dl_owl2.fuzzydl.degree.degree.Degree
    Helper class that provides a standard way to create an ABC using inheritance.
    __eq__(degree: fuzzy_dl_owl2.fuzzydl.degree.degree.Degree) → bool
    __str__() → str
    add_to_expression(expr: fuzzy_dl_owl2.fuzzydl.milp.expression.Expression) →
        fuzzy_dl_owl2.fuzzydl.milp.expression.Expression
    clone() → Self
    create_inequality_with_degree_rhs(
        expr: fuzzy_dl_owl2.fuzzydl.milp.expression.Expression,
        inequality_type: fuzzy_dl_owl2.fuzzydl.util.constants.InequalityType,
    ) → fuzzy_dl_owl2.fuzzydl.milp.inequation.Inequation
    static get_degree(value: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable) → Self
    get_variable() → fuzzy_dl_owl2.fuzzydl.milp.variable.Variable
    is_number_not_one() → bool
    is_number_zero() → bool
    is_numeric() → bool
    multiply_constant(constant: float) → fuzzy_dl_owl2.fuzzydl.milp.expression.Expression
    subtract_from_expression(
        expr: fuzzy_dl_owl2.fuzzydl.milp.expression.Expression,
    ) → fuzzy_dl_owl2.fuzzydl.milp.expression.Expression
    variable: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable

```

## fuzzy\_dl\_owl2.fuzzydl.domain\_axiom

## Classes

*DomainAxiom*

Role domain axiom

## Module Contents

```

class DomainAxiom(role: str, concept: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept)
    Role domain axiom
    concept: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
    role: str

```

**fuzzy\_dl\_owl2.fuzzydl.exception****Submodules****fuzzy\_dl\_owl2.fuzzydl.exception.fuzzy\_ontology\_exception****Exceptions**

<i>FuzzyOntologyException</i>	Common base class for all non-exit exceptions.
-------------------------------	--

**Module Contents**

**exception FuzzyOntologyException**(*message: str*)

Bases: Exception

Common base class for all non-exit exceptions.

**fuzzy\_dl\_owl2.fuzzydl.exception.inconsistent\_ontology\_exception****Exceptions**

<i>InconsistentOntologyException</i>	Common base class for all non-exit exceptions.
--------------------------------------	--

**Module Contents**

**exception InconsistentOntologyException**(*message: str*)

Bases: Exception

Common base class for all non-exit exceptions.

**fuzzy\_dl\_owl2.fuzzydl.feature\_function****Classes**

<i>FeatureFunction</i>	Function involving several features.
------------------------	--------------------------------------

**Module Contents**

**class FeatureFunction**(*feature: Self*)

**class FeatureFunction**(*feature: str*)

**class FeatureFunction**(*n: float*)

**class FeatureFunction**(*feature: list[Self]*)

**class FeatureFunction**(*feature1: Self, feature2: Self*)

**class FeatureFunction**(*n: float, feature: Self*)

Function involving several features.

**\_\_repr\_\_**() → str

**\_\_str\_\_**() → str

**get\_features**() → set[str]

Gets an array of features that take part in the function.

**get\_number**() → float

**get\_type()** → *fuzzy\_dl\_owl2.fuzzydl.util.constants.FeatureFunctionType*

**to\_expression()**  
    *a: fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual,*  
    *milp: fuzzy\_dl\_owl2.fuzzydl.milp.milp\_helper.MILPHelper,*  
    *) → fuzzy\_dl\_owl2.fuzzydl.milp.expression.Expression | None*  
    Gets an array of features that take part in the function.

**fuzzy\_dl\_owl2.fuzzydl.fuzzydl\_to\_owl2**

## Classes

---

<i>FuzzydlToOwl2</i>	Convert FuzzyDL to OWL2
----------------------	-------------------------

---

## Functions

---

<i>main()</i>
---------------

---

## Module Contents

```
class FuzzydlToOwl2(input_file: str, output_file: str, base_iri: str =  
    'http://www.semanticweb.org/ontologies/fuzzydl_ontology#')  
    Convert FuzzyDL to OWL2  
  
    add_entity_annotation(annotation: str, entity: pyowl2.abstracts.entity.OWLEntity) → None  
        Add annotation to an entity  
  
    add_ontology_annotation(annotation: str) → None  
        Add annotation to the ontology  
  
    annotate_gci(gci: fuzzy_dl_owl2.fuzzydl.general_concept_inclusion.GeneralConceptInclusion) →  
        None  
        Annotate a General Concept Inclusion (GCI)  
  
    annotate_pcd(  
        c1: pyowl2.abstracts.class_expression.OWLClassExpression,  
        pcd: fuzzy_dl_owl2.fuzzydl.primitive_concept_definition.PrimitiveConceptDefinition,  
    ) → None  
        Annotate a Primitive Concept Definition (PCD)  
  
    annotation_property_iri(o: object) → pyowl2.base.iri.IRI  
        Convert datatype to IRI string  
  
    class_iri(o: object) → pyowl2.base.iri.IRI  
        Convert class to IRI string  
  
    data_property_iri(o: object) → pyowl2.base.iri.IRI  
        Convert data property to IRI string  
  
    datatype_iri(o: object) → pyowl2.base.iri.IRI  
        Convert datatype to IRI string  
  
    exist_data_property(role: str) → bool  
        Check if a data property exists
```

```

exist_object_property(role: str) → bool
    Check if an object property exists

get_annotations_for_axiom(
    value: float | fuzzy_dl_owl2.fuzzydl.degree.degree_numeric.DegreeNumeric,
) → set[pyowl2.base.annotation.OWLAnnotation]
    Get annotations for an axiom with degree

get_base(c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) →
    pyowl2.abstracts.class_expression.OWLClassExpression
    Get the base class for a concept

get_class(name: str) → pyowl2.abstracts.class_expression.OWLClassExpression

get_class(c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) →
    pyowl2.abstracts.class_expression.OWLClassExpression
    Get or create an OWL class

get_data_property(
    role: str,
) → pyowl2.expressions.data_property.OWLDataProperty | pyowl2.expressions.object_prop-
    erty.OWLObjectProperty
    Get or create a data property

get_individual(name: str) → pyowl2.individual.named_individual.OWLNamedIndividual
    Get or create a named individual

get_new_atomic_class(name: str) → pyowl2.abstracts.class_expression.OWLClassExpression
    Get or create a new atomic class

get_object_property(
    role: str,
) → pyowl2.expressions.data_property.OWLDataProperty | pyowl2.expressions.object_prop-
    erty.OWLObjectProperty
    Get or create an object property

individual_iri(o: object) → pyowl2.base.iri.IRI
    Convert individual object to IRI string

iri(o: object, iri_type: type = OWLClass) → pyowl2.base.iri.IRI
    Convert object to IRI string

object_property_iri(o: object) → pyowl2.base.iri.IRI
    Convert object property to IRI string

run() → None
    Execute the conversion process

to_owl_annotation(annotation: str) → pyowl2.base.annotation.OWLAnnotation
    Convert a string to an OWL annotation

concepts: dict[str, pyowl2.abstracts.class_expression.OWLClassExpression]

datatypes: dict[str, pyowl2.base.datatype.OWLDatatype]

fuzzyLabel: pyowl2.base.annotation_property.OWLAnnotationProperty

input_FDL: str

modifiers: dict[str, pyowl2.base.datatype.OWLDatatype]

num_classes: int = 0

```

```
ontology: pyowl2.ontology.OWLontology
ontology_iri: pyowl2.base.iri.IRI
ontology_path: str = 'Uninferable#'
output_FOWL: str
```

```
main()
```

```
fuzzy_dl_owl2.fuzzydl.general_concept_inclusion
```

## Classes

<i>GeneralConceptInclusion</i>	General concept inclusion axiom.
--------------------------------	----------------------------------

## Module Contents

```
class GeneralConceptInclusion(
    subsumer: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    subsumed: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    degree: fuzzy_dl_owl2.fuzzydl.degree.degree.Degree,
    type_: fuzzy_dl_owl2.fuzzydl.util.constants.LogicOperatorType,
)
    General concept inclusion axiom.
    __eq__(other: Self) → bool
    __ge__(other: Self) → bool
    __gt__(other: Self) → bool
    __hash__() → int
    __le__(other: Self) → bool
    __lt__(other: Self) → bool
    __ne__(other: Self) → bool
    __repr__() → str
    __str__() → str
    clone() → Self
    get_degree() → fuzzy_dl_owl2.fuzzydl.degree.degree.Degree
    get_subsumed() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
    get_subsumer() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
    get_type() → fuzzy_dl_owl2.fuzzydl.util.constants.LogicOperatorType
    set_degree(deg: fuzzy_dl_owl2.fuzzydl.degree.degree.Degree) → None
    set_subsumed(new_concept: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → None
    set_subsumer(new_concept: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → None
    degree: fuzzy_dl_owl2.fuzzydl.degree.degree.Degree
```

```

subsumed: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
subsumer: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
type: fuzzy_dl_owl2.fuzzydl.util.constants.LogicOperatorType

```

**fuzzy\_dl\_owl2.fuzzydl.individual**

**Submodules**

**fuzzy\_dl\_owl2.fuzzydl.individual.created\_individual**

**Classes**

*CreatedIndividual*

## Module Contents

```

class CreatedIndividual(
    name: str,
    parent: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual | None = None,
    role_name: str | None = None,
)
class CreatedIndividual(name: str)
    Bases: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual
    __eq__(value: Self) → bool
    __ge__(value: Self) → bool
    __gt__(value: Self) → bool
    __hash__() → int
    __le__(value: Self) → bool
    __lt__(value: Self) → bool
    __ne__(value: Self) → bool
    __str__() → str
    clone() → Self
    clone_special_attributes(ind: Self) → None
    get_depth() → int
    get_integer_id() → int
    get_parent() → fuzzy_dl_owl2.fuzzydl.individual.individual.Individual | None
    get_parent_name() → str
    get_representative_if_exists(
        type: fuzzy_dl_owl2.fuzzydl.util.constants.InequalityType,
        f_name: str,
        f: fuzzy_dl_owl2.fuzzydl.concept.concrete.fuzzy_number.triangular_fuzzy_number.TriangularFuzzyNumber,
    ) → Self | None

```

Return *b* individual *p* with *b* representative of *b* set of individuals if it exists. Given *b* fuzzy number *F*, *b* representative individual is the set of individuals that are greater or equal (or less or equal) than *F*. The representative individual is related to *p* via *b* concrete feature *f*.

**Parameters**

- **type** (*InequalityType*) – Type of the representative individual (GREATER\_EQUAL, LESS\_EQUAL)
- **f\_name** (*str*) – Name of the feature for which the individual is *b* filler
- **f** (*TriangularFuzzyNumber*) – Fuzzy number

**Returns**

A new individual with *b* representative individual

**Return type**

*Optional[Self]*

**get\_role\_name()** → *str*

**individual\_set\_intersection\_of**(  
    *set1: sortedcontainers.SortedSet[Self],*  
    *set2: sortedcontainers.SortedSet[Self],*  
) → *sortedcontainers.SortedSet[Self]*  
Gets the intersection of two concept labels.

**is\_blockable()** → *bool*

**is\_concrete()** → *bool*

**mark\_indirectly\_blocked()** → *None*  
Marks the subtree of a node as indirectly blocked

**set\_concrete\_individual()** → *None*  
Sets that the individual is concrete.

**fuzzy\_dl\_owl2.fuzzydl.individual.individual**

**Classes**

*Individual*

**Module Contents**

```
class Individual(name: str)

    __eq__(value: Self) → bool
    __ne__(value: Self) → bool
    __repr__() → str
    __str__() → str

    add_concept(c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → None
    add_concrete_restriction(f_name: str, ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion)
        → None
    Adds a negated datatype restriction to the individual.
```



```

add_to_nominal_list(ind_name: str) → None

clone() → Self

clone_attributes(ind: Self) → None

get_concepts() → set[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept]

get_nominal_list() → set[str]

abstractmethod get_representative_if_exists(
    type: fuzzy_dl_owl2.fuzzydl.util.constants.RepresentativeIndividualType,
    f_name: str,
    f: fuzzy_dl_owl2.fuzzydl.concept.concrete.fuzzy_number.triangular_fuzzy_number.TriangularFuzzyNumber,
)

is_blockable() → bool

prune() → None

set_label(ind_name: str) → None

set_name(name: str) → None

DEFAULT_NAME: str = 'i'

concrete_role_restrictions: dict[str,
list[fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion]]

fillers_to_show: dict[str, set[str]]

list_of_concepts: set[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept]

name: str

nominal_list: set[str]

not_self_roles: set[str] = []

role_relations: dict[str, list[fuzzy_dl_owl2.fuzzydl.relation.Relation]]

role_restrictions: dict[str,
list[fuzzy_dl_owl2.fuzzydl.restriction.restriction.Restriction]]

```

**fuzzy\_dl\_owl2.fuzzydl.individual.representative\_individual**

## Classes

<i>RepresentativeIndividual</i>	New concrete individual being a representative of a set of individuals.
---------------------------------	---

## Module Contents

```

class RepresentativeIndividual(
    c_type: fuzzy_dl_owl2.fuzzydl.util.constants.RepresentativeIndividualType,
    f_name: str,
    f: fuzzy_dl_owl2.fuzzydl.concept.concrete.fuzzy_number.triangular_fuzzy_number.TriangularFuzzyNumber,
    ind: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,
)

```

New concrete individual being a representative of a set of individuals. Given an individual  $p$  and a fuzzy number  $F$ , a representative individual is the set of individuals that are greater or equal (or less or equal) than  $F$ . Then,  $p$  is related to the representative individual in some way.

**get\_feature\_name()** → str

**get\_fuzzy\_number()** → *fuzzy\_dl\_owl2.fuzzydl.concept.concrete.fuzzy\_number.triangular\_fuzzy\_number.TriangularFuzzyNumber*

**get\_individual()** → *fuzzy\_dl\_owl2.fuzzydl.individual.created\_individual.CreatedIndividual*

**get\_type()** → *fuzzy\_dl\_owl2.fuzzydl.util.constants.RepresentativeIndividualType*

**f:** *fuzzy\_dl\_owl2.fuzzydl.concept.concrete.fuzzy\_number.triangular\_fuzzy\_number.TriangularFuzzyNumber*

**f\_name:** str

**ind:** *fuzzy\_dl\_owl2.fuzzydl.individual.created\_individual.CreatedIndividual*

**type:** *fuzzy\_dl\_owl2.fuzzydl.util.constants.RepresentativeIndividualType*

**fuzzy\_dl\_owl2.fuzzydl.knowledge\_base**

### Classes

<i>ClassicalSolver</i>	Solver for classical logic semantics.
<i>CreatedIndividualHandler</i>	
<i>DatatypeReasoner</i>	
<i>IndividualHandler</i>	
<i>KnowledgeBase</i>	
<i>LukasiewiczSolver</i>	
<i>ZadehSolver</i>	Solver for Zadeh fuzzy logic semantics.

### Module Contents

#### class ClassicalSolver

Solver for classical logic semantics.

**static solve\_all**(  
    *rel*: fuzzy\_dl\_owl2.fuzzydl.relation.Relation,  
    *restrict*: fuzzy\_dl\_owl2.fuzzydl.restriction.restriction.Restriction,  
    *kb*: KnowledgeBase,  
) → None

Solves a universal restriction fuzzy assertion with respect to a reference fuzzy KB.

**static solve\_and**(*ass*: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion, *kb*: KnowledgeBase) → None

Solves a conjunction fuzzy assertion with respect to a reference fuzzy KB.

**static solve\_or**(*ass*: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion, *kb*: KnowledgeBase) → None

Solves a disjunction fuzzy assertion with respect to a reference fuzzy KB.

**static solve\_some**(*ass*: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion, *kb*: KnowledgeBase) → None

Solves a existential restriction fuzzy assertion with respect to a reference fuzzy KB.

### class CreatedIndividualHandler

**static get\_representative**(  
*current\_individual*: fuzzy\_dl\_owl2.fuzzydl.individual.created\_individual.CreatedIndividual,  
*type*: fuzzy\_dl\_owl2.fuzzydl.util.constants.InequalityType,  
*f\_name*: str,  
*f*: fuzzy\_dl\_owl2.fuzzydl.concept.concrete.fuzzy\_number.triangular\_fuzzy\_number.TriangularFuzzyNumber,  
*kb*: KnowledgeBase,  
) → fuzzy\_dl\_owl2.fuzzydl.individual.created\_individual.CreatedIndividual

Gets b individual p with b representative of b set of individuals. Given b fuzzy number F, b representative individual is the set of individuals that are greater or equal (or less or equal) than F. The representative individual is related to p via b concrete feature f.

#### Parameters

- **current\_individual** (CreatedIndividual) – The individual we want the representative
- **type** (InequalityType) – Type of the representative individual (GREATER\_EQUAL, LESS\_EQUAL)
- **f\_name** (str) – Name of the feature for which the individual is b filler
- **f** (TriangularFuzzyNumber) – Fuzzy number
- **kb** (KnowledgeBase) – KnowledgeBase

#### Returns

A new individual with b representative individual

#### Return type

*CreatedIndividual*

**static is\_blocked**(*current\_individual*:  
fuzzy\_dl\_owl2.fuzzydl.individual.created\_individual.CreatedIndividual, *kb*:  
KnowledgeBase) → bool

Gets if the individual is blocked with respect to a fuzzy KB.

**static is\_directly\_anywhere\_pairwise\_blocked**(  
*current\_individual*: fuzzy\_dl\_owl2.fuzzydl.individual.created\_individual.CreatedIndividual,  
*kb*: KnowledgeBase,  
) → bool

Test if the individual is anywhere pair-wise directly blocked with respect to a fuzzy KB

**static is\_directly\_anywhere\_simple\_blocked**(  
*current\_individual*: fuzzy\_dl\_owl2.fuzzydl.individual.created\_individual.CreatedIndividual,  
*kb*: KnowledgeBase,  
) → bool

**Gets if the individual is directly anywhere simple blocked with respect to a fuzzy KB.**

Case SUBSET or SET blocking. It is assumed that the individual and all ancestors are not blocked.

**static is\_directly\_blocked**(  
*current\_individual*: fuzzy\_dl\_owl2.fuzzydl.individual.created\_individual.CreatedIndividual,  
*kb*: KnowledgeBase,  
) → bool

Gets if the individual is directly blocked with respect to a fuzzy KB. A node v is directly blocked iff none of its ancestors are blocked and there exists an ancestor w such that  $L(v) = L(w)$ , where  $L(*)$  is the set of Concept's labels for a node. In this case we say that w directly blocks v.

```
static is_directly_pairwise_blocked(  
    current_individual: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,  
    kb: KnowledgeBase,  
    ) → bool  
    Test if the individual is pair-wise directly blocked with respect to a fuzzy KB.
```

```
static is_directly_simple_blocked(  
    current_individual: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,  
    kb: KnowledgeBase,  
    ) → bool  
    Gets if the individual is directly blocked with respect to a fuzzy KB.  
    Case SUBSET or SET blocking It is assumed that the individual and all ancestors are not blocked
```

```
static is_indirectly_anywhere_pairwise_blocked(  
    current_individual: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,  
    kb: KnowledgeBase,  
    ) → bool  
    Gets if the individual is indirectly anywhere pairwise blocked with respect to a fuzzy KB.
```

```
static is_indirectly_anywhere_simple_blocked(  
    current_individual: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,  
    kb: KnowledgeBase,  
    ) → bool  
    Gets if the individual is indirectly anywhere blocked with respect to a fuzzy KB. Case SUBSET or SET  
    blocking.
```

```
static is_indirectly_blocked(  
    current_individual: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,  
    kb: KnowledgeBase,  
    ) → bool  
    Gets if the individual is indirectly blocked with respect to a fuzzy KB. A node v is indirectly blocked  
    iff one of its ancestors are blocked.
```

```
static is_indirectly_pairwise_blocked(  
    current_individual: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,  
    kb: KnowledgeBase,  
    ) → bool  
    Gets if the individual is indirectly blocked with respect to a fuzzy KB.
```

```
static is_indirectly_simple_blocked(  
    current_individual: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,  
    kb: KnowledgeBase,  
    ) → bool  
    Gets if the individual is indirectly blocked with respect to a fuzzy KB. Case SUBSET or SET blocking.  
    A node v is indirectly blocked iff one of its ancestors are blocked.
```

```
static mark_indirectly_simple_unchecked(  
    current_individual: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,  
    kb: KnowledgeBase,  
    ) → None  
    Marks the subtree of a node as indirectly unblocked
```

```
static match_concept_labels(  
    current_individual: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,  
    b: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,  
    kb: KnowledgeBase,  
    ) → bool  
    Checks if two individuals match concept labels
```

```

static match_set_concept_labels(
    current_individual: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,
    b: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,
) → bool
    Check that two concept labels are equal

static match_subset_concept_labels(
    current_individual: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,
    b: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,
) → bool
    Check that every concept in the labels of this is also in b

static matching_individual(
    current_individual: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,
    kb: KnowledgeBase,
) → set[fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual]
    Checks if there is a matching individual to this one

static unblock(current_individual:
    fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual, kb:
    KnowledgeBase) → None

static unblock_directly_blocked(
    current_individual: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,
    kb: KnowledgeBase,
) → None
    Unblocks an directly blocked individual.

static unblock_indirectly_blocked(
    current_individual: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,
    kb: KnowledgeBase,
) → None
    Unblocks an indirectly blocked individual.

static unblock_pairwise(
    current_individual: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,
    kb: KnowledgeBase,
) → None
    Unblock the individual

static update_role_successors(name: str, role_name: str, kb: KnowledgeBase) → None

```

#### class DatatypeReasoner

```

static apply_at_least_value_rule(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion, kb:
    KnowledgeBase) → None

static apply_at_most_value_rule(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion, kb:
    KnowledgeBase) → None

static apply_exact_value_rule(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion, kb:
    KnowledgeBase) → None

static apply_not_at_least_value_rule(
    b: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,
    ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion,
    kb: KnowledgeBase,
) → None

static apply_not_at_most_value_rule(
    b: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,

```

```
    ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion,
    kb: KnowledgeBase,
) → None

static apply_not_exact_value_rule(
    b: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,
    ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion,
    kb: KnowledgeBase,
) → None

static apply_not_rule(
    b: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,
    ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion,
    kb: KnowledgeBase,
    type: fuzzy_dl_owl2.fuzzydl.util.constants.InequalityType,
) → None

static apply_rule(
    ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion,
    kb: KnowledgeBase,
    type: fuzzy_dl_owl2.fuzzydl.util.constants.InequalityType,
) → None

static geq_equation(
    y: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x1: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x2: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    milp: fuzzy_dl_owl2.fuzzydl.milp.milp_helper.MILPHelper,
) → None

static get_bounds(t: fuzzy_dl_owl2.fuzzydl.concrete_feature.ConcreteFeature) → list[float] | None

static get_created_individual_and_variables(
    ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,
    role: str,
    t: fuzzy_dl_owl2.fuzzydl.concrete_feature.ConcreteFeature,
    k: list[float],
    kb: KnowledgeBase,
) → list[Any]

static get_feature(f_name: str, kb: KnowledgeBase) →
    fuzzy_dl_owl2.fuzzydl.concrete_feature.ConcreteFeature

static get_xb(
    b: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,
    t: fuzzy_dl_owl2.fuzzydl.concrete_feature.ConcreteFeature,
    kb: KnowledgeBase,
) → fuzzy_dl_owl2.fuzzydl.milp.variable.Variable

static rule_feature_function(
    ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,
    t: fuzzy_dl_owl2.fuzzydl.concrete_feature.ConcreteFeature,
    fun: fuzzy_dl_owl2.fuzzydl.feature_function.FeatureFunction,
    kb: KnowledgeBase,
    x_b: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x_is_c: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x_f: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    k: list[float],
    type: fuzzy_dl_owl2.fuzzydl.util.constants.InequalityType,
) → None
```

```

static rule_not_simple_restriction(
    n: Any,
    kb: KnowledgeBase,
    x_b: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x_f: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x_is_c: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    k: list[float],
    type: fuzzy_dl_owl2.fuzzydl.util.constants.InequalityType,
) → None

static rule_not_triangular_fuzzy_number(
    b: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,
    kb: KnowledgeBase,
    f_name: str,
    x_b: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x_f: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x_is_c: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    n: fuzzy_dl_owl2.fuzzydl.concept.concrete.fuzzy_number.triangular_fuzzy_number.TriangularFuzzyNumber,
    k: list[float],
    type: fuzzy_dl_owl2.fuzzydl.util.constants.InequalityType,
) → None

static rule_simple_restriction(
    n: Any,
    kb: KnowledgeBase,
    x_b: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x_is_c: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x_f: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    k: list[float],
    type: fuzzy_dl_owl2.fuzzydl.util.constants.InequalityType,
) → None

static rule_triangular_fuzzy_number(
    b: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,
    kb: KnowledgeBase,
    f_name: str,
    x_b: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x_f: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x_is_c: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    n: fuzzy_dl_owl2.fuzzydl.concept.concrete.fuzzy_number.triangular_fuzzy_number.TriangularFuzzyNumber,
    type: fuzzy_dl_owl2.fuzzydl.util.constants.InequalityType,
) → None

static write_feature_equation(
    ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,
    fun: fuzzy_dl_owl2.fuzzydl.feature_function.FeatureFunction,
    x_b: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x_is_c: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x_f: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    k: list[float],
    type: fuzzy_dl_owl2.fuzzydl.util.constants.InequalityType,
    kb: KnowledgeBase,
)

static write_fuzzy_number_equation(
    x_f: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x_b: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x_b_prime: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,

```

```
    type: fuzzy_dl_owl2.fuzzydl.util.constants.InequalityType,
    kb: KnowledgeBase,
)

static write_not_feature_equation(
    deg: fuzzy_dl_owl2.fuzzydl.degree.degree_expression.DegreeExpression,
    x_b: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x_f: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x_is_c: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    k: list[float],
    type: fuzzy_dl_owl2.fuzzydl.util.constants.InequalityType,
    kb: KnowledgeBase,
) → None

static write_not_fuzzy_number_equation(
    x_b: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x_b_prime: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x_b_prime_is_f: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x_f: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x_is_c: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x_is_f: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    k: list[float],
    type: fuzzy_dl_owl2.fuzzydl.util.constants.InequalityType,
    kb: KnowledgeBase,
) → None
```

#### **class IndividualHandler**

```
static add_not_self_restriction(ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,
                               role: str, kb: KnowledgeBase) → None

static add_relation(
    ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,
    role_name: str,
    b: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,
    degree: fuzzy_dl_owl2.fuzzydl.degree.degree.Degree,
    kb: KnowledgeBase,
) → fuzzy_dl_owl2.fuzzydl.relation.Relation | None
    Adds b relation to the individual.

static add_restriction(
    ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,
    role_name: str,
    c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    degree: fuzzy_dl_owl2.fuzzydl.degree.degree.Degree,
    kb: KnowledgeBase,
) → None

static add_restriction(
    ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,
    role_name: str,
    ind_name: str,
    degree: fuzzy_dl_owl2.fuzzydl.degree.degree.Degree,
    kb: KnowledgeBase,
) → None

static common_part_add_restriction(
    ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,
    role_name: str,
    restrict: fuzzy_dl_owl2.fuzzydl.restriction.restriction.Restriction,
    kb: KnowledgeBase,
) → None
```



```
static solve_not_self_rule(ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,  
                           role_name: str, kb: KnowledgeBase) → None
```

Apply not self rule.

```
static solve_relation_restriction(  
    rel: fuzzy_dl_owl2.fuzzydl.relation.Relation,  
    restrict: fuzzy_dl_owl2.fuzzydl.restriction.restriction.Restriction,  
    kb: KnowledgeBase,  
) → None
```

Apply b universal restriction to b relation of the individual.

```
static unblock_simple(ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual, kb:  
                      KnowledgeBase) → None
```

**Unblock the individual.**

Case subset/set blocking

**class KnowledgeBase**

```
add_assertion(new_ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) → None
```

```
add_assertion(  
    a: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,  
    c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,  
    n: fuzzy_dl_owl2.fuzzydl.degree.degree.Degree,  
) → None
```

```
add_assertion(  
    a: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,  
    restrict: fuzzy_dl_owl2.fuzzydl.restriction.restriction.Restriction,  
) → None
```

Adds a fuzzy assertion.

```
add_assertions(list_of_assertions: list[fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion]) → None
```

Adds a list of fuzzy assertions.

```
add_atomic_concepts_disjoint(disjoint_concepts: list[str]) → None
```

Adds some disjoint concept axioms.

**Parameters**

**disjoint\_concepts** (*list[str]*) – A vector of concept names.

```
add_axiom_to_A_equiv_C(a: str, conc: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → None
```

```
add_axiom_to_A_is_a_C(  
    a: str,  
    pcd: fuzzy_dl_owl2.fuzzydl.primitive_concept_definition.PrimitiveConceptDefinition,  
    pcd_dict: dict[str, set[fuzzy_dl_owl2.fuzzydl.primitive_concept_defini-  
                           tion.PrimitiveConceptDefinition]],  
) → None
```

```
add_axiom_to_C_is_a_A(  
    conc1: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,  
    conc2: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,  
    degree: fuzzy_dl_owl2.fuzzydl.degree.degree.Degree,  
    logic_type: fuzzy_dl_owl2.fuzzydl.util.constants.LogicOperatorType,  
) → None
```

Adds a GCI (*conc2*, *conc1*, *degree*, *type*) to **add\_axiom\_to\_C\_is\_a\_A**.

**Parameters**

- **conc1** (*Concept*) – Subsumer concept.

- **conc2** (*Concept*) – Subsumed concept.
- **degree** (*Degree*) – Lower bound for the degree.
- **logic\_type** (*LogicOperatorType*) – Type of the GCI (semantics according to the implication).

```
add_axiom_to_C_is_a_D(  
    conc1: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,  
    conc2: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,  
    degree: fuzzy_dl_owl2.fuzzydl.degree.degree.Degree,  
    logic_type: fuzzy_dl_owl2.fuzzydl.util.constants.LogicOperatorType,  
) → None  
Adds a GCI (conc2, conc1, degree, type) to axioms_C_is_a_D.
```

#### Parameters

- **conc1** (*Concept*) – Subsumer concept.
- **conc2** (*Concept*) – Subsumed concept.
- **degree** (*Degree*) – Lower bound for the degree.
- **logic\_type** (*LogicOperatorType*) – Type of the GCI (semantics according to the implication).

```
add_axiom_to_C_is_a_X(  
    conc1: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,  
    conc2: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,  
    degree: fuzzy_dl_owl2.fuzzydl.degree.degree.Degree,  
    logic_type: fuzzy_dl_owl2.fuzzydl.util.constants.LogicOperatorType,  
    atomic: bool,  
) → None  
Adds a GCI (conc2, conc1, degree, type) to axioms_C_is_a_A or axioms_C_is_a_D.
```

#### Parameters

- **conc1** (*Concept*) – Subsumer concept.
- **conc2** (*Concept*) – Subsumed concept.
- **degree** (*Degree*) – Lower bound for the degree.
- **logic\_type** (*LogicOperatorType*) – Type of the GCI (semantics according to the implication).
- **atomic** (*bool*) – true for C isA A; false for C isA D

```
add_axiom_to_do_A_is_a_X(a: str, pcd: fuzzy_dl_owl2.fuzzydl.primitive_concept_ -  
                           definition.PrimitiveConceptDefinition) → None
```

```
add_axiom_to_inc(a: str, pcd:  
    fuzzy_dl_owl2.fuzzydl.primitive_concept_definition.PrimitiveConceptDefinition)  
→ None
```

```
add_axioms_to_tg() → None
```

```
add_concept(concept_name: str, conc:  
    fuzzy_dl_owl2.fuzzydl.concept.concrete.fuzzy_concrete_concept.FuzzyConcreteConcept)  
→ None
```

Adds a fuzzy concept to the array of concepts in the fuzzy KB.

```
add_concepts_disjoint(disjoint_concepts: list[str]) → None
```

```
add_concepts_disjoint(c1: str, c2: str) → None
```

```

add_concepts_disjoint(
    c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    d: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
) → None
    Adds some disjoint concept axioms.

add_created_individual(ind_name: str, ind:
    fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual) →
    None
    Adds a created individual to the KB.

add_datatype_restriction(
    restriction_type: fuzzy_dl_owl2.fuzzydl.util.constants.RestrictionType,
    o: Any,
    f_name: str,
) → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
    Adds a datatype restriction of the form (restriction_type, f_name, o).

    Parameters
        • restriction_type (RestrictionType) – Type of the datatype restriction.
        • o (Any) – Value of the datatype restriction.
        • f_name (str) – Concrete feature.

    Returns
        A datatype restriction.

    Return type
        Concept

add_disjoint_union_concept(disjoint_union_concepts: list[str]) → None
    Adds a disjoint union concept axiom.

    Parameters
        disjoint_union_concepts (list[str]) – A vector of concepts names.

add_equivalence_relation(role: str) → None
    Adds a fuzzy equivalence relation.

add_equivalent_concepts(equiv_concepts: list[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept]) →
    None
    Adds some equivalent concept axioms.

    Parameters
        equiv_concepts (list[Concept]) – An array list of vector of equivalent fuzzy con-
        cepts.

add_equivalent_roles(equiv_roles: list[str]) → None
    Adds some equivalent funcRole axioms.

    Parameters
        equiv_roles (list[str]) – An array list of equivalent fuzzy funcRole names.

add_fuzzy_number(
    f_name: str,
    f: fuzzy_dl_owl2.fuzzydl.concept.concrete.fuzzy_number.triangular_fuzzy_num-
        ber.TriangularFuzzyNumber,
) → None
    Adds a fuzzy number to the fuzzy KB.

add_gci(
    conc1: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    conc2: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,

```

```
    degree: fuzzy_dl_owl2.fuzzydl.degree.degree.Degree,  
    logic_type: fuzzy_dl_owl2.fuzzydl.util.constants.LogicOperatorType,  
) → None
```

Adds a General Concept Inclusion (conc2, conc1, degree, type).

#### Parameters

- **conc1** ([Concept](#)) – Subsumer concept.
- **conc2** ([Concept](#)) – Subsumed concept.
- **degree** ([Degree](#)) – Lower bound for the degree.
- **logic\_type** ([LogicOperatorType](#)) – Type of the GCI (semantics according to the implication).

```
add_individual(ind_name: str, ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual) → None
```

Adds a individual to the KB.

```
add_individual_to_concept(concept_id: int, ind:  
    fuzzy_dl_owl2.fuzzydl.individual.individual.Individual) → None
```

Add the individual a to the individual list of the concept.

```
add_inverse_roles(role: str, inv_role: str) → None
```

Adds an inverse funcRole axiom.

```
add_labels_with_nodes(node: str, ind_name: str) → None
```

```
add_modifier(mod_name: str, mod: fuzzy_dl_owl2.fuzzydl.modifier.modifier.Modifier) → None
```

Adds a fuzzy modifier to the fuzzy KB.

```
add_mutually_disjoint(c1: str, c2: str) → None
```

```
add_negated_datatype_restriction(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) →  
    None
```

```
add_negated_equations(  
    i: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,  
    c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,  
) → None
```

For some and all concepts, add  $x_{\{v:C\}} = 1 - x_{\{v:\text{not } C\}}$ .

```
add_parent_recursively(role_c: str, all_parents: dict[str, float], current_role: str, n1: float) →  
    None
```

Used in the computation of the transitive closure of the Role Inclusion Axioms.

```
add_relation(  
    ind_A: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,  
    role: str,  
    ind_B: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,  
    degree: fuzzy_dl_owl2.fuzzydl.degree.degree.Degree,  
) → fuzzy_dl_owl2.fuzzydl.relation.Relation
```

Adds a fuzzy relation of the form (ind\_A, ind\_B, role, degree)

#### Parameters

- **ind\_A** ([Individual](#)) – A subject individual.
- **role** ([str](#)) – An abstract role.
- **ind\_B** ([Individual](#)) – An object individual.
- **degree** ([Degree](#)) – Lower bound for the degree.

#### Returns

Added relation.

**Return type***Relation***add\_relation\_with\_role\_parent(**

*ind*: `fuzzy_dl_owl2.fuzzydl.individual.individual.Individual`,  
*role\_c*: *str*,  
*role\_p*: *str*,  
*n*: *float*,

) → None

**add\_relation\_with\_role\_parent\_in\_lukasiewicz**(*r*: `fuzzy_dl_owl2.fuzzydl.relation.Relation`,  
*role\_p*: *str*, *n*: *float*) → None

**add\_similarity\_relation**(*role*: *str*) → None

Adds a fuzzy similarity relation.

**add\_simple\_inverse\_roles**(*role*: *str*, *inv\_role*: *str*) → None

States that two roles are inverse without recursion.

**add\_subsumption(**

*conc2*: `fuzzy_dl_owl2.fuzzydl.concept.concept.Concept`,  
*conc1*: `fuzzy_dl_owl2.fuzzydl.concept.concept.Concept`,  
*degree*: `fuzzy_dl_owl2.fuzzydl.degree.degree.Degree`,  
*logic\_type*: `fuzzy_dl_owl2.fuzzydl.util.constants.LogicOperatorType`,

) → None

Adds a General Concept Inclusion (*conc2*, *conc1*, *degree*, *type*) even if the left side is atomic.

**Parameters**

- **conc1** (`Concept`) – Subsumed concept.
- **conc2** (`Concept`) – Subsumer concept.
- **degree** (`Degree`) – Lower bound for the degree.
- **logic\_type** (`LogicOperatorType`) – Type of the GCI (semantics according to the implication).

**add\_tdef\_links**(*g*: `networkx.DiGraph`, *A\_t\_C*: `dict[str, int]`, *use\_tdr*: *bool*) → bool

We return true if we know that there are cycles because of *t\_synonyms*. False does not mean that there are no cycles!

**add\_tdr\_links**(*g*: `networkx.DiGraph`, *A\_t\_C*: `dict[str, int]`, *used\_roles*: `set[str]`, *v*: *int*) → bool

We return true if we know that there are cycles because of *t\_synonyms*. False does not mean that there are no cycles!

**add\_tinc\_links**(*g*: `networkx.DiGraph`, *A\_t\_C*: `dict[str, int]`, *use\_tdr*: *bool*) → bool

We return true if we know that there are cycles because of *t\_synonyms*. False does not mean that there are no cycles!

**add\_tmp\_feature**(*feature*: *str*) → None

Add a feature from the DL parser.

**check\_fuzzy\_number\_concept\_exists**(*conc\_name*: *str*) → bool

Checks if there exists a fuzzy number with the indicated name.

**check\_individual\_exists**(*ind\_name*: *str*) → bool

Checks if there exists an individual with the given name.

**check\_role**(*role\_name*: *str*, *conc*: `fuzzy_dl_owl2.fuzzydl.concept.concept.Concept`) → None

Checks the disjointness between abstract and concrete roles.

**Parameters**

- **role\_name** (*str*) – A role name.

- **conc** ([Concept](#)) – A concept appearing in a restrictions involving the role.

**check\_trans\_role\_applied**(

*rel*: [fuzzy\\_dl\\_owl2.fuzzydl.relation.Relation](#),

*restrict*: [fuzzy\\_dl\\_owl2.fuzzydl.restriction.restriction.Restriction](#),

) → bool

Checks if transitivity has been applied to a universal restriction.

**Parameters**

- **rel** ([Relation](#)) – A relation.
- **restrict** ([Restriction](#)) – A restriction.

**Returns**

true if the transitivity rule has been applied; false otherwise.

**Return type**

bool

**classify**() → None

**clone**() → Self

Gets a copy of a knowledge base.

**clone\_without\_abox**() → Self

Gets a copy of a knowledge base except the ABox.

**compute\_blocking\_type**() → None

Computes the type of the blocking in {NO\_BLOCKING, SUBSET\_BLOCKING, SET\_BLOCKING, (ANYWHERE) DOUBLE\_BLOCKING}. If the type is in {SUBSET\_BLOCKING, SET\_BLOCKING, (ANYWHERE) DOUBLE\_BLOCKING}, it also computes whether it is dynamic or not.

**compute\_language**() → None

Computes the language of the fuzzy KB, from ALC to SHIF(D).

**compute\_variables\_old\_calculus**(*fcc*: [fuzzy\\_dl\\_owl2.fuzzydl.concept.concrete.fuzzy\\_concrete\\_concept.FuzzyConcreteConcept](#)) → None

**concept\_absorption**(*pcd*:

[fuzzy\\_dl\\_owl2.fuzzydl.primitive\\_concept\\_definition.PrimitiveConceptDefinition](#),  
*atomic*: bool) → bool

**concept\_absorption**(*tau*: [fuzzy\\_dl\\_owl2.fuzzydl.general\\_concept\\_inclusion.GeneralConceptInclusion](#),  
*atomic*: bool) → bool

**concept\_exists**(*name*: str) → bool

Checks if there exists a concept with the given name.

**convert\_strings\_into\_integers**() → None

Transforms string datatype restrictions into integer datatype restrictions.

**create\_roles\_with\_all\_parents**() → None

Computes transitive closure of the Role Inclusion Axioms.

**create\_roles\_with\_trans\_children**() → None

Used in the computation of the transitive closure of the Role Inclusion Axioms.

**define\_atomic\_concept**(

*concept\_name*: str,

*conc*: [fuzzy\\_dl\\_owl2.fuzzydl.concept.concept.Concept](#),

*implication*: [fuzzy\\_dl\\_owl2.fuzzydl.util.constants.LogicOperatorType](#),

*n*: float,

) → None

Adds an atomic fuzzy concept definition.

**define\_boolean\_concrete\_feature**(*fun\_role*: str) → None

Define a concrete feature with range boolean.

**Parameters**

**fun\_role** (str) – Name of the concrete feature.

**define\_concept**(*concept\_name*: str, *conc*: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept) → None

Adds a fuzzy concept definition.

**define\_concreate\_feature**(*role*: str) → None

**define\_equivalent\_concepts**(  
*c1*: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept,  
*c2*: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept,  
) → None

Adds a concept equivalence axiom.

**define\_integer\_concrete\_feature**(*fun\_role*: str, *d1*: int, *d2*: int) → None

Define a concrete feature with range integers in [d1, d2].

**Parameters**

- **fun\_role** (str) – Name of the concrete feature.
- **d1** (int) – Lower bound of the range.
- **d2** (int) – Upper bound of the range.

**define\_real\_concrete\_feature**(*fun\_role*: str, *d1*: float, *d2*: float) → None

Define a concrete feature with range real numbers in [d1, d2].

**Parameters**

- **fun\_role** (str) – Name of the concrete feature.
- **d1** (int) – Lower bound of the range.
- **d2** (int) – Upper bound of the range.

**define\_string\_concrete\_feature**(*fun\_role*: str) → None

Define a concrete feature with range string.

**Parameters**

**fun\_role** (str) – Name of the concrete feature.

**define\_synonym**(*concept\_name\_1*: str, *concept\_name\_2*: str) → None

Adds a fuzzy synonym definition.

**define\_synonyms**(*concept\_name\_1*: str, *concept\_name\_2*: str) → None

Adds a fuzzy synonym definition.

**definition\_absorption**(*gci*:  
fuzzy\_dl\_owl2.fuzzydl.general\_concept\_inclusion.GeneralConceptInclusion)  
→ bool

**Parameters**

**gci** (GeneralConceptInclusion) – A GCI.

**Returns**

true if there are changes; false otherwise.

**Return type**

bool

**definition\_absorption\_to\_do**(*pcd*: fuzzy\_dl\_owl2.fuzzydl.primitive\_concept\_definition.PrimitiveConceptDefinition) → bool

**Parameters**

**pcd** (*PrimitiveConceptDefinition*) – A primitive concept definition.

**Returns**

true if there are changes; false otherwise.

**Return type**

bool

**degree\_if\_not\_one**(*deg*: fuzzy\_dl\_owl2.fuzzydl.degree.degree.Degree) → str

**degree\_if\_not\_one**(*d*: float) → str

Return a string representation of the degree if it is different to 1.0.

**disjoint\_with\_defined\_concept**(*a*: str) → bool

Computes if there is some disjoint(a, b) in tDis with b being a head of an axiom in Tdef

**exists\_primate\_concept\_definition**(

*pcds*: set[fuzzy\_dl\_owl2.fuzzydl.primitive\_concept\_definition.PrimitiveConceptDefinition/,

*pcd*: fuzzy\_dl\_owl2.fuzzydl.primitive\_concept\_definition.PrimitiveConceptDefinition,

) → bool

**exit\_condition**() → None

Add every GCI to tG using the form *top* isA (C → D).

**exit\_condition\_A\_is\_a\_X**(*pcd*: fuzzy\_dl\_owl2.fuzzydl.primitive\_concept\_definition.PrimitiveConceptDefinition) → None

**exit\_condition\_C\_is\_a\_X**(*gci*: fuzzy\_dl\_owl2.fuzzydl.general\_concept\_inclusion.GeneralConceptInclusion) → None

**form\_inv\_role\_inc\_axioms**() → None

Computes relations for the inverse roles and Role Inclusion Axioms (R ⇒ P, n) implies (inv(R) ⇒ inv(P), n)

**form\_inv\_role\_relations**() → None

Computes relations for the inverse roles

**form\_inv\_trans\_roles**() → None

Computes relations for the inverse roles and transitive roles.

**gci\_transform\_define\_atomic\_concept**(

*concept\_name*: str,

*conc*: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept,

*implication*: fuzzy\_dl\_owl2.fuzzydl.util.constants.LogicOperatorType,

*n*: float,

)

**gci\_transformation**(*tau*: fuzzy\_dl\_owl2.fuzzydl.general\_concept\_inclusion.GeneralConceptInclusion, *atomic*: bool) → bool

**gci\_transformation**(*pcd*: fuzzy\_dl\_owl2.fuzzydl.primitive\_concept\_definition.PrimitiveConceptDefinition) → bool

**gci\_transformation\_add\_axiom\_to\_C\_is\_a\_X**(

*conc1*: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept,

*conc2*: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept,

*degree*: fuzzy\_dl\_owl2.fuzzydl.degree.degree.Degree,

*logic\_type*: fuzzy\_dl\_owl2.fuzzydl.util.constants.LogicOperatorType,

) → None



**gci\_transformations\_A\_is\_a\_C()** → None  
**gci\_transformations\_C\_is\_a\_A()** → None  
**gci\_transformations\_C\_is\_a\_D()** → None  
**get\_A\_t\_C()** → dict[str, int]  
**get\_classification\_node()** → *fuzzy\_dl\_owl2.fuzzydl.classification\_node.ClassificationNode* | None  
**get\_concept(name: str)** → *fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*  
 Gets a concept with indicated name.  
**get\_concept\_from\_number(n: int)** → str | None  
 Gets the concept name encoded by a number.  
**get\_correct\_version\_of\_individual(ass: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion)** → None  
**get\_correct\_version\_of\_individual(rel: fuzzy\_dl\_owl2.fuzzydl.relation.Relation)** → None  
 Use right version of the individual (needed when we clone the KB or merge individuals)  
**get\_inclusion\_degree(subsumed: str, subsumer: str)** → float  
 Computes the inclusion degree between two roles.  
**Parameters**  

- **subsumed** (str) – Subsumed funcRole.
- **subsumer** (str) – Subsumer funcRole.

**Returns**  
 Inclusion degree of subsumed in subsumer.  
**Return type**  
 float  
**get\_individual(ind\_name: str,**  
**)** → *fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual* | *fuzzy\_dl\_owl2.fuzzydl.individual.created\_individual.CreatedIndividual*  
 Gets an individual with the indicated name (creating it if necessary).  
**get\_individuals()** → dict[str, *fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual*]  
 Gets all individuals of the KB.  
**get\_inverses\_of\_inverse\_role(role: str)** → set[str] | None  
 Gets the set of inverse roles of some inverse of a given role.  
**get\_language()** → str  
 Gets the language of the fuzzy KB, from ALC to SHIF(D).  
**get\_logic()** → *fuzzy\_dl\_owl2.fuzzydl.util.constants.FuzzyLogic*  
 Gets the fuzzy logic of the fuzzy knowledge base.  
**get\_named\_individuals()** → list[*fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual*]  
**get\_new\_atomic\_concept()** → *fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*  
**get\_new\_concrete\_individual(parent: fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual,**  
**f\_name: str,**  
**)** → *fuzzy\_dl\_owl2.fuzzydl.individual.created\_individual.CreatedIndividual*  
**get\_new\_individual()** → *fuzzy\_dl\_owl2.fuzzydl.individual.created\_individual.CreatedIndividual*

**get\_new\_individual**(  
    *parent*: fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual,  
    *f\_name*: str,  
) → fuzzy\_dl\_owl2.fuzzydl.individual.created\_individual.CreatedIndividual

**get\_new\_individual\_common\_code**(  
    *parent*: fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual,  
    *f\_name*: str,  
) → fuzzy\_dl\_owl2.fuzzydl.individual.created\_individual.CreatedIndividual

**get\_number\_from\_concept**(*concept\_name*: str) → int  
    Gets a number to encode a concept name.

**get\_number\_of\_domain\_restrictions**() → int

**get\_number\_of\_range\_restrictions**() → int

**get\_subsumption\_flags**(*b*: fuzzy\_dl\_owl2.fuzzydl.classification\_node.ClassificationNode) → float  
    Retrieves the value subFlags(a, b)

**get\_tmp\_feature**(*feature*: str) → str  
    Gets a feature from the DL parser.

**get\_truth\_constants**(*s*: str) → float | None  
    Gets a truth constant from the DL parser.

**goedel\_implies**(  
    *conc1*: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept,  
    *conc2*: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept,  
    *degree*: fuzzy\_dl\_owl2.fuzzydl.degree.degree.Degree,  
) → None  
    Adds a Goedel General Concept Inclusion.

**Parameters**

- **conc1** ([Concept](#)) – Subsumed concept.
- **conc2** ([Concept](#)) – Subsumer concept.
- **degree** ([Degree](#)) – Lower bound for the degree.

**has\_functional\_abstract\_roles**() → bool

**has\_nominals\_in\_abox**() → bool  
    Checks if the ABox contains the b-some constructor.

**Returns**  
        true if the ABox contains the b-some constructor; false otherwise.

**Return type**  
        bool

**has\_nominals\_in\_tbox**() → bool  
    Checks if the TBox contains the b-some constructor.

**Returns**  
        true if the TBox contains the b-some constructor; false otherwise.

**Return type**  
        bool

**has\_only\_crisp\_sub\_concepts**(*c*: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept) → bool  
    Checks if a concept *c* is only composed of crisp concepts or not.

**implies**(  
*conc1*: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept,  
*conc2*: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept,  
*degree*: fuzzy\_dl\_owl2.fuzzydl.degree.degree.Degree,  
) → None  
 Adds a General Concept Inclusion (*conc1*, *conc2*, *degree*).

**Parameters**

- **conc1** (Concept) – Subsumed concept.
- **conc2** (Concept) – Subsumer concept.
- **degree** (Degree) – Lower bound for the degree.

**is\_assertion\_processed**(*ass*: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion) → None  
 Checks if an assertion has already been processed.

**is\_atomic\_crisp\_concept**(*c*: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept) → bool  
 Checks if a concept is atomic and crisp.

**is\_classified**() → bool  
 Checks if the knowledge base has already been classified.

**is\_concrete\_type**(*c*: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept) → bool  
 Computes if the type is one of the concretes (concrete, fuzzy number, or their complements)

**is\_crisp\_concept**(*concept\_name*: str) → bool  
 Checks if a concept is crisp.

**Parameters**  
**concept\_name** (str) – Name of the concept.

**Returns**  
 true if the semantics is classical logic or if the concept is crisp, false otherwise.

**Return type**  
 bool

**is\_crisp\_role**(*role\_name*: str) → bool  
 Checks if a role is crisp.

**Parameters**  
**role\_name** (str) – Name of the role.

**Returns**  
 true if the semantics is classical logic or if the role is crisp, false otherwise.

**Return type**  
 bool

**is\_lazy\_unfoldable**() → bool  
 Checks if the fuzzy KB is already lazy unfoldable.

**is\_loaded**() → bool  
 Checks if the fuzzy KB is loaded.

**is\_redundant\_A\_is\_a\_C**(  
*concept\_name*: str,  
*conc*: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept,  
*implication*: fuzzy\_dl\_owl2.fuzzydl.util.constants.LogicOperatorType,  
*n*: float,  
) → bool

```
is_redundant_gci(  
    C: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,  
    D: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,  
    implication: fuzzy_dl_owl2.fuzzydl.util.constants.LogicOperatorType,  
    n: float,  
) → bool
```

Checks if  $C \Rightarrow D$  redundant.

#### Parameters

- **C** ([Concept](#)) – Subsumed concept.
- **D** ([Concept](#)) – Subsumer concept.
- **implication** ([LogicOperatorType](#)) – A fuzzy implication.
- **n** ([float](#)) – Degree of truth.

#### Raises

[InconsistentOntologyException](#) – If C is *top* concept and D is *bottom* concept.

```
is_tbox_acyclic() → bool
```

Check if  $t\_inclusions \cup t\_definitions$  is acyclic

```
kleene_dienes_implies(  
    conc1: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,  
    conc2: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,  
    degree: fuzzy_dl_owl2.fuzzydl.degree.degree.Degree,  
) → None
```

Adds a Kleene-Dienes General Concept Inclusion.

#### Parameters

- **conc1** ([Concept](#)) – Subsumed concept.
- **conc2** ([Concept](#)) – Subsumer concept.
- **degree** ([Degree](#)) – Lower bound for the degree.

```
lukasiewicz_implies(  
    conc1: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,  
    conc2: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,  
    degree: fuzzy_dl_owl2.fuzzydl.degree.degree.Degree,  
) → None
```

Adds a Lukasiewicz General Concept Inclusion.

#### Parameters

- **conc1** ([Concept](#)) – Subsumed concept.
- **conc2** ([Concept](#)) – Subsumer concept.
- **degree** ([Degree](#)) – Lower bound for the degree.

```
mark_process_assertion(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) → None
```

Marks assertion as processed.

```
merge(  
    a: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,  
    b: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,  
) → None
```

Merges two individuals.

#### Parameters

- **a** ([Individual](#)) – An individual. As an effect, it will contain a merged individual.
- **b** ([Individual](#)) – Another individual.

**merge\_fillers**(*ind*: fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual, *func\_role*: str) → None

If individual *ind* has two or more fillers via the functional role *funcRole*, they are merged into just one filler concept.

#### Parameters

- **ind** (Individual) – Subject individual.
- **func\_role** (str) – A functional role.

**nominal\_absorption**(

*conc1*: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept,  
*conc2*: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept,  
*degree*: fuzzy\_dl\_owl2.fuzzydl.degree.degree.Degree,

) → bool

**optimize**(*e*: fuzzy\_dl\_owl2.fuzzydl.milp.expression.Expression) →

*fuzzy\_dl\_owl2.fuzzydl.milp.solution.Solution*

It optimizes an expression.

#### Parameters

**e** (Expression) – Expression to be optimized.

#### Returns

An optimal solution of the expression.

#### Return type

*Solution*

**partition\_loop\_A\_is\_a\_B**() → None

**partition\_loop\_A\_is\_a\_C**() → None

**partition\_loop\_C\_is\_a\_A**() → None

**partition\_loop\_C\_is\_a\_D**() → None

**partition\_loop\_to\_do\_A\_is\_a\_B**() → None

**partition\_loop\_to\_do\_A\_is\_a\_C**() → None

**preprocess\_tbox**() → None

Computes if the fuzzy KB has an acyclic TBox. If not, add primitive and concept definitions as GCIs.

**print\_tbox**() → None

**read\_object\_from\_file**(*file\_path*: str) → *KnowledgeBase*

**remove\_A\_is\_a\_B**(*key*: str, *pcd*:

*fuzzy\_dl\_owl2.fuzzydl.primitive\_concept\_definition.PrimitiveConceptDefinition*) →  
 None

**remove\_A\_is\_a\_C**(*key*: str, *pcd*:

*fuzzy\_dl\_owl2.fuzzydl.primitive\_concept\_definition.PrimitiveConceptDefinition*) →  
 None

**remove\_A\_is\_a\_X**(

*key*: str,  
*pcd*: fuzzy\_dl\_owl2.fuzzydl.primitive\_concept\_definition.PrimitiveConceptDefinition,  
*pcd\_dict*: dict[str, *set*[fuzzy\_dl\_owl2.fuzzydl.primitive\_concept\_definition.PrimitiveConceptDefinition]],

) → None

```
remove_A_is_a_X(  
    key: str,  
    pcd: fuzzy_dl_owl2.fuzzydl.primitive_concept_definition.PrimitiveConceptDefinition,  
    atomic: bool,  
) → None  
  
remove_C_is_a_A(key: str, gci:  
    fuzzy_dl_owl2.fuzzydl.general_concept_inclusion.GeneralConceptInclusion) →  
    None  
  
remove_C_is_a_D(key: str, gci:  
    fuzzy_dl_owl2.fuzzydl.general_concept_inclusion.GeneralConceptInclusion) →  
    None  
  
remove_C_is_a_X(key: str, gci:  
    fuzzy_dl_owl2.fuzzydl.general_concept_inclusion.GeneralConceptInclusion, atomic:  
    bool) → None  
  
represent_tbox_with_gcis() → None  
  
restrict_range(x_b: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable, k1: float, k2: float) → None  
restrict_range(  
    x_b: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,  
    x_f: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,  
    k1: float,  
    k2: float,  
) → None  
    Restricts the range of a variable to [k1, k2].  
  
role_absorption(tau:  
    fuzzy_dl_owl2.fuzzydl.primitive_concept_definition.PrimitiveConceptDefinition) →  
    bool  
  
role_absorption(tau: fuzzy_dl_owl2.fuzzydl.general_concept_inclusion.GeneralConceptInclusion,  
    atomic: bool) → bool  
  
role_domain(role: str, conc: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → None  
    Adds a domain funcRole axiom.  
  
role_implies(subsumed: str, subsumer: str) → None  
role_implies(subsumed: str, subsumer: str, n: float) → None  
    Adds a Role Inclusion Axiom (subsumed, subsumer, degree).  
  
role_is_functional(role: str) → None  
    Adds a functional funcRole axiom.  
  
role_is_inverse_functional(role: str) → None  
    Adds an inverse functional funcRole axiom.  
  
role_is_reflexive(role: str) → None  
    Adds a reflexive funcRole axiom.  
  
role_is_symmetric(role: str) → None  
    Adds a symmetric funcRole axiom.  
  
role_is_transitive(role: str) → None  
    Adds a transitive funcRole axiom.  
  
role_range(role: str, conc: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → None  
    Adds a funcRole range axiom.
```

**role\_subsumes**(*subsumer: str, subsumed: str, n: float*) → None

Adds a Role Inclusion Axiom (subsumer, subsumed, degree).

**Parameters**

- **subsumer** (*str*) – Subsumer funcRole.
- **subsumed** (*str*) – Subsumed funcRole.
- **n** (*float*) – Lower bound for the degree.

**role\_subsumes\_bool**(*subsumer: str, subsumed: str, n: float*) → bool

**role\_subsumes\_bool**(*subsumer: str, subsumed: str, n: float, p\_list: dict[str, dict[str, float]]*) → bool

Adds a Role Inclusion Axiom (subsumer, subsumed, degree).

**rule\_all**(*ass: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion*) → None

**rule\_and**(*ass: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion*) → None

**rule\_ass\_nom**(

*a: fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual,*

*c: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept,*

*v: str,*

) → None

Applies the rule AssNom to a node v and an assertion <a : C>.

**Parameters**

- **a** (*Individual*) – Individual of an assertion.
- **c** (*Concept*) – Concept of an assertion.
- **v** (*str*) – Node that is an a-node.

**rule\_atomic**(*ass: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion*) → None

**rule\_bottom**(*ass: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion*) → None

**rule\_choquet**(*ass: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion*) → None

**rule\_complemented**(

*i: fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual,*

*c: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept,*

) → None

**rule\_complemented\_at\_least\_datatype\_restriction**(

*b: fuzzy\_dl\_owl2.fuzzydl.individual.created\_individual.CreatedIndividual,*

*ass: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion,*

) → None

**rule\_complemented\_at\_most\_datatype\_restriction**(

*b: fuzzy\_dl\_owl2.fuzzydl.individual.created\_individual.CreatedIndividual,*

*ass: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion,*

) → None

**rule\_complemented\_atomic**(*ass: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion*) → None

**rule\_complemented\_choquet**(*ass: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion*) → None

**rule\_complemented\_complex\_assertion**(*ass: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion*)  
→ None

```
rule_complemented_concrete(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) → None

rule_complemented_exact_datatype_restriction(
    b: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,
    ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion,
) → None

rule_complemented_extended_negative_threshold(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion)
    → None

rule_complemented_extended_positive_threshold(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion)
    → None

rule_complemented_fuzzy_number(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) →
    None

rule_complemented_goedel_implication(ass:
    fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) →
    None

rule_complemented_has_value(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) → None

rule_complemented_lazy_unfolding(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) →
    None

rule_complemented_modified(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) → None

rule_complemented_negative_threshold(ass:
    fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) →
    None

rule_complemented_owa(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) → None

rule_complemented_positive_threshold(ass:
    fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) →
    None

rule_complemented_quantified_owa(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) →
    None

rule_complemented_quasi_sugeno(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) →
    None

rule_complemented_self(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) → None

rule_complemented_sigma_concept(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) →
    None

rule_complemented_sugeno(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) → None

rule_complemented_weighted_concept(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion)
    → None

rule_complemented_weighted_max(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) →
    None

rule_complemented_weighted_min(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) →
    None
```



```

rule_complemented_weighted_sum(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) →
    None
rule_complemented_weighted_sum_zero(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion)
    → None
rule_complemented_zadeh_implication(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion)
    → None
rule_concrete(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) → None
rule_domain_lazy_unfolding(domain_role: str, rel: fuzzy_dl_owl2.fuzzydl.relation.Relation) →
    None
rule_extended_negative_threshold(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) →
    None
rule_extended_positive_threshold(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) →
    None
rule_fuzzy_number(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) → None
rule_goedel_and(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) → None
rule_goedel_implication(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) → None
rule_goedel_or(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) → None
rule_has_value(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) → None
rule_lazy_unfolding(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) → None
rule_loose_lower_approximation(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) →
    None
rule_loose_upper_approximation(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) →
    None
rule_lower_approximation(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) → None
rule_lukasiewicz_and(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) → None
rule_lukasiewicz_or(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) → None
rule_modified(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) → None
rule_n2() → None
rule_n3() → None
rule_negative_threshold(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) → None
rule_or(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) → None
rule_owa(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) → None
rule_positive_threshold(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) → None
rule_quantified_owa(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) → None
rule_quasi_sugeno(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) → None
rule_range_lazy_unfolding(range_role: str, rel: fuzzy_dl_owl2.fuzzydl.relation.Relation) → None

```

**rule\_self**(*ass*: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion) → None

**rule\_sigma\_concept**(*ass*: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion) → None

**rule\_some**(*ass*: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion) → None

**rule\_sugeno**(*ass*: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion) → None

**rule\_threshold\_common**(  
    *x\_a\_in\_c*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,  
    *x\_a\_in\_tc*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,  
    *y*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,  
) → None

**rule\_tight\_lower\_approximation**(*ass*: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion) → None

**rule\_tight\_upper\_approximation**(*ass*: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion) → None

**rule\_top**(*ass*: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion) → None

**rule\_upper\_approximation**(*ass*: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion) → None

**rule\_weighted\_concept**(*ass*: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion) → None

**rule\_weighted\_max**(*ass*: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion) → None

**rule\_weighted\_min**(*ass*: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion) → None

**rule\_weighted\_sum**(*ass*: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion) → None

**rule\_weighted\_sum\_zero**(*ass*: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion) → None

**rule\_zadeh\_implication**(*ass*: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion) → None

**save\_absorbed\_tbox\_to\_file**(*output*: Callable) → None

**save\_tbox\_common\_part\_to\_file**(*output*: Callable) → None

**save\_tbox\_to\_file**(*output*: Callable) → None

**save\_to\_file**(*file\_name*: str) → None  
Saves a fuzzy KB into a text file.

**set\_crisp\_concept**(*c*: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept) → None  
Defines a concept to be crisp.

**set\_crisp\_role**(*role\_name*: str) → None  
Defines a role to be crisp.

**set\_dynamic\_blocking**() → None  
Sets dynamic blocking unless the current blocking is pairwise blocking.

**set\_logic**(*logic*: fuzzy\_dl\_owl2.fuzzydl.util.constants.FuzzyLogic) → None  
Sets the fuzzy logic of the fuzzy knowledge base.

**set\_truth\_constants**(*s*: str, *w*: float) → None  
Sets a truth constant from the DL parser.

**set\_unsatisfiable\_KB**() → None

**show\_statistics**() → None

**solveabox()** → None

Solves all the fuzzy assertions.

**solve\_assertions()** → None

Solves all the fuzzy assertions.

**solve\_cardinality\_list()** → None

Solve the list of sigma-count pending tasks

**solve\_choquet\_integral\_assertion()**

*ind:* fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual,  
*c:* fuzzy\_dl\_owl2.fuzzydl.concept.choquet\_integral.ChoquetIntegral,

) → None

Solves an assertion of the form (individual, concept) with respect to a fuzzy KB.

**solve\_choquet\_integral\_complemented\_assertion()**

*ind:* fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual,  
*c:* fuzzy\_dl\_owl2.fuzzydl.concept.operator\_concept.OperatorConcept,

) → None

Solves an assertion of the form (individual, not concept) with respect to a fuzzy KB.

**solve\_concept\_assertion()**

*ind:* fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual,  
*concept:* fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept,

) → None

**solve\_concept\_complemented\_assertion()**

*ind:* fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual,  
*lower\_limit:* fuzzy\_dl\_owl2.fuzzydl.degree.degree.Degree,  
*concept:* fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept,

) → None

**solve\_concrete\_value\_assertions()** → None

Solves the datatypes restrictions.

**solve\_crisp\_concrete\_concept\_assertion()**

*ind:* fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual,  
*concept:* fuzzy\_dl\_owl2.fuzzydl.concept.concrete.crisp\_concrete\_concept.CrispConcreteConcept,

) → None

This function define the equations for the individual belonging to the crisp set.

#### Parameters

- **ind** (*Individual*) – current individual
- **Variables** –
  - $x \Rightarrow$  variable associated with the individual
  - $x' \Rightarrow$  generic variable associated with an individual belonging to this crisp concept
- **lines** (*Draw the four*) –
  - $(b, 1) - (k_2, 0) \rightarrow y_2 \leq (x - k_2) / (k_2 - b)$
  - $(a, 1) - (k_2, 0) \rightarrow y_1 \leq (x - k_2) / (k_2 - a)$
  - $(b, 1) - (k_1, 0) \rightarrow y_3 \geq (x - k_1) / (k_1 - b)$
  - $(a, 1) - (k_1, 0) \rightarrow y_2 \geq (x - k_1) / (k_1 - a)$
- **constraints** (*Along with the following*) –
  - $y_1 + y_2 + y_3 = 1$
  - $x' + y_1 + y_3 \leq 1$

$$-x' - y_2 \geq 0$$

**solve\_domain\_and\_range\_axioms()** → None

Solves all the domain and range restrictions.

**solve\_functional\_roles()** → None

Solves the functional role axioms.

**solve\_fuzzy\_concrete\_concept\_complement\_assertion**(  
  *ind*: fuzzy\_dl\_owl2.fuzzydl.individual.created\_individual.CreatedIndividual,  
  *lower\_limit*: fuzzy\_dl\_owl2.fuzzydl.degree.degree.Degree,  
  *curr\_concept*: fuzzy\_dl\_owl2.fuzzydl.concept.operator\_concept.OperatorConcept,  
) → None

Solves an assertion of the form (individual, complement of the concept, degree) with respect to a fuzzy KB.

**solve\_gci**(  
  *ind*: fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual,  
  *gci*: fuzzy\_dl\_owl2.fuzzydl.general\_concept\_inclusion.GeneralConceptInclusion,  
) → None

**solve\_gci**(*ind*: fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual) → None  
Solves a GCI for a given individual.

**solve\_goedel\_gci**(  
  *ind*: fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual,  
  *gci*: fuzzy\_dl\_owl2.fuzzydl.general\_concept\_inclusion.GeneralConceptInclusion,  
) → None

**solve\_inverse\_roles()** → None  
Solves the inverse funcRole axioms.

**solve\_kb()** → None  
Prepares the fuzzy knowledge base to reason with it.

**solve\_kleene\_dienes\_gci**(  
  *ind*: fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual,  
  *gci*: fuzzy\_dl\_owl2.fuzzydl.general\_concept\_inclusion.GeneralConceptInclusion,  
) → None

**solve\_left\_concrete\_concept\_assertion**(  
  *ind*: fuzzy\_dl\_owl2.fuzzydl.individual.created\_individual.CreatedIndividual,  
  *concept*: fuzzy\_dl\_owl2.fuzzydl.concept.concrete.left\_concrete\_concept.LeftConcreteConcept,  
) → None

**solve\_linear\_concrete\_concept\_assertion**(  
  *ind*: fuzzy\_dl\_owl2.fuzzydl.individual.created\_individual.CreatedIndividual,  
  *concept*: fuzzy\_dl\_owl2.fuzzydl.concept.concrete.linear\_concrete\_concept.LinearConcreteConcept,  
) → None

**solve\_linear\_modifier\_assertion**(  
  *ind*: fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual,  
  *con*: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept,  
  *modifier*: fuzzy\_dl\_owl2.fuzzydl.modifier.linear\_modifier.LinearModifier,  
) → None

**solve\_lukasiewicz\_gci**(  
  *ind*: fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual,  
  *gci*: fuzzy\_dl\_owl2.fuzzydl.general\_concept\_inclusion.GeneralConceptInclusion,  
) → None

```

solve_modifier_assertion(
    ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,
    concept: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    modifier: fuzzy_dl_owl2.fuzzydl.modifier.modifier.Modifier,
) → None
    Solves an assertion of the form (individual, concept, lower degree) with respect to a fuzzy KB.

solve_modifier_complemented_assertion(
    ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,
    concept: fuzzy_dl_owl2.fuzzydl.concept.operator_concept.OperatorConcept,
    degree: fuzzy_dl_owl2.fuzzydl.degree.degree.Degree,
) → None
    Solves an assertion of the form (individual, negated concept, lower degree) with respect to a fuzzy KB.

solve_one_exist_assertion() → None
    Solves one existential assertion.

solve_owa_assertion(
    ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,
    c: fuzzy_dl_owl2.fuzzydl.concept.owa_concept.OwaConcept | fuzzy_dl_owl2.fuzzydl.concept.qowa_concept.QowaConcept,
) → None
    Solves an assertion of the form (individual, concept) with respect to a fuzzy KB.

solve_owa_complemented_assertion(
    ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,
    curr_concept: fuzzy_dl_owl2.fuzzydl.concept.operator_concept.OperatorConcept,
) → None
    Solves an assertion of the form (individual, not concept) with respect to a fuzzy KB.

solve_reflexive_role(role: str) → None
    Solves a reflexive funcRole axiom.

solve_reflexive_roles(ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual) → None

solve_reflexive_roles() → None
    Solves a reflexive funcRole axiom.

solve_right_concrete_concept_assertion(
    ind: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,
    concept: fuzzy_dl_owl2.fuzzydl.concept.concrete.right_concrete_concept.RightConcreteConcept,
) → None

solve_role_inclusion_axioms() → None

solve_role_inclusion_axioms(
    ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,
    r: fuzzy_dl_owl2.fuzzydl.relation.Relation,
) → None
    Solves the fuzzy funcRole inclusion axioms.

solve_sugeno_integral_assertion(
    ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,
    concept: fuzzy_dl_owl2.fuzzydl.concept.sugeno_integral.SugenoIntegral | fuzzy_dl_owl2.fuzzydl.concept.quasi_sugeno_integral.QsugenoIntegral,
) → None
    Solves an assertion of the form (individual, concept) with respect to a fuzzy KB.

solve_sugeno_integral_complemented_assertion(
    ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,
    curr_concept: fuzzy_dl_owl2.fuzzydl.concept.operator_concept.OperatorConcept,
) → None

```

Solves an assertion of the form (individual, not concept) with respect to a fuzzy KB.

```
solve_trapezoidal_concrete_concept_assertion(  
  ind: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,  
  concept: fuzzy_dl_owl2.fuzzydl.concept.concrete.trapezoidal_concrete_con-  
    cept.TrapezoidalConcreteConcept,  
  ) → None
```

```
solve_triangular_concrete_concept_assertion(  
  individual: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,  
  concept: fuzzy_dl_owl2.fuzzydl.concept.concrete.triangular_concrete_con-  
    cept.TriangularConcreteConcept,  
  ) → None
```

```
solve_triangular_modifier_assertion(  
  individual: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,  
  concept: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,  
  modifier: fuzzy_dl_owl2.fuzzydl.modifier.triangular_modifier.TriangularModifier,  
  ) → None
```

```
solve_w_max_assertion(  
  ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,  
  concept: fuzzy_dl_owl2.fuzzydl.concept.weighted_max_concept.WeightedMaxConcept,  
  ) → None
```

Solves an assertion of the form (individual, concept) with respect to a fuzzy KB.

```
solve_w_max_complemented_assertion(  
  ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,  
  curr_concept: fuzzy_dl_owl2.fuzzydl.concept.operator_concept.OperatorConcept,  
  ) → None
```

Solves an assertion of the form (individual, not concept) with respect to a fuzzy KB.

```
solve_w_min_assertion(  
  ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,  
  concept: fuzzy_dl_owl2.fuzzydl.concept.weighted_min_concept.WeightedMinConcept,  
  ) → None
```

Solves an assertion of the form (individual, concept) with respect to a fuzzy KB.

```
solve_w_min_complemented_assertion(  
  ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,  
  curr_concept: fuzzy_dl_owl2.fuzzydl.concept.operator_concept.OperatorConcept,  
  ) → None
```

Solves an assertion of the form (individual, not concept) with respect to a fuzzy KB.

```
solve_w_sum_assertion(  
  ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,  
  concept: fuzzy_dl_owl2.fuzzydl.concept.weighted_sum_concept.WeightedSumConcept,  
  ) → None
```

Solves an assertion of the form (individual, concept) with respect to a fuzzy KB.

```
solve_w_sum_complemented_assertion(  
  ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,  
  curr_concept: fuzzy_dl_owl2.fuzzydl.concept.operator_concept.OperatorConcept,  
  ) → None
```

Solves an assertion of the form (individual, not concept) with respect to a fuzzy KB.

```
solve_w_sum_zero_assertion(  
  ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,  
  concept: fuzzy_dl_owl2.fuzzydl.concept.weighted_sum_zero_concept.WeightedSumZeroConcept,  
  ) → None
```

Solves an assertion of the form (individual, concept) with respect to a fuzzy KB.

```
solve_w_sum_zero_complemented_assertion(
    ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,
    curr_concept: fuzzy_dl_owl2.fuzzydl.concept.operator_concept.OperatorConcept,
) → None
```

Solves an assertion of the form (individual, not concept) with respect to a fuzzy KB.

```
solve_zadeh_gci(
    ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,
    gci: fuzzy_dl_owl2.fuzzydl.general_concept_inclusion.GeneralConceptInclusion,
) → None
```

```
synonym_absorption_A_is_a_B(pcdl: fuzzy_dl_owl2.fuzzydl.primitive_concept_
    definition.PrimitiveConceptDefinition) → bool
```

Absorbs synonyms in axioms\_A\_is\_a\_B.

**Returns**

true if there are changes; false otherwise.

**Return type**

bool

```
synonym_absorption_to_do_A_is_a_B(pcdl: fuzzy_dl_owl2.fuzzydl.primitive_concept_
    definition.PrimitiveConceptDefinition) → bool
```

Absorbs synonyms in axioms\_to\_do\_A\_is\_a\_B. note that  $A \Rightarrow B$  is in  $t\_inclusions$ .

**Returns**

true if there are changes; false otherwise.

**Return type**

bool

```
unblock_children(ancestor: str) → None
```

Unblocks the children of the individual with the given name.

**Parameters**

**ancestor** (str) – Name of the ancestor individual.

```
unblock_individual(node_name: str) → None
```

Unblocks the individual and descendants of the individual with the given name. :param node\_name: Name of the ancestor individual. :type node\_name: str

```
write_object_to_file(file_path: str) → None
```

```
zadeh_implies(
    conc1: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    conc2: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
) → None
```

Adds a Zadeh General Concept Inclusion.

**Parameters**

- **conc1** (Concept) – Subsumed concept.
- **conc2** (Concept) – Subsumer concept.

**ABOX\_EXPANDED:** bool = False

**CLASSIFIED:** bool = False

**KB\_LOADED:** bool = False

```
KB_UNSAT: bool = False

abstract_roles: set[str]

acyclic_tbox: bool = False

applied_trans_role_rules: list[str] = []

assertions: list[fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion] = []

atomic_concepts: dict[str, fuzzy_dl_owl2.fuzzydl.concept.concept.Concept]

axioms_A_equiv_C: dict[str, set[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept]]

axioms_A_is_a_B: dict[str, set[fuzzy_dl_owl2.fuzzydl.primitive_concept_
definition.PrimitiveConceptDefinition]]

axioms_A_is_a_C: dict[str, set[fuzzy_dl_owl2.fuzzydl.primitive_concept_
definition.PrimitiveConceptDefinition]]

axioms_C_equiv_D:
list[fuzzy_dl_owl2.fuzzydl.concept_equivalence.ConceptEquivalence] = []

axioms_C_is_a_A: dict[str,
set[fuzzy_dl_owl2.fuzzydl.general_concept_inclusion.GeneralConceptInclusion]]

axioms_C_is_a_D: dict[str,
set[fuzzy_dl_owl2.fuzzydl.general_concept_inclusion.GeneralConceptInclusion]]

axioms_to_do_A_is_a_B: dict[str, set[fuzzy_dl_owl2.fuzzydl.primitive_concept_
definition.PrimitiveConceptDefinition]]

axioms_to_do_A_is_a_C: dict[str, set[fuzzy_dl_owl2.fuzzydl.primitive_concept_
definition.PrimitiveConceptDefinition]]

axioms_to_do_C_is_a_A: dict[str,
set[fuzzy_dl_owl2.fuzzydl.general_concept_inclusion.GeneralConceptInclusion]]

axioms_to_do_C_is_a_D: dict[str,
set[fuzzy_dl_owl2.fuzzydl.general_concept_inclusion.GeneralConceptInclusion]]

axioms_to_do_tmp_A_is_a_C: dict[str, set[fuzzy_dl_owl2.fuzzydl.primitive_
concept_definition.PrimitiveConceptDefinition]]

axioms_to_do_tmp_C_is_a_A: dict[str,
set[fuzzy_dl_owl2.fuzzydl.general_concept_inclusion.GeneralConceptInclusion]]

axioms_to_do_tmp_C_is_a_D: dict[str,
set[fuzzy_dl_owl2.fuzzydl.general_concept_inclusion.GeneralConceptInclusion]]

blocked_assertions: dict[str,
list[fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion]]

blocked_exist_assertions: dict[str,
list[fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion]]

blocking_dynamic: bool = False

blocking_type: fuzzy_dl_owl2.fuzzydl.util.constants.BlockingDynamicType

concept_individual_list: dict[int, sortedcontainers.SortedSet[fuzzy_dl_owl2.
fuzzydl.individual.created_individual.CreatedIndividual]]
```



```

concrete_concepts: dict[str, fuzzy_dl_owl2.fuzzydl.concept.concrete.fuzzy_
concrete_concept.FuzzyConcreteConcept]

concrete_features: dict[str,
fuzzy_dl_owl2.fuzzydl.concrete_feature.ConcreteFeature]

concrete_fuzzy_concepts: bool = False

concrete_roles: set[str]

directly_blocked_children: dict[str, list[str]]

disjoint_variables: dict[str, set[str]]

domain_restrictions: dict[str,
set[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept]]

exist_assertions: list[fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion] = []

functional_roles: set[str]

fuzzy_numbers: dict[str, fuzzy_dl_owl2.fuzzydl.concept.concrete.fuzzy_number.
triangular_fuzzy_number.TriangularFuzzyNumber]

individuals: dict[str, fuzzy_dl_owl2.fuzzydl.individual.individual.Individual]

inverse_functional_roles: set[str]

inverse_roles: dict[str, set[str]]

labels_with_nodes: dict[str, set[str]]

language: str = ''

lazy_unfondable: bool = False

max_depth: int = 1

milp: fuzzy_dl_owl2.fuzzydl.milp.milp_helper.MILPHelper

modifiers: dict[str, fuzzy_dl_owl2.fuzzydl.modifier.modifier.Modifier]

nodes_classification:
list[fuzzy_dl_owl2.fuzzydl.classification_node.ClassificationNode] = []

num_assertions: int = 0

num_defined_concepts: int = 0

num_defined_individuals: int = 0

num_relations: int = 0

number_of_concepts: dict[str, int]

number_of_roles: dict[str, int]

old_01_variables: int = 0

old_binary_variables: int = 0

order: dict[str, int]

```

```
positive_concrete_value_assertions:
list[fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion] = []

processed_assertions: set[int]

r_successors: dict[str, list[str]]

range_restrictions: dict[str,
set[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept]]

reflexive_roles: set[str]

roles_with_all_parents: dict[str, dict[str, float]]

roles_with_parents: dict[str, dict[str, float]]

roles_with_trans_children: dict[str, list[str]]

rule_acyclic_tbox: bool = False

rules_applied: dict[fuzzy_dl_owl2.fuzzydl.util.constants.KnowledgeBaseRules,
int]

show_language: bool = False

similarity_relations: set[str]

subsumption_flags: dict[str, dict[str, float]]

symmetric_roles: set[str]

t_G:
list[fuzzy_dl_owl2.fuzzydl.general_concept_inclusion.GeneralConceptInclusion] =
[]

t_definitions: dict[str, fuzzy_dl_owl2.fuzzydl.concept.concept.Concept]

t_disjoints: dict[str, set[str]]

t_inclusions: dict[str, set[fuzzy_dl_owl2.fuzzydl.primitive_concept_definition.
PrimitiveConceptDefinition]]

t_synonyms: dict[str, set[str]]

temp_relations_list: dict[str, list[fuzzy_dl_owl2.fuzzydl.relation.Relation]]

temp_string_concept_list: list[fuzzy_dl_owl2.fuzzydl.concept.concept.Concept] =
[]

temp_string_list: list[str] = []

tmp_features: list[str] = []

transitive_roles: set[str]

truth_constants: dict[str, float]

x_prime_individuals: dict[str, list[str]]

y_prime_individuals: dict[str, list[str]]

class LukasiewiczSolver
```

**static and\_**(*n1*: float, *n2*: float) → float

Gets the value *n1* and *n2*, according to Lukasiewicz t-norm

**static and\_equation**(

*x*: list[fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable],

*z*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,

*milp*: fuzzy\_dl\_owl2.fuzzydl.milp.milp\_helper.MILPHelper,

) → None

**static and\_equation**(

*z*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,

*x1*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,

*x2*: float,

*milp*: fuzzy\_dl\_owl2.fuzzydl.milp.milp\_helper.MILPHelper,

) → None

**static and\_equation**(

*z*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,

*x1*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,

*x2*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,

*milp*: fuzzy\_dl\_owl2.fuzzydl.milp.milp\_helper.MILPHelper,

) → None

**static and\_geq\_equation**(

*z*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,

*x1*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,

*x2*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,

*milp*: fuzzy\_dl\_owl2.fuzzydl.milp.milp\_helper.MILPHelper,

) → None

**static and\_geq\_equation**(

*z*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,

*x1*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,

*x2*: float,

*milp*: fuzzy\_dl\_owl2.fuzzydl.milp.milp\_helper.MILPHelper,

) → None

**static and\_leq\_equation**(

*z*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,

*x1*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,

*x2*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,

*milp*: fuzzy\_dl\_owl2.fuzzydl.milp.milp\_helper.MILPHelper,

) → None

Compute  $z \leq x1 \text{ AND } x2$

**static or\_equation**(

*x*: list[fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable],

*z*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,

*milp*: fuzzy\_dl\_owl2.fuzzydl.milp.milp\_helper.MILPHelper,

) → None

Compute  $z = x1 \text{ OR } x2 \text{ OR } \dots \text{ OR } xN$

**static solve\_all**(

*rel*: fuzzy\_dl\_owl2.fuzzydl.relation.Relation,

*restrict*: fuzzy\_dl\_owl2.fuzzydl.restriction.restriction.Restriction,

*kb*: KnowledgeBase,

) → None

Solves a universal restriction fuzzy assertion with respect to a reference fuzzy KB.

**static solve\_and**(*ass*: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion, *kb*: KnowledgeBase) → None

Solves a conjunction fuzzy assertion with respect to a reference fuzzy KB.

**static solve\_or**(*ass*: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion, *kb*: KnowledgeBase) → None

Solves a disjunction fuzzy assertion with respect to a reference fuzzy KB.

**static solve\_some**(*ass*: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion, *kb*: KnowledgeBase) → None

Solves a existential restriction fuzzy assertion with respect to a reference fuzzy KB.

**class ZadehSolver**

Solver for Zadeh fuzzy logic semantics.

**static and\_**(*n1*: float, *n2*: float) → float

Gets the value *n1* and *n2*, according to Goedel t-norm

**static and\_equation**(  
  *x*: list[fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable],  
  *z*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,  
  *milp*: fuzzy\_dl\_owl2.fuzzydl.milp.milp\_helper.MILPHelper,  
) → None

**static and\_equation**(  
  *x*: list[fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable],  
  *t*: fuzzy\_dl\_owl2.fuzzydl.milp.term.Term,  
  *milp*: fuzzy\_dl\_owl2.fuzzydl.milp.milp\_helper.MILPHelper,  
) → None

**static and\_equation**(  
  *z*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,  
  *x1*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,  
  *x2*: float,  
  *milp*: fuzzy\_dl\_owl2.fuzzydl.milp.milp\_helper.MILPHelper,  
) → None

**static and\_equation**(  
  *z*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,  
  *x1*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,  
  *x2*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,  
  *milp*: fuzzy\_dl\_owl2.fuzzydl.milp.milp\_helper.MILPHelper,  
) → None

**static and\_equation**(  
  *x1*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,  
  *x2*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,  
  *milp*: fuzzy\_dl\_owl2.fuzzydl.milp.milp\_helper.MILPHelper,  
) → None

**static and\_geq\_equation**(  
  *z*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,  
  *x1*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,  
  *x2*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,  
  *milp*: fuzzy\_dl\_owl2.fuzzydl.milp.milp\_helper.MILPHelper,  
) → None

**static and\_geq\_equation**(  
  *z*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,  
  *x1*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,  
  *x2*: float,  
  *milp*: fuzzy\_dl\_owl2.fuzzydl.milp.milp\_helper.MILPHelper,  
) → None

**static and\_leq\_equation**(  
  *z*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,  
  *x1*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,

```

    x2: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    milp: fuzzy_dl_owl2.fuzzydl.milp.milp_helper.MILPHelper,
) → None
    Compute  $z \leq x1 \text{ AND } x2$ 

static and_negated_equation(
    z: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x1: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x2: float,
    milp: fuzzy_dl_owl2.fuzzydl.milp.milp_helper.MILPHelper,
) → None
    Compute  $z = (1 - x1) \text{ AND } x2$ 

static goedel_implies_equation(
    z: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x1: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x2: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    milp: fuzzy_dl_owl2.fuzzydl.milp.milp_helper.MILPHelper,
) → None
    Compute  $z = x1 \text{ G-implies } x2$ 

static goedel_not_equation(
    y: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    z: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    milp: fuzzy_dl_owl2.fuzzydl.milp.milp_helper.MILPHelper,
) → None
    Compute  $y = \text{NOT } z$ 

static kleene_dienes_implies_equation(
    z: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x1: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x2: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    milp: fuzzy_dl_owl2.fuzzydl.milp.milp_helper.MILPHelper,
) → None
    Compute  $z \leq x1 \text{ KD-implies } x2$ 

static or_equation(
    z: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x1: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x2: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    milp: fuzzy_dl_owl2.fuzzydl.milp.milp_helper.MILPHelper,
) → None

static or_equation(
    x: list[fuzzy_dl_owl2.fuzzydl.milp.variable.Variable],
    z: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    milp: fuzzy_dl_owl2.fuzzydl.milp.milp_helper.MILPHelper,
) → None

static or_negated_equation(
    z: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x1: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,
    x2: float,
    milp: fuzzy_dl_owl2.fuzzydl.milp.milp_helper.MILPHelper,
) → None
    Compute  $z = (1 - x1) \text{ OR } x2$ 

static solve_all(
    rel: fuzzy_dl_owl2.fuzzydl.relation.Relation,
    restrict: fuzzy_dl_owl2.fuzzydl.restriction.restriction.Restriction,
    kb: KnowledgeBase,
) → None

```

Solves a universal restriction fuzzy assertion with respect to a reference fuzzy KB.

```
static solve_and(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion, kb: KnowledgeBase) →  
None
```

Solves a conjunction fuzzy assertion with respect to a reference fuzzy KB.

```
static solve_or(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion, kb: KnowledgeBase) →  
None
```

Solves a disjunction fuzzy assertion with respect to a reference fuzzy KB.

```
static solve_some(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion, kb: KnowledgeBase) →  
None
```

Solves a existential restriction fuzzy assertion with respect to a reference fuzzy KB.

```
static zadeh_implies_equation(  
    z: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,  
    x1: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,  
    x2: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,  
    milp: fuzzy_dl_owl2.fuzzydl.milp.milp_helper.MILPHelper,  
) → None
```

```
static zadeh_implies_equation(  
    z: float,  
    x1: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,  
    x2: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,  
    milp: fuzzy_dl_owl2.fuzzydl.milp.milp_helper.MILPHelper,  
) → None
```

```
static zadeh_implies_leq_equation(  
    z: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,  
    x1: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,  
    x2: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable,  
    milp: fuzzy_dl_owl2.fuzzydl.milp.milp_helper.MILPHelper,  
) → None
```

Compute  $z \leq x1$  Z-implies  $x2$ , where  $x1$  is binary

## fuzzy\_dl\_owl2.fuzzydl.label

### Classes

<i>Label</i>	Label (weighted concept used in created individuals)
--------------	--

### Module Contents

```
class Label(concept: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept, weight:  
            fuzzy_dl_owl2.fuzzydl.degree.degree.Degree)  
    Label (weighted concept used in created individuals)  
  
    __eq__(cw: Self) → bool  
  
    __ne__(cw: Self) → bool  
  
    __str__() → str  
  
    static weights_equal(w1: fuzzy_dl_owl2.fuzzydl.degree.degree.Degree, w2:  
                        fuzzy_dl_owl2.fuzzydl.degree.degree.Degree) → bool  
  
    Checks if two degrees are equal
```

**concept:** `fuzzy_dl_owl2.fuzzydl.concept.concept.Concept`

**weight:** `fuzzy_dl_owl2.fuzzydl.degree.degree.Degree`

`fuzzy_dl_owl2.fuzzydl.milp`

Submodules

`fuzzy_dl_owl2.fuzzydl.milp.expression`

Classes

<i>Expression</i>	Linear expression of the form $c + c_1 * x_1 + c_2 * x_2 + \dots + c_N * x_N$
-------------------	---

## Module Contents

**class** `Expression`(*constant*: `fuzzy_dl_owl2.fuzzydl.util.constants.NUMBER`)

**class** `Expression`(*constant*: `fuzzy_dl_owl2.fuzzydl.util.constants.NUMBER`, *\*terms*: `fuzzy_dl_owl2.fuzzydl.milp.term.Term`)

**class** `Expression`(*\*terms*: `fuzzy_dl_owl2.fuzzydl.milp.term.Term`)

**class** `Expression`(*expr*: `Self`)

**class** `Expression`(*v*: `list[fuzzy_dl_owl2.fuzzydl.milp.variable.Variable]` | `set[fuzzy_dl_owl2.fuzzydl.milp.variable.Variable]`)

Linear expression of the form  $c + c_1 * x_1 + c_2 * x_2 + \dots + c_N * x_N$

`__add__`(*value*: `int` | `float` | `Self` | `fuzzy_dl_owl2.fuzzydl.milp.term.Term`) → `Self`

`__eq__`(*value*: `Self`) → `bool`

`__hash__`() → `int`

`__mul__`(*scalar*: `fuzzy_dl_owl2.fuzzydl.util.constants.NUMBER`) → `Self`

`__ne__`(*value*: `Self`) → `bool`

`__neg__`() → `Self`

`__radd__`(*scalar*: `fuzzy_dl_owl2.fuzzydl.util.constants.NUMBER`) → `Self`

`__repr__`() → `str`

`__rmul__`(*scalar*: `fuzzy_dl_owl2.fuzzydl.util.constants.NUMBER`) → `Self`

`__rsub__`(*scalar*: `fuzzy_dl_owl2.fuzzydl.util.constants.NUMBER`) → `Self`

`__str__`() → `str`

Gets a printable name of the expression.

`__sub__`(*expr*: `int` | `float` | `Self` | `fuzzy_dl_owl2.fuzzydl.milp.term.Term`) → `Self`

`__truediv__`(*scalar*: `fuzzy_dl_owl2.fuzzydl.util.constants.NUMBER`) → `Self`

**static** `add_constant`(*expr*: `Self`, *constant*: `fuzzy_dl_owl2.fuzzydl.util.constants.NUMBER`) → `Self`

Adds a constant to an expression.

**static** `add_expressions`(*expr1*: `Self`, *expr2*: `Self`) → `Self`

Adds two expressions.

**add\_term**(*term*: fuzzy\_dl\_owl2.fuzzydl.milp.term.Term) → None

Adds a term to an expression.

**static add\_term**(*exp*: Self, *term*: fuzzy\_dl\_owl2.fuzzydl.milp.term.Term) → Self

Adds a term to an expression.

**clone**() → Self

**get\_constant**() → fuzzy\_dl\_owl2.fuzzydl.util.constants.NUMBER

**get\_constant\_term**(*var*: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable) →  
fuzzy\_dl\_owl2.fuzzydl.util.constants.NUMBER

Given a variable, gets its coefficient in the expression.

**get\_terms**() → list[fuzzy\_dl\_owl2.fuzzydl.milp.term.Term]

**increment\_constant**() → None

Increments the constant in one.

**static multiply\_constant**(*expr*: Self, *constant*: fuzzy\_dl\_owl2.fuzzydl.util.constants.NUMBER) →  
Self

Multiplies a constant and an expression.

**static negate\_expression**(*expr*: Self) → Self

Changes the sign of all the elements of an expression.

**set\_constant**(*constant*: fuzzy\_dl\_owl2.fuzzydl.util.constants.NUMBER) → None

**static subtract\_expressions**(*expr1*: Self, *expr2*: Self) → Self

Subtracts two expressions.

## fuzzy\_dl\_owl2.fuzzydl.milp.inequation

### Attributes

<i>EqualTo</i>
<i>GreaterThan</i>
<i>LessThan</i>

### Classes

<i>Inequation</i>	Inequality of the form $c + c_1 * x_1 + c_2 * x_2 + \dots$ ( $\geq$   $\leq$   $=$ ) 0.
-------------------	---

### Module Contents

**class Inequation**(*exp*: fuzzy\_dl\_owl2.fuzzydl.milp.expression.Expression, *i\_type*:  
fuzzy\_dl\_owl2.fuzzydl.util.constants.InequalityType)

Inequality of the form  $c + c_1 * x_1 + c_2 * x_2 + \dots$  ( $\geq$  |  $\leq$  |  $=$ ) 0.

**\_\_eq\_\_**(*value*: Self) → bool

**\_\_hash\_\_**() → int



```

__ne__(value: Self) → bool
__repr__() → str
__str__() → str
    Gets a printable name of the object.
clone() → Self
static equal_to(exp: fuzzy_dl_owl2.fuzzydl.milp.expression.Expression) → Self
get_constant() → float
get_string_type() → str
    Gets a string representation of the type.
get_terms() → list[fuzzy_dl_owl2.fuzzydl.milp.term.Term]
get_type() → fuzzy_dl_owl2.fuzzydl.util.constants.InequalityType
static greater_then(exp: fuzzy_dl_owl2.fuzzydl.milp.expression.Expression) → Self
is_zero() → bool
static less_than(exp: fuzzy_dl_owl2.fuzzydl.milp.expression.Expression) → Self
expr: fuzzy_dl_owl2.fuzzydl.milp.expression.Expression
type: fuzzy_dl_owl2.fuzzydl.util.constants.InequalityType

```

EqualTo

GreaterThan

LessThan

*fuzzy\_dl\_owl2.fuzzydl.milp.milp\_helper*

Classes

*MILPHelper*

MILP problem manager, storing the problem and calling an external solver.

## Module Contents

**class MILPHelper**

MILP problem manager, storing the problem and calling an external solver.

**add\_cardinality\_list**(sc: *fuzzy\_dl\_owl2.fuzzydl.concept.sigma\_count.SigmaCount*) → None

$\text{SigmaCount}(r, C, O, d)^I(w) = d^I(x\text{Sigma})$

### Parameters

**sc** (*SigmaCount*) – xSigma: Free variable taking the value  $\text{sigma}_{\{i2 \text{ in } O\}}$  r(i1, i2) otimes C(i2) i1: Name of an individual, subject of the relation. O: Set of individuals candidates to be the object of the relation. r: Role. C: Concept.

**add\_contradiction**() → None

Add a contradiction to make the fuzzy KB unsatisfiable

**add\_crisp\_concept**(concept\_name: str) → None

Defines a concept to be crisp.

**add\_crisp\_role**(*role\_name: str*) → None

Defines a role to be crisp.

**add\_equality**(*var1: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable, var2: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable*) → None

Add an equality of the form:  $var1 = var2$ .

**add\_new\_constraint**(  
    *expr: fuzzy\_dl\_owl2.fuzzydl.milp.expression.Expression,*  
    *constraint\_type: fuzzy\_dl\_owl2.fuzzydl.util.constants.InequalityType,*  
) → None

**add\_new\_constraint**(*x: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable, n: float*) → None

**add\_new\_constraint**(*ass: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion, n: float*) → None

**add\_new\_constraint**(*x: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable, d: fuzzy\_dl\_owl2.fuzzydl.degree.degree.Degree*) → None

**add\_new\_constraint**(*ass: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion*) → None

**add\_new\_constraint**(  
    *expr: fuzzy\_dl\_owl2.fuzzydl.milp.expression.Expression,*  
    *constraint\_type: fuzzy\_dl\_owl2.fuzzydl.util.constants.InequalityType,*  
    *degree: fuzzy\_dl\_owl2.fuzzydl.degree.degree.Degree,*  
) → None

**add\_new\_constraint**(  
    *expr: fuzzy\_dl\_owl2.fuzzydl.milp.expression.Expression,*  
    *constraint\_type: fuzzy\_dl\_owl2.fuzzydl.util.constants.InequalityType,*  
    *n: float,*  
) → None

**add\_string\_feature**(*role: str*) → None

Adds a string feature.

**add\_string\_value**(*value: str, int\_value: int*) → None

Relates the value of a string feature with an integer value.

#### Parameters

- **value** (*str*) – Value of a string feature.
- **int\_value** (*int*) – Corresponding integer value.

**change\_variable\_names**(*old\_name: str, new\_name: str, old\_is\_created\_individual: bool*) → None

Replaces the name of the variables including an individual name with the name of another individual name.

#### Parameters

- **old\_name** (*str*) – Old individual name.
- **new\_name** (*str*) – New individual name.
- **old\_is\_created\_individual** (*bool*) – Indicates whether the old individual is a created individual or not.

**check\_if\_replacement\_is\_needed**(  
    *v1: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,*  
    *s1: str,*  
    *v2: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable,*  
    *s2: str,*  
) → bool

**clone**() → Self

**exists\_nominal\_variable**(*i: str*) → bool

Checks if there exists a variable taking the value of an individual *i* belonging to the nominal concept {*i*}.

**exists\_variable**(

*a: fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual*,  
*b: fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual*,  
*role: str*,

) → bool

Checks if a variable taking the value of a role assertion exists.

#### Parameters

- **a** (*Individual*) – Object individual.
- **b** (*Individual*) – Subject individual.
- **role** (*str*) – A role name.

**get\_name\_for\_integer**(*i: int*) → str | None

Gets the name of the *i*-th variable.

**get\_negated\_nominal\_variable**(*i1: str, i2: str*) → *fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable*

Gets a variable taking the value of an individual *i1* not belonging to the nominal concept {*i2*}.

#### Parameters

- **i1** (*str*) – An individual that is subject of the assertion.
- **i2** (*str*) – An individual representing the nominal concept.

#### Returns

A variable taking the value of the assertion *i1*: not {*i2*}.

#### Return type

*Variable*

**get\_new\_variable**(*v\_type: fuzzy\_dl\_owl2.fuzzydl.util.constants.VariableType*) →  
*fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable*

Gets a new variable with the indicated type.

**get\_nominal\_variable**(*i1: str*) → *fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable*

**get\_nominal\_variable**(*i1: str, i2: str*) → *fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable*

**get\_number\_for\_assertion**(*ass: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion*) → int

Gets an integer codification of an assertion.

**get\_ordered\_permutation**(

*x: list[fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable]*,

) → list[*fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable*]

**get\_ordered\_permutation**(

*x: list[fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable]*,

*z: list[list[fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable]]*,

) → list[*fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable*]

**get\_variable**(*var\_name: str*) → *fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable*

**get\_variable**(

*var\_name: str*,

*v\_type: fuzzy\_dl\_owl2.fuzzydl.util.constants.VariableType*,

) → *fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable*

**get\_variable**(*ass: fuzzy\_dl\_owl2.fuzzydl.assertion.assertion.Assertion*) →

*fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable*

**get\_variable**(*rel: fuzzy\_dl\_owl2.fuzzydl.relation.Relation*) →

*fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable*

```
get_variable(  
    ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,  
    restrict: fuzzy_dl_owl2.fuzzydl.restriction.restriction.Restriction,  
) → fuzzy_dl_owl2.fuzzydl.milp.variable.Variable  
get_variable(  
    ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,  
    c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,  
) → fuzzy_dl_owl2.fuzzydl.milp.variable.Variable  
get_variable(  
    ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,  
    concept_name: str,  
) → fuzzy_dl_owl2.fuzzydl.milp.variable.Variable  
get_variable(  
    a: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,  
    b: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,  
    role: str,  
) → fuzzy_dl_owl2.fuzzydl.milp.variable.Variable  
get_variable(  
    a: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,  
    b: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,  
    role: str,  
    v_type: fuzzy_dl_owl2.fuzzydl.util.constants.VariableType,  
) → fuzzy_dl_owl2.fuzzydl.milp.variable.Variable  
get_variable(  
    a: str,  
    b: str,  
    role: str,  
    v_type: fuzzy_dl_owl2.fuzzydl.util.constants.VariableType,  
) → fuzzy_dl_owl2.fuzzydl.milp.variable.Variable  
get_variable(  
    ind: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,  
) → fuzzy_dl_owl2.fuzzydl.milp.variable.Variable  
get_variable(  
    ind: fuzzy_dl_owl2.fuzzydl.individual.created_individual.CreatedIndividual,  
    v_type: fuzzy_dl_owl2.fuzzydl.util.constants.VariableType,  
) → None  
  
has_nominal_variable(terms: list[fuzzy_dl_owl2.fuzzydl.milp.term.Term]) → bool  
    Checks if a collection of terms has a nominal variable.  
  
has_variable(name: str) → bool  
  
has_variable(ass: fuzzy_dl_owl2.fuzzydl.assertion.assertion.Assertion) → bool  
  
is_crisp_concept(concept_name: str) → bool  
    Checks if a concept is crisp or not.  
  
is_crisp_role(role_name: str) → bool  
    Checks if a role is crisp or not.  
  
is_nominal_variable(i: str) → bool  
    Checks if a variable 'i' is a nominal variable.  
  
optimize(objective: fuzzy_dl_owl2.fuzzydl.milp.expression.Expression) →  
    fuzzy_dl_owl2.fuzzydl.milp.solution.Solution | None  
    It optimizes an expression using a solvers from MILPProvider.  
  
    Parameters  
    objective (Expression) – Expression to be optimized.
```

**Raises**

**ValueError** – If MILPProvider is not known.

**Returns**

An optimal solution of the expression

**Return type**

*Optional*[*Solution*]

**print\_instance\_of\_labels**(*f\_name: str, ind\_name: str, value: float*) → None

**print\_instance\_of\_labels**(*name: str, value: float*) → None

Shows the membership degrees to some linguistic labels.

**set\_binary\_variables**() → None

Transforms every [0,1]-variable into a {0,1} variable.

**set\_nominal\_variables**(*value: bool*) → None

**solve\_gurobi**(

*objective: fuzzy\_dl\_owl2.fuzzydl.milp.expression.Expression,*

*)* → *fuzzy\_dl\_owl2.fuzzydl.milp.solution.Solution* | None

Solves a MILP problem using Gurobi.

**solve\_mip**(*objective: fuzzy\_dl\_owl2.fuzzydl.milp.expression.Expression*) →

*fuzzy\_dl\_owl2.fuzzydl.milp.solution.Solution* | None

**solve\_pulp**(

*objective: fuzzy\_dl\_owl2.fuzzydl.milp.expression.Expression,*

*)* → *fuzzy\_dl\_owl2.fuzzydl.milp.solution.Solution* | None

**PARTITION:** bool = False

**PRINT\_LABELS:** bool = True

**PRINT\_VARIABLES:** bool = True

**cardinalities:** list[*fuzzy\_dl\_owl2.fuzzydl.concept.sigma\_count.SigmaCount*] = []

**constraints:** list[*fuzzy\_dl\_owl2.fuzzydl.milp.inequation.Inequation*] = []

**crisp\_concepts:** set[str]

**crisp\_roles:** set[str]

**nominal\_variables:** bool = False

**number\_of\_variables:** dict[str, int]

**show\_vars:** *fuzzy\_dl\_owl2.fuzzydl.milp.show\_variables\_helper.ShowVariablesHelper*

**string\_features:** set[str]

**string\_values:** dict[int, str]

**variables:** list[*fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable*] = []

**fuzzy\_dl\_owl2.fuzzydl.milp.show\_variables\_helper**

**Classes**

---

*ShowVariablesHelper*

---

**Module Contents****class ShowVariablesHelper****add\_abstract\_filler\_to\_show**(*role\_name: str*) → None**add\_abstract\_filler\_to\_show**(*role\_name: str, ind\_name: str*) → None

Shows the membership degree to some atomic concepts of the fillers of an abstract role.

**add\_concept\_to\_show**(*conc\_name: str*) → None

Show membership degree of every instance of an atomic concept.

**add\_concrete\_filler\_to\_show**(*f\_name: str*) → None**add\_concrete\_filler\_to\_show**(*f\_name: str, ind\_name: str*) → None**add\_concrete\_filler\_to\_show**(*f\_name: str,**ind\_name: str,**ar: list[fuzzy\_dl\_owl2.fuzzydl.concept.concrete.fuzzy\_concrete\_concept.FuzzyConcreteConcept],*

) → None

Shows the value of the fillers of a concrete feature.

**add\_individual\_to\_show**(*ind\_name: str*) → None

Shows the value of an individual to every atomic concept.

**add\_variable**(*var: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable, name\_to\_show: str*) → None

Add a variable to shown, showing it with a given name.

**Parameters**

- **var** (*Variable*) – A variable.
- **name\_to\_show** (*str*) – Name of the variable when shown.

**clone**() → Self**get\_labels**(*var\_name: str*) →

list[fuzzy\_dl\_owl2.fuzzydl.concept.concrete.fuzzy\_concrete\_concept.FuzzyConcreteConcept]

Gets the fuzzy concrete concepts marked to be shown for a variable.

**get\_name**(*var: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable*) → str

Gets the name of a variable.

**get\_variables**() → list[fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable]

Gets the variables to be shown.

**show\_abstract\_role\_fillers**(*role\_name: str, ind\_name: str*) → bool

Returns whether a given individuals is marked for showing every filler of an abstract role.

**Parameters**

- **role\_name** (*str*) – Name of the abstract role.
- **ind\_name** (*str*) – Name of the individual.

**show\_concepts**(*concept\_name: str*) → bool

Returns whether an atomic concept is marked to show the membership degree of every individual.

**Parameters**

- **concept\_name** (*str*) – Name of atomic concept.

**Returns**

true if the concept is marked to be shown; false otherwise.

**Return type**

bool

**show\_concrete\_fillers**(*f\_name*: str, *ind\_name*: str) → bool

Returns whether a given individuals is marked for showing every filler of a concrete feature.

**Parameters**

- **f\_name** (str) – Name of the concrete feature.
- **ind\_name** (str) – Name of the individual.

**show\_individuals**(*ind\_name*: str) → bool

Checks whether an individual is marked to be shown or not.

**show\_variable**(*var*: [fuzzy\\_dl\\_owl2.fuzzydl.milp.variable.Variable](#)) → bool

Checks whether the variable exists or not.

**abstract\_fillers**: dict[str, set[str]]

**concepts**: set[str]

**concrete\_fillers**: dict[str, set[str]]

**global\_abstract\_fillers**: set[str]

**global\_concrete\_fillers**: set[str]

**individuals**: set[str]

**labels\_for\_fillers**: dict[str, list[[fuzzy\\_dl\\_owl2.fuzzydl.concept.concrete.fuzzy\\_concrete\\_concept.FuzzyConcreteConcept](#)]]

**variables**: dict[[fuzzy\\_dl\\_owl2.fuzzydl.milp.variable.Variable](#), str]

**fuzzy\_dl\_owl2.fuzzydl.milp.solution**

**Classes**

*Solution*

**Module Contents**

**class** **Solution**(*consistent*: bool)

**class** **Solution**(*sol*: float)

**\_\_hash\_\_**() → int

**\_\_repr\_\_**() → str

**\_\_str\_\_**() → str

**add\_showed\_variable**(*var\_name*: str, *value*: float) → None

Sets the value of a showed variable.

**get\_showed\_variables**() → dict[str, float]

Gets the values of some variables after solving a query over a consistent KB.

**get\_solution()** → bool | float

Gets the solution to some query over a consistent KB.

**is\_consistent\_kb()** → bool

Indicates whether the original KB is consistent or not.

**CONSISTENT\_KB:** bool = True

**INCONSISTENT\_KB:** bool = False

## fuzzy\_dl\_owl2.fuzzydl.milp.term

### Classes

---

*Term*

---

### Module Contents

**class** **Term**(*coeff: float*, *var: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable*)

**class** **Term**(*var: fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable*)

**\_\_add\_\_**(*term: Self*) → Self

**\_\_eq\_\_**(*term: Self*) → bool

**\_\_hash\_\_**() → int

**\_\_mul\_\_**(*scalar: float*) → Self

**\_\_ne\_\_**(*term: Self*) → bool

**\_\_neg\_\_**() → Self

**\_\_repr\_\_**() → str

**\_\_rmul\_\_**(*scalar: float*) → Self

**\_\_str\_\_**() → str

**\_\_sub\_\_**(*term: Self*) → Self

**\_\_truediv\_\_**(*scalar: float*) → Self

**clone**() → Self

**get\_coeff**() → float

**get\_var**() → *fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable*

## fuzzy\_dl\_owl2.fuzzydl.milp.variable

### Attributes

---

*BinaryVar*

*FreeVar*

---

continues on next page



Table 80 – continued from previous page

*IntegerVar**UpVar***Classes***Variable***Module Contents**

```

class Variable(name: str, v_type: fuzzy_dl_owl2.fuzzydl.util.constants.VariableType)

    __eq__(value: Self) → bool
    __hash__() → int
    __ne__(value: object) → bool
    __repr__() → str
    __str__() → str
    clone() → Self
    static get_binary_variable(name: str) → Self
    static get_continuous_variable(name: str) → Self
    get_datatype_filler_type() → bool
    static get_integer_variable(name: str) → Self
    get_lower_bound() → float
    static get_new_variable(v_type: fuzzy_dl_owl2.fuzzydl.util.constants.VariableType) → Self
    static get_semi_continuous_variable(name: str) → Self
    get_type() → fuzzy_dl_owl2.fuzzydl.util.constants.VariableType
    get_upper_bound() → float
    set_binary_variable() → None
    set_datatype_filler_variable() → None
    set_name(name: str) → None
    set_type(v_type: fuzzy_dl_owl2.fuzzydl.util.constants.VariableType) → None
    VARIABLE_NAME: str = 'y'
    VARIABLE_NUMBER: int = 0
    datatype_filler: bool = False
    lower_bound: float = 0.0

```

**name:** str

**type:** `fuzzy_dl_owl2.fuzzydl.util.constants.VariableType` = None

**upper\_bound:** float = 0.0

BinaryVar

FreeVar

IntegerVar

UpVar

`fuzzy_dl_owl2.fuzzydl.modifier`

Submodules

`fuzzy_dl_owl2.fuzzydl.modifier.linear_modifier`

Classes

---

*LinearModifier*

Linear modifier with parameter c

---

## Module Contents

**class** `LinearModifier`(*name: str, c: float*)

Bases: `fuzzy_dl_owl2.fuzzydl.modifier.modifier.Modifier`

Linear modifier with parameter c

**\_\_and\_\_**(*value: Self*) → Self

**\_\_hash\_\_**() → int

**\_\_neg\_\_**() → `fuzzy_dl_owl2.fuzzydl.concept.concept.Concept`

**\_\_or\_\_**(*value: Self*) → Self

**clone**() → Self

**compute\_name**() → str

**get\_membership\_degree**(*value: float*) → float

Gets the image in [0,1] of a real number to the modifier.

### Parameters

**value** (*float*) – A real number in the range of values of the modifier function.

### Returns

Image in [0,1] of x to the explicit modifier function.

### Return type

float

**modify**(

*concept: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept,*

) → `fuzzy_dl_owl2.fuzzydl.concept.modified.linearly_modified_concept.LinearlyModifiedConcept`

Modifies a fuzzy concept.

### Parameters

**concept** (*Concept*) – A fuzzy concept

**Returns**

Fuzzy concept resulting from the application of the modifier to c.

**Return type**

*Concept*

property a: float

property b: float

property c: float

**fuzzy\_dl\_owl2.fuzzydl.modifier.modifier**

**Classes**

*Modifier*

Fuzzy modifier.

**Module Contents**

**class Modifier**(*name: str*)

Bases: abc.ABC

Fuzzy modifier.

**\_\_repr\_\_**() → str

**\_\_str\_\_**() → str

**abstractmethod clone**() → Self

**abstractmethod compute\_name**() → str

**abstractmethod get\_membership\_degree**(*value: float*) → float

Gets the image in [0,1] of a real number to the modifier.

**Parameters**

**value** (*float*) – A real number in the range of values of the modifier function.

**Returns**

Image in [0,1] of x to the explicit modifier function.

**Return type**

float

**abstractmethod modify**(*concept: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*) → *fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*

Modifies a fuzzy concept.

**Parameters**

**concept** (*Concept*) – A fuzzy concept

**Returns**

Fuzzy concept resulting from the application of the modifier to c.

**Return type**

*Concept*

**set\_name**(*name: str*) → None

**name:** str

**fuzzy\_dl\_owl2.fuzzydl.modifier.triangular\_modifier****Classes**

---

*TriangularModifier*Fuzzy modifier.

---

**Module Contents****class** **TriangularModifier**(*name: str, a: float, b: float, c: float*)Bases: *fuzzy\_dl\_owl2.fuzzydl.modifier.modifier.Modifier*

Fuzzy modifier.

**\_\_and\_\_**(*value: Self*) → *Self***\_\_hash\_\_**() → *int***\_\_neg\_\_**() → *fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept***\_\_or\_\_**(*value: Self*) → *Self***clone**() → *Self***compute\_name**() → *str***get\_membership\_degree**(*x: float*) → *float*

Gets the image in [0,1] of a real number to the modifier.

**Parameters****value** (*float*) – A real number in the range of values of the modifier function.**Returns**

Image in [0,1] of x to the explicit modifier function.

**Return type***float***modify**(*concept: fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*) →*fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*

Modifies a fuzzy concept.

**Parameters****concept** (*Concept*) – A fuzzy concept**Returns**

Fuzzy concept resulting from the application of the modifier to c.

**Return type***Concept***property a:** *float***property b:** *float***property c:** *float***fuzzy\_dl\_owl2.fuzzydl.parser****Submodules****fuzzy\_dl\_owl2.fuzzydl.parser.dl\_parser**

## Attributes

*FILENAME*

*LOG\_DIR*

*TODAY*

## Classes

*DLParser*

## Module Contents

### class DLParser

Bases: object

**static** `get_grammatics()` → `pyparsing.ParserElement`

This function generate the grammatics to parse the predicate with formula “formula”.

#### Parameters

**formula** (= *The predicate formula used for the parsing.*)

#### Return type

The parsed result given by pyparsing.

**static** `get_kb(*args)` → `tuple[fuzzy_dl_owl2.fuzzydl.knowledge_base.KnowledgeBase, list[fuzzy_dl_owl2.fuzzydl.query.query.Query]]`

**static** `load_config(*args)` → `None`

**static** `main(*args)` → `None`

**static** `parse_string(instring: str)` → `pyparsing.ParseResults`

**static** `parse_string_opt(filename: str)` → `pyparsing.ParseResults`

**kb:** `fuzzy_dl_owl2.fuzzydl.knowledge_base.KnowledgeBase` = `None`

**queries\_list:** `list[fuzzy_dl_owl2.fuzzydl.query.query.Query]` = `[]`

**FILENAME:** `str`

**LOG\_DIR:** `str`

**TODAY:** `datetime.datetime`

**fuzzy\_dl\_owl2.fuzzydl.primitive\_concept\_definition**

## Classes

*PrimitiveConceptDefinition*

General concept inclusion axiom.

## Module Contents

```
class PrimitiveConceptDefinition(  
    defined: str,  
    definition: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,  
    implication: fuzzy_dl_owl2.fuzzydl.util.constants.LogicOperatorType,  
    degree: float,  
)  
    General concept inclusion axiom.  
  
    __eq__(other: Self) → bool  
    __ge__(other: Self) → bool  
    __gt__(other: Self) → bool  
    __hash__() → int  
    __le__(other: Self) → bool  
    __lt__(other: Self) → bool  
    __ne__(other: Self) → bool  
    __repr__() → str  
    __str__() → str  
    clone() → Self  
    get_defined_concept() → str  
    get_definition() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept  
    get_degree() → float  
    get_type() → fuzzy_dl_owl2.fuzzydl.util.constants.LogicOperatorType  
    set_definition(definition: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept) → None  
    set_degree(deg: float) → None  
    defined: str  
    definition: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept  
    degree: float  
    implication: fuzzy_dl_owl2.fuzzydl.util.constants.LogicOperatorType
```

**fuzzy\_dl\_owl2.fuzzydl.query**

## Submodules

**fuzzy\_dl\_owl2.fuzzydl.query.all\_instances\_query**

## Classes

---

*AllInstancesQuery*

Min instance query for every individual of a knowledge base.

---

## Module Contents

```
class AllInstancesQuery(concept: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept)
    Bases: fuzzy_dl_owl2.fuzzydl.query.query.Query
    Min instance query for every individual of a knowledge base.
    __str__() → str
        Solves the query over a fuzzy KB
    get_degrees() → list[float]
    get_individuals() → list[fuzzy_dl_owl2.fuzzydl.individual.individual.Individual]
    preprocess(kb: fuzzy_dl_owl2.fuzzydl.knowledge_base.KnowledgeBase) → None
        Performs some preprocessing steps of the query over a fuzzy KB.
    solve(kb: fuzzy_dl_owl2.fuzzydl.knowledge_base.KnowledgeBase) →
        fuzzy_dl_owl2.fuzzydl.milp.solution.Solution
        Solve the query using given knowledge base
    solve_new(kb: fuzzy_dl_owl2.fuzzydl.knowledge_base.KnowledgeBase) →
        fuzzy_dl_owl2.fuzzydl.milp.solution.Solution
        Specific algorithm to solve the instance retrieval.
    conc
    degrees: list[float] = []
    individuals: list[fuzzy_dl_owl2.fuzzydl.individual.individual.Individual] = []
    name = 'Instances of Uninferable?'
```

## fuzzy\_dl\_owl2.fuzzydl.query.bnp\_query

### Classes

<i>BnpQuery</i>	Helper class that provides a standard way to create an ABC using
-----------------	--

## Module Contents

```
class BnpQuery(c: fuzzy_dl_owl2.fuzzydl.concept.concrete.fuzzy_number.triangular_fuzzy_
    number.TriangularFuzzyNumber)
    Bases: fuzzy_dl_owl2.fuzzydl.query.query.Query
    Helper class that provides a standard way to create an ABC using inheritance.
    __str__() → str
        Solves the query over a fuzzy KB
    preprocess(kb: fuzzy_dl_owl2.fuzzydl.knowledge_base.KnowledgeBase) → None
        Performs some preprocessing steps of the query over a fuzzy KB.
    solve(kb: fuzzy_dl_owl2.fuzzydl.knowledge_base.KnowledgeBase) →
        fuzzy_dl_owl2.fuzzydl.milp.solution.Solution
        Solve the query using given knowledge base
    c: fuzzy_dl_owl2.fuzzydl.concept.concrete.fuzzy_number.triangular_fuzzy_number.
        TriangularFuzzyNumber
```

**fuzzy\_dl\_owl2.fuzzydl.query.classification\_query****Classes**

<i>ClassificationQuery</i>	Helper class that provides a standard way to create an ABC using
----------------------------	--

**Module Contents****class ClassificationQuery**

Bases: *fuzzy\_dl\_owl2.fuzzydl.query.query.Query*

Helper class that provides a standard way to create an ABC using inheritance.

**\_\_str\_\_**() → str

Solves the query over a fuzzy KB

**preprocess**(*kb*: *fuzzy\_dl\_owl2.fuzzydl.knowledge\_base.KnowledgeBase*) → None

Performs some preprocessing steps of the query over a fuzzy KB.

**solve**(*kb*: *fuzzy\_dl\_owl2.fuzzydl.knowledge\_base.KnowledgeBase*) →  
*fuzzy\_dl\_owl2.fuzzydl.milp.solution.Solution*

Solve the query using given knowledge base

**fuzzy\_dl\_owl2.fuzzydl.query.defuzzify****Submodules****fuzzy\_dl\_owl2.fuzzydl.query.defuzzify.defuzzify\_query****Classes**

<i>DefuzzifyQuery</i>	Helper class that provides a standard way to create an ABC using
-----------------------	--

**Module Contents****class DefuzzifyQuery(**

*c*: *fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*,  
*ind*: *fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual*,  
*feature\_name*: str,

)

Bases: *fuzzy\_dl\_owl2.fuzzydl.query.query.Query*

Helper class that provides a standard way to create an ABC using inheritance.

**abstractmethod get\_obj\_expression**(*variable*: *fuzzy\_dl\_owl2.fuzzydl.milp.variable.Variable*) →  
*fuzzy\_dl\_owl2.fuzzydl.milp.expression.Expression*

**preprocess**(*kb*: *fuzzy\_dl\_owl2.fuzzydl.knowledge\_base.KnowledgeBase*) → None

Performs some preprocessing steps of the query over a fuzzy KB.

**solve**(*kb*: *fuzzy\_dl\_owl2.fuzzydl.knowledge\_base.KnowledgeBase*) →  
*fuzzy\_dl\_owl2.fuzzydl.milp.solution.Solution* | None

Solve the query using given knowledge base

**a**: *fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual*



```

conc: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
f_name: str
obj_expr: fuzzy_dl_owl2.fuzzydl.milp.expression.Expression = None

```

**fuzzy\_dl\_owl2.fuzzydl.query.defuzzify.lom\_defuzzify\_query**

## Classes

<i>LomDefuzzifyQuery</i>	Largest of maxima defuzzification query
--------------------------	---

## Module Contents

```

class LomDefuzzifyQuery(
    c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,
    feature_name: str,
)
    Bases: fuzzy_dl_owl2.fuzzydl.query.defuzzify.defuzzify_query.DefuzzifyQuery
    Largest of maxima defuzzification query
    __str__() → str
        Solves the query over a fuzzy KB
    get_obj_expression(variable: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable) →
        fuzzy_dl_owl2.fuzzydl.milp.expression.Expression

```

**fuzzy\_dl\_owl2.fuzzydl.query.defuzzify.mom\_defuzzify\_query**

## Classes

<i>MomDefuzzifyQuery</i>	Middle of maxima defuzzification query.
--------------------------	---

## Module Contents

```

class MomDefuzzifyQuery(
    c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,
    feature_name: str,
)
    Bases: fuzzy_dl_owl2.fuzzydl.query.defuzzify.defuzzify_query.DefuzzifyQuery
    Middle of maxima defuzzification query.
    __str__() → str
        Solves the query over a fuzzy KB
    get_obj_expression(variable: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable) →
        fuzzy_dl_owl2.fuzzydl.milp.expression.Expression
    preprocess(kb: fuzzy_dl_owl2.fuzzydl.knowledge_base.KnowledgeBase) → None
        Performs some preprocessing steps of the query over a fuzzy KB.
    solve(kb: fuzzy_dl_owl2.fuzzydl.knowledge_base.KnowledgeBase) →
        fuzzy_dl_owl2.fuzzydl.milp.solution.Solution
        Solve the query using given knowledge base

```

## fuzzy\_dl\_owl2.fuzzydl.query.defuzzify.som\_defuzzify\_query

## Classes

<i>SomDefuzzifyQuery</i>	Smallest of maxima defuzzification query.
--------------------------	---

## Module Contents

```

class SomDefuzzifyQuery(
    c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,
    f_name: str,
)

Bases: fuzzy_dl_owl2.fuzzydl.query.defuzzify.defuzzify_query.DefuzzifyQuery
Smallest of maxima defuzzification query.

__str__() → str
    Solves the query over a fuzzy KB

get_obj_expression(q: fuzzy_dl_owl2.fuzzydl.milp.variable.Variable) →
    fuzzy_dl_owl2.fuzzydl.milp.expression.Expression

```

## fuzzy\_dl\_owl2.fuzzydl.query.instance\_query

## Classes

<i>InstanceQuery</i>	Instance checking query
----------------------	-------------------------

## Module Contents

```

class InstanceQuery(
    concept: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    individual: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,
)

Bases: fuzzy_dl_owl2.fuzzydl.query.query.Query, abc.ABC
Instance checking query

conc: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept

ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual

obj_expr: fuzzy_dl_owl2.fuzzydl.milp.expression.Expression = None

```

## fuzzy\_dl\_owl2.fuzzydl.query.kb\_satisfiable\_query

## Classes

<i>KbSatisfiableQuery</i>	Knowledge base satisfiability degree
---------------------------	--------------------------------------

## Module Contents

**class KbSatisfiableQuery**Bases: *fuzzy\_dl\_owl2.fuzzydl.query.query.Query*

Knowledge base satisfiability degree

**\_\_str\_\_**() → str

Solves the query over a fuzzy KB

**is\_consistent\_kb**(*kb*: *fuzzy\_dl\_owl2.fuzzydl.knowledge\_base.KnowledgeBase*) → bool**preprocess**(*kb*: *fuzzy\_dl\_owl2.fuzzydl.knowledge\_base.KnowledgeBase*) → None

Performs some preprocessing steps of the query over a fuzzy KB.

**solve**(*kb*: *fuzzy\_dl\_owl2.fuzzydl.knowledge\_base.KnowledgeBase*) →  
*fuzzy\_dl\_owl2.fuzzydl.milp.solution.Solution*

Solve the query using given knowledge base

**fuzzy\_dl\_owl2.fuzzydl.query.max****Submodules****fuzzy\_dl\_owl2.fuzzydl.query.max.max\_instance\_query****Classes***MaxInstanceQuery*

Lowest upper bound of a concept assertion

**Module Contents****class MaxInstanceQuery(***concept*: *fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*,*individual*: *fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual*,**)**Bases: *fuzzy\_dl\_owl2.fuzzydl.query.instance\_query.InstanceQuery*

Lowest upper bound of a concept assertion

**\_\_str\_\_**() → str

Solves the query over a fuzzy KB

**preprocess**(*kb*: *fuzzy\_dl\_owl2.fuzzydl.knowledge\_base.KnowledgeBase*) → None

Performs some preprocessing steps of the query over a fuzzy KB.

**solve**(*kb*: *fuzzy\_dl\_owl2.fuzzydl.knowledge\_base.KnowledgeBase*) →  
*fuzzy\_dl\_owl2.fuzzydl.milp.solution.Solution*

Solve the query using given knowledge base

**fuzzy\_dl\_owl2.fuzzydl.query.max.max\_query****Classes***MaxQuery*

Maximize expression query

**Module Contents**

```
class MaxQuery(expr: fuzzy_dl_owl2.fuzzydl.milp.expression.Expression)
    Bases: fuzzy_dl_owl2.fuzzydl.query.query.Query
    Maximize expression query
    __str__() → str
        Solves the query over a fuzzy KB
    preprocess(kb: fuzzy_dl_owl2.fuzzydl.knowledge_base.KnowledgeBase) → None
        Performs some preprocessing steps of the query over a fuzzy KB.
    solve(kb: fuzzy_dl_owl2.fuzzydl.knowledge_base.KnowledgeBase) →
        fuzzy_dl_owl2.fuzzydl.milp.solution.Solution
        Solve the query using given knowledge base
    obj_expr: fuzzy_dl_owl2.fuzzydl.milp.expression.Expression
```

**fuzzy\_dl\_owl2.fuzzydl.query.max.max\_related\_query**

### Classes

<i>MaxRelatedQuery</i>	Lowest upper bound of a role assertion (ind1, ind2, role)
------------------------	---

### Module Contents

```
class MaxRelatedQuery(
    a: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,
    b: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,
    role_name: str,
)
    Bases: fuzzy_dl_owl2.fuzzydl.query.related_query.RelatedQuery
    Lowest upper bound of a role assertion (ind1, ind2, role)
    __str__() → str
        Solves the query over a fuzzy KB
    preprocess(kb: fuzzy_dl_owl2.fuzzydl.knowledge_base.KnowledgeBase) → None
        Performs some preprocessing steps of the query over a fuzzy KB.
    solve(kb: fuzzy_dl_owl2.fuzzydl.knowledge_base.KnowledgeBase) →
        fuzzy_dl_owl2.fuzzydl.milp.solution.Solution
        Solve the query using given knowledge base
    ind1: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual
    ind2: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual
    role: str
```

**fuzzy\_dl\_owl2.fuzzydl.query.max.max\_satisfiable\_query**

### Classes

<i>MaxSatisfiableQuery</i>	Maximal satisfiability degree of a fuzzy concept.
----------------------------	---

## Module Contents

```
class MaxSatisfiableQuery(c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept)
class MaxSatisfiableQuery(c: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept, a:
    fuzzy_dl_owl2.fuzzydl.individual.individual.Individual)
    Bases: fuzzy_dl_owl2.fuzzydl.query.satisfiable_query.SatisfiableQuery
    Maximal satisfiability degree of a fuzzy concept.
    __str__() → str
        Solves the query over a fuzzy KB
    preprocess(kb: fuzzy_dl_owl2.fuzzydl.knowledge_base.KnowledgeBase) → None
        Performs some preprocessing steps of the query over a fuzzy KB.
    solve(kb: fuzzy_dl_owl2.fuzzydl.knowledge_base.KnowledgeBase) →
        fuzzy_dl_owl2.fuzzydl.milp.solution.Solution
        Solve the query using given knowledge base
```

**fuzzy\_dl\_owl2.fuzzydl.query.max.max\_subsumes\_query**

## Classes

<i>MaxSubsumesQuery</i>	Maximize subsumption query.
-------------------------	-----------------------------

## Module Contents

```
class MaxSubsumesQuery(
    c1: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    c2: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    type_: fuzzy_dl_owl2.fuzzydl.util.constants.LogicOperatorType,
)
    Bases: fuzzy_dl_owl2.fuzzydl.query.subsumption_query.SubsumptionQuery
    Maximize subsumption query.
    __str__() → str
        Solves the query over a fuzzy KB
    preprocess(kb: fuzzy_dl_owl2.fuzzydl.knowledge_base.KnowledgeBase) → None
        Performs some preprocessing steps of the query over a fuzzy KB.
    solve(kb: fuzzy_dl_owl2.fuzzydl.knowledge_base.KnowledgeBase) →
        fuzzy_dl_owl2.fuzzydl.milp.solution.Solution
        Solve the query using given knowledge base
```

**fuzzy\_dl\_owl2.fuzzydl.query.min**

## Submodules

**fuzzy\_dl\_owl2.fuzzydl.query.min.min\_instance\_query**

## Classes

<i>MinInstanceQuery</i>	Greatest lower bound of a concept assertion.
-------------------------	--

## Module Contents

```
class MinInstanceQuery(  
    concept: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,  
    individual: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,  
)  
  
    Bases: fuzzy_dl_owl2.fuzzydl.query.instance_query.InstanceQuery  
  
    Greatest lower bound of a concept assertion.  
  
    __str__() → str  
        Solves the query over a fuzzy KB  
  
    preprocess(kb: fuzzy_dl_owl2.fuzzydl.knowledge_base.KnowledgeBase) → None  
        Performs some preprocessing steps of the query over a fuzzy KB.  
  
    solve(kb: fuzzy_dl_owl2.fuzzydl.knowledge_base.KnowledgeBase) →  
        fuzzy_dl_owl2.fuzzydl.milp.solution.Solution  
        Solve the query using given knowledge base
```

**fuzzy\_dl\_owl2.fuzzydl.query.min.min\_query**

## Classes

<i>MinQuery</i>	Minimize expression query.
-----------------	----------------------------

## Module Contents

```
class MinQuery(expr: fuzzy_dl_owl2.fuzzydl.milp.expression.Expression)  
  
    Bases: fuzzy_dl_owl2.fuzzydl.query.query.Query  
  
    Minimize expression query.  
  
    __str__() → str  
        Solves the query over a fuzzy KB  
  
    preprocess(kb: fuzzy_dl_owl2.fuzzydl.knowledge_base.KnowledgeBase) → None  
        Performs some preprocessing steps of the query over a fuzzy KB.  
  
    solve(kb: fuzzy_dl_owl2.fuzzydl.knowledge_base.KnowledgeBase) → None  
        Solve the query using given knowledge base  
  
    obj_expr
```

**fuzzy\_dl\_owl2.fuzzydl.query.min.min\_related\_query**

## Classes

<i>MinRelatedQuery</i>	Greatest lower bound of a role assertion (ind1, ind2, role).
------------------------	--

## Module Contents

```
class MinRelatedQuery(  
    a: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,  
    b: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,  
    role_name: str,  
)
```

Bases: *fuzzy\_dl\_owl2.fuzzydl.query.related\_query.RelatedQuery*

Greatest lower bound of a role assertion (ind1, ind2, role).

**\_\_str\_\_**() → str

Solves the query over a fuzzy KB

**preprocess**(kb: *fuzzy\_dl\_owl2.fuzzydl.knowledge\_base.KnowledgeBase*) → None

Performs some preprocessing steps of the query over a fuzzy KB.

**solve**(kb: *fuzzy\_dl\_owl2.fuzzydl.knowledge\_base.KnowledgeBase*) →  
*fuzzy\_dl\_owl2.fuzzydl.milp.solution.Solution*

Solve the query using given knowledge base

**ind1**: *fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual*

**ind2**: *fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual*

**role**: str

**fuzzy\_dl\_owl2.fuzzydl.query.min.min\_satisfiable\_query**

### Classes

<i>MinSatisfiableQuery</i>	Minimal satisfiability degree of a fuzzy concept.
----------------------------	---

### Module Contents

**class MinSatisfiableQuery**(c: *fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*)

**class MinSatisfiableQuery**(c: *fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*, a:  
*fuzzy\_dl\_owl2.fuzzydl.individual.individual.Individual*)

Bases: *fuzzy\_dl\_owl2.fuzzydl.query.satisfiable\_query.SatisfiableQuery*

Minimal satisfiability degree of a fuzzy concept.

**\_\_str\_\_**() → str

Solves the query over a fuzzy KB

**preprocess**(kb: *fuzzy\_dl\_owl2.fuzzydl.knowledge\_base.KnowledgeBase*) → None

Performs some preprocessing steps of the query over a fuzzy KB.

**solve**(kb: *fuzzy\_dl\_owl2.fuzzydl.knowledge\_base.KnowledgeBase*) →  
*fuzzy\_dl\_owl2.fuzzydl.milp.solution.Solution*

Solve the query using given knowledge base

**fuzzy\_dl\_owl2.fuzzydl.query.min.min\_subsumes\_query**

### Classes

<i>MinSubsumesQuery</i>	Minimize subsumption query.
-------------------------	-----------------------------

### Module Contents

**class MinSubsumesQuery**(  
  c1: *fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*,  
  c2: *fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept*,  
  type\_: *fuzzy\_dl\_owl2.fuzzydl.util.constants.LogicOperatorType*,  
)

Bases: *fuzzy\_dl\_owl2.fuzzydl.query.subsumption\_query.SubsumptionQuery*

Minimize subsumption query.

**\_\_str\_\_**() → str

Solves the query over a fuzzy KB

**preprocess**(*kb*: *fuzzy\_dl\_owl2.fuzzydl.knowledge\_base.KnowledgeBase*) → None

Performs some preprocessing steps of the query over a fuzzy KB.

**solve**(*kb*: *fuzzy\_dl\_owl2.fuzzydl.knowledge\_base.KnowledgeBase*) →  
*fuzzy\_dl\_owl2.fuzzydl.milp.solution.Solution*

Solve the query using given knowledge base

## **fuzzy\_dl\_owl2.fuzzydl.query.query**

### **Classes**

<i>Query</i>	Helper class that provides a standard way to create an ABC using
--------------	--

### **Module Contents**

#### **class Query**

Bases: *abc.ABC*

Helper class that provides a standard way to create an ABC using inheritance.

**abstractmethod \_\_str\_\_**() → str

Solves the query over a fuzzy KB

**get\_total\_time**() → float

**abstractmethod preprocess**(*knowledge\_base*:  
*fuzzy\_dl\_owl2.fuzzydl.knowledge\_base.KnowledgeBase*) → None

Performs some preprocessing steps of the query over a fuzzy KB.

**set\_initial\_time**() → None

**set\_total\_time**() → None

**abstractmethod solve**(*knowledge\_base*: *fuzzy\_dl\_owl2.fuzzydl.knowledge\_base.KnowledgeBase*) →  
*fuzzy\_dl\_owl2.fuzzydl.milp.solution.Solution*

Solve the query using given knowledge base

**initial\_time**: int = 0

**total\_time**: int = 0

## **fuzzy\_dl\_owl2.fuzzydl.query.related\_query**

### **Classes**

<i>RelatedQuery</i>	Entailment of a role assertion query
---------------------	--------------------------------------



## Module Contents

### class RelatedQuery

Bases: `fuzzy_dl_owl2.fuzzydl.query.query.Query`, `abc.ABC`

Entailment of a role assertion query

**ind1:** `fuzzy_dl_owl2.fuzzydl.individual.individual.Individual` = None

**ind2:** `fuzzy_dl_owl2.fuzzydl.individual.individual.Individual` = None

**obj\_expr:** `fuzzy_dl_owl2.fuzzydl.milp.expression.Expression` = None

**role:** `str` = None

### fuzzy\_dl\_owl2.fuzzydl.query.satisfiable\_query

#### Classes

<i>SatisfiableQuery</i>	Fuzzy concept satisfiability query.
-------------------------	-------------------------------------

## Module Contents

**class SatisfiableQuery**(*c*: `fuzzy_dl_owl2.fuzzydl.concept.concept.Concept`, *a*: `fuzzy_dl_owl2.fuzzydl.individual.individual.Individual`)

**class SatisfiableQuery**(*c*: `fuzzy_dl_owl2.fuzzydl.concept.concept.Concept`)

Bases: `fuzzy_dl_owl2.fuzzydl.query.query.Query`

Fuzzy concept satisfiability query.

### fuzzy\_dl\_owl2.fuzzydl.query.subsumption\_query

#### Classes

<i>SubsumptionQuery</i>	Helper class that provides a standard way to create an ABC using
-------------------------	--

## Module Contents

**class SubsumptionQuery**(

*c1*: `fuzzy_dl_owl2.fuzzydl.concept.concept.Concept`,  
*c2*: `fuzzy_dl_owl2.fuzzydl.concept.concept.Concept`,  
*s\_type*: `fuzzy_dl_owl2.fuzzydl.util.constants.LogicOperatorType`,

)

Bases: `fuzzy_dl_owl2.fuzzydl.query.query.Query`

Helper class that provides a standard way to create an ABC using inheritance.

**c1:** `fuzzy_dl_owl2.fuzzydl.concept.concept.Concept`

**c2:** `fuzzy_dl_owl2.fuzzydl.concept.concept.Concept`

**obj\_expr:** `fuzzy_dl_owl2.fuzzydl.milp.expression.Expression` = None

**type:** `fuzzy_dl_owl2.fuzzydl.util.constants.LogicOperatorType`

**fuzzy\_dl\_owl2.fuzzydl.range\_axiom****Classes**

<i>RangeAxiom</i>	Role range axiom
-------------------	------------------

**Module Contents**

```
class RangeAxiom(role: str, concept: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept)
    Role range axiom
    concept:   fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
    role:    str
```

**fuzzy\_dl\_owl2.fuzzydl.relation****Classes**

<i>Relation</i>	Represents a role assertion of the form (object individual, role, lower bound for the degree) with respect to a subject individual.
-----------------	---

**Module Contents**

```
class Relation(
    role_name: str,
    ind1: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,
    ind2: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual,
    degree: fuzzy_dl_owl2.fuzzydl.degree.degree.Degree,
)
    Represents a role assertion of the form (object individual, role, lower bound for the degree) with respect to
    a subject individual.
    __repr__() → str
    __str__() → str
    clone() → Self
    get_degree() → fuzzy_dl_owl2.fuzzydl.degree.degree.Degree
    get_name_without_degree() → str
        Gets a printable name of the role assertion without the lower bound
    get_object_individual() → fuzzy_dl_owl2.fuzzydl.individual.individual.Individual
    get_role_name() → str
    get_subject_individual() → fuzzy_dl_owl2.fuzzydl.individual.individual.Individual
    set_object_individual(ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual) → None
    set_subject_individual(ind: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual) → None
    degree:   fuzzy_dl_owl2.fuzzydl.degree.degree.Degree
    ind_a:   fuzzy_dl_owl2.fuzzydl.individual.individual.Individual
```

```
ind_b: fuzzy_dl_owl2.fuzzydl.individual.individual.Individual
role_name: str
```

*fuzzy\_dl\_owl2.fuzzydl.restriction*

### Submodules

*fuzzy\_dl\_owl2.fuzzydl.restriction.has\_value\_restriction*

### Classes

<i>HasValueRestriction</i>	Universal restriction formed by a role, a individual and a lower bound degree.
----------------------------	--

### Module Contents

```
class HasValueRestriction(role_name: str, individual: str, degree:
    fuzzy_dl_owl2.fuzzydl.degree.degree.Degree)
    Bases: fuzzy_dl_owl2.fuzzydl.restriction.restriction.Restriction
    Universal restriction formed by a role, a individual and a lower bound degree.
    get_individual() → str
    get_name_without_degree() → str
        Gets the name of the restriction without the degree.
    ind_name: str
```

*fuzzy\_dl\_owl2.fuzzydl.restriction.restriction*

### Classes

<i>Restriction</i>	Universal restriction formed by a role, a concept and a lower bound degree.
--------------------	---

### Module Contents

```
class Restriction(
    role_name: str,
    concept: fuzzy_dl_owl2.fuzzydl.concept.concept.Concept,
    degree: fuzzy_dl_owl2.fuzzydl.degree.degree.Degree,
)
    Universal restriction formed by a role, a concept and a lower bound degree.
    __repr__() → str
    __str__() → str
    clone() → Self
    get_concept() → fuzzy_dl_owl2.fuzzydl.concept.concept.Concept
    get_degree() → fuzzy_dl_owl2.fuzzydl.degree.degree.Degree
    get_name_without_degree() → str
        Gets the name of the restriction without the degree.
```

`get_role_name()`  $\rightarrow$  str

`concept:` [`fuzzy\_dl\_owl2.fuzzydl.concept.concept.Concept`](#)

`degree:` [`fuzzy\_dl\_owl2.fuzzydl.degree.degree.Degree`](#)

`role_name:` str

[`fuzzy\_dl\_owl2.fuzzydl.role\_parent\_with\_degree`](#)

## Classes

---

<a href="#"><code>RoleParentWithDegree</code></a>
---

Pair of elements (role, degree in [0,1]).
---

---

## Module Contents

**class** `RoleParentWithDegree`(*parent: str, degree: float*)

Pair of elements (role, degree in [0,1]). Given a role, represents a role parent and the inclusion degree.

`get_degree()`  $\rightarrow$  float

`get_parent()`  $\rightarrow$  str

`degree:` float

`parent:` str

[`fuzzy\_dl\_owl2.fuzzydl.util`](#)

## Submodules

[`fuzzy\_dl\_owl2.fuzzydl.util.config\_reader`](#)

## Classes

---

<a href="#"><code>ConfigReader</code></a>
---

## Module Contents

**class** `ConfigReader`

**static** `load_parameters`(*config\_file: str, args: list[str]*)  $\rightarrow$  None

`ANYWHERE_DOUBLE_BLOCKING:` bool = True

`ANYWHERE_SIMPLE_BLOCKING:` bool = True

`DEBUG_PRINT:` bool = True

`EPSILON:` float = 0.001

`MAX_INDIVIDUALS:` int = -1

`MILP_PROVIDER:` [`fuzzy\_dl\_owl2.fuzzydl.util.constants.MILPProvider`](#)

`NUMBER_DIGITS:` int = 2

`OPTIMIZATIONS:` int = 1

```
OWL_ANNOTATION_LABEL: str = 'fuzzyLabel'
```

```
RULE_ACYCLIC_TBOXES: bool = True
```

## fuzzy\_dl\_owl2.fuzzydl.util.constants

### Attributes

*KNOWLEDGE\_BASE\_SEMANTICS*

*MAXVAL*

*MAXVAL2*

*NUMBER*

*RESULTS\_PATH*

*SEPARATOR*

*STAR\_SEPARATOR*

### Classes

<i>BlockingDynamicType</i>	Create a collection of name/value pairs.
<i>ConceptType</i>	Create a collection of name/value pairs.
<i>ConcreteFeatureType</i>	Create a collection of name/value pairs.
<i>CreatedIndividualBlockingType</i>	Create a collection of name/value pairs.
<i>FeatureFunctionType</i>	Create a collection of name/value pairs.
<i>FuzzyDLKeyword</i>	Create a collection of name/value pairs.
<i>FuzzyLogic</i>	Enum where members are also (and must be) strings
<i>InequalityType</i>	Enum where members are also (and must be) strings
<i>KnowledgeBaseRules</i>	Create a collection of name/value pairs.
<i>LogicOperatorType</i>	Create a collection of name/value pairs.
<i>MILPPProvider</i>	Enum where members are also (and must be) strings
<i>RepresentativeIndividualType</i>	Create a collection of name/value pairs.
<i>RestrictionType</i>	Create a collection of name/value pairs.
<i>VariableType</i>	Enum where members are also (and must be) strings

### Module Contents

```
class BlockingDynamicType(*args, **kws)
```

Bases: `enum.Enum`

Create a collection of name/value pairs.

Example enumeration:

```
>>> class Color(Enum):
...     RED = 1
...     BLUE = 2
...     GREEN = 3
```

Access them by:

- attribute access:

```
>>> Color.RED
<Color.RED: 1>
```

- value lookup:

```
>>> Color(1)
<Color.RED: 1>
```

- name lookup:

```
>>> Color['RED']
<Color.RED: 1>
```

Enumerations can be iterated over, and know how many members they have:

```
>>> len(Color)
3
```

```
>>> list(Color)
[<Color.RED: 1>, <Color.BLUE: 2>, <Color.GREEN: 3>]
```

Methods can be added to enumerations, and members can have their own attributes – see the documentation for details.

```
__repr__() → str
```

```
__str__() → str
```

```
ANYWHERE_DOUBLE_BLOCKING = 6
```

```
ANYWHERE_SET_BLOCKING = 5
```

```
ANYWHERE_SUBSET_BLOCKING = 4
```

```
DOUBLE_BLOCKING = 3
```

```
NO_BLOCKING = 0
```

```
SET_BLOCKING = 2
```

```
SUBSET_BLOCKING = 1
```

```
class ConceptType(*args, **kws)
```

Bases: `enum.Enum`

Create a collection of name/value pairs.

Example enumeration:

```
>>> class Color(Enum):
...     RED = 1
...     BLUE = 2
...     GREEN = 3
```

Access them by:

- attribute access:

```
>>> Color.RED
<Color.RED: 1>
```

- value lookup:

```
>>> Color(1)
<Color.RED: 1>
```

- name lookup:

```
>>> Color['RED']
<Color.RED: 1>
```

Enumerations can be iterated over, and know how many members they have:

```
>>> len(Color)
3
```

```
>>> list(Color)
[<Color.RED: 1>, <Color.BLUE: 2>, <Color.GREEN: 3>]
```

Methods can be added to enumerations, and members can have their own attributes – see the documentation for details.

`__repr__()` → str

`__str__()` → str

`ALL` = 7

`AND` = 0

`ATOMIC` = 17

`AT_LEAST_VALUE` = 22

`AT_MOST_VALUE` = 21

`BOTTOM` = 20

`CHOQUET_INTEGRAL` = 49

`COMPLEMENT` = 18

`CONCRETE` = 39

`CONCRETE_COMPLEMENT` = 40

`EXACT_VALUE` = 23

`EXT_NEG_THRESHOLD` = 37

`EXT_POS_THRESHOLD` = 35

`FUZZY_NUMBER` = 44

`FUZZY_NUMBER_COMPLEMENT` = 10

`GOEDEL_AND` = 1

`GOEDEL_IMPLIES` = 15

`GOEDEL_OR` = 4

`HAS_VALUE` = 62

LOOSE\_LOWER\_APPROX = 14  
LOOSE\_UPPER\_APPROX = 13  
LOWER\_APPROX = 9  
LUKASIEWICZ\_AND = 2  
LUKASIEWICZ\_OR = 5  
MODIFIED = 41  
MODIFIED\_COMPLEMENT = 42  
NEG\_THRESHOLD = 33  
NOT\_AT\_LEAST\_VALUE = 25  
NOT\_AT\_MOST\_VALUE = 24  
NOT\_CHOQUET\_INTEGRAL = 52  
NOT\_EXACT\_VALUE = 26  
NOT\_EXT\_NEG\_THRESHOLD = 38  
NOT\_EXT\_POS\_THRESHOLD = 36  
NOT\_GOEDEL\_IMPLIES = 16  
NOT\_HAS\_VALUE = 63  
NOT\_NEG\_THRESHOLD = 34  
NOT\_OWA = 47  
NOT\_POS\_THRESHOLD = 32  
NOT\_QUANTIFIED\_OWA = 48  
NOT\_QUASI\_SUGENO\_INTEGRAL = 54  
NOT\_SELF = 61  
NOT\_SIGMA\_CONCEPT = 67  
NOT\_SUGENO\_INTEGRAL = 53  
NOT\_WEIGHTED = 28  
NOT\_W\_MAX = 56  
NOT\_W\_MIN = 58  
NOT\_W\_SUM = 30  
NOT\_W\_SUM\_ZERO = 60  
NOT\_ZADEH\_IMPLIES = 65  
OR = 3  
OWA = 45  
POS\_THRESHOLD = 31



```

QUANTIFIED_OWA = 46
QUASI_SUGENO_INTEGRAL = 51
SELF = 43
SIGMA_CONCEPT = 66
SOME = 6
SUGENO_INTEGRAL = 50
TIGHT_LOWER_APPROX = 12
TIGHT_UPPER_APPROX = 11
TOP = 19
UPPER_APPROX = 8
WEIGHTED = 27
W_MAX = 55
W_MIN = 57
W_SUM = 29
W_SUM_ZERO = 59
ZADEH_IMPLIES = 64

```

```
class ConcreteFeatureType(*args, **kws)
```

Bases: `enum.Enum`

Create a collection of name/value pairs.

Example enumeration:

```

>>> class Color(Enum):
...     RED = 1
...     BLUE = 2
...     GREEN = 3

```

Access them by:

- attribute access:

```

>>> Color.RED
<Color.RED: 1>

```

- value lookup:

```

>>> Color(1)
<Color.RED: 1>

```

- name lookup:

```

>>> Color['RED']
<Color.RED: 1>

```

Enumerations can be iterated over, and know how many members they have:

```
>>> len(Color)
3
```

```
>>> list(Color)
[<Color.RED: 1>, <Color.BLUE: 2>, <Color.GREEN: 3>]
```

Methods can be added to enumerations, and members can have their own attributes – see the documentation for details.

```
__repr__() → str
```

```
__str__() → str
```

```
BOOLEAN = 3
```

```
INTEGER = 1
```

```
REAL = 2
```

```
STRING = 0
```

```
class CreatedIndividualBlockingType(*args, **kws)
```

Bases: `enum.Enum`

Create a collection of name/value pairs.

Example enumeration:

```
>>> class Color(Enum):
...     RED = 1
...     BLUE = 2
...     GREEN = 3
```

Access them by:

- attribute access:

```
>>> Color.RED
<Color.RED: 1>
```

- value lookup:

```
>>> Color(1)
<Color.RED: 1>
```

- name lookup:

```
>>> Color['RED']
<Color.RED: 1>
```

Enumerations can be iterated over, and know how many members they have:

```
>>> len(Color)
3
```

```
>>> list(Color)
[<Color.RED: 1>, <Color.BLUE: 2>, <Color.GREEN: 3>]
```

Methods can be added to enumerations, and members can have their own attributes – see the documentation for details.

```
__repr__() → str
```

```
__str__() → str
```

```
BLOCKED = 0
```

```
NOT_BLOCKED = 1
```

```
UNCHECKED = 2
```

```
class FeatureFunctionType(*args, **kws)
```

Bases: `enum.Enum`

Create a collection of name/value pairs.

Example enumeration:

```
>>> class Color(Enum):
...     RED = 1
...     BLUE = 2
...     GREEN = 3
```

Access them by:

- attribute access:

```
>>> Color.RED
<Color.RED: 1>
```

- value lookup:

```
>>> Color(1)
<Color.RED: 1>
```

- name lookup:

```
>>> Color['RED']
<Color.RED: 1>
```

Enumerations can be iterated over, and know how many members they have:

```
>>> len(Color)
3
```

```
>>> list(Color)
[<Color.RED: 1>, <Color.BLUE: 2>, <Color.GREEN: 3>]
```

Methods can be added to enumerations, and members can have their own attributes – see the documentation for details.

```
__repr__() → str
```

```
__str__() → str
```

```
ATOMIC = 0
```

```
NUMBER = 1
```

```
PRODUCT = 5
```

**SUBTRACTION = 3**

**SUM = 2**

**class FuzzyDLKeyword(\*args, \*\*kws)**

Bases: `enum.Enum`

Create a collection of name/value pairs.

Example enumeration:

```
>>> class Color(Enum):  
...     RED = 1  
...     BLUE = 2  
...     GREEN = 3
```

Access them by:

- attribute access:

```
>>> Color.RED  
<Color.RED: 1>
```

- value lookup:

```
>>> Color(1)  
<Color.RED: 1>
```

- name lookup:

```
>>> Color['RED']  
<Color.RED: 1>
```

Enumerations can be iterated over, and know how many members they have:

```
>>> len(Color)  
3
```

```
>>> list(Color)  
[<Color.RED: 1>, <Color.BLUE: 2>, <Color.GREEN: 3>]
```

Methods can be added to enumerations, and members can have their own attributes – see the documentation for details.

**\_\_eq\_\_**(*value: object*) → bool

**\_\_repr\_\_**() → str

**\_\_str\_\_**() → str

**get\_name**() → str

**get\_value**() → `pyparsing.CaselessKeyword` | `pyparsing.Literal`

**ALL**

**ALL\_INSTANCES\_QUERY**

**AND**

BINARY  
BNP\_QUERY  
BOOLEAN  
BOTTOM  
CHOQUET  
CLASSICAL  
CONSTRAINTS  
CRISP  
CRISP\_CONCEPT  
CRISP\_ROLE  
DEFINE\_CONCEPT  
DEFINE\_FUZZY\_CONCEPT  
DEFINE\_FUZZY\_EQUIVALENCE  
DEFINE\_FUZZY\_LOGIC  
DEFINE\_FUZZY\_NUMBER  
DEFINE\_FUZZY\_NUMBER\_RANGE  
DEFINE\_FUZZY\_SIMILARITY  
DEFINE\_MODIFIER  
DEFINE\_PRIMITIVE\_CONCEPT  
DEFINE\_TRUTH\_CONSTANT  
DEFUZZIFY\_LOM\_QUERY  
DEFUZZIFY\_MOM\_QUERY  
DEFUZZIFY\_SOM\_QUERY  
DISJOINT  
DISJOINT\_UNION  
DOMAIN  
EQUALS  
EQUIVALENT\_CONCEPTS  
FEATURE\_DIV  
FEATURE\_MUL  
FEATURE\_SUB  
FEATURE\_SUM  
FREE

FUNCTIONAL  
GOEDEL\_AND  
GOEDEL\_IMPLIES  
GOEDEL\_OR  
GREATER\_THAN\_OR\_EQUAL\_TO  
HAS\_VALUE  
IMPLIES  
IMPLIES\_ROLE  
INSTANCE  
INTEGER  
INVERSE  
INVERSE\_FUNCTIONAL  
KLEENE\_DIENES\_IMPLIES  
LEFT\_SHOULDER  
LESS\_THAN\_OR\_EQUAL\_TO  
LINEAR  
LINEAR\_MODIFIER  
LOOSE\_LOWER\_APPROXIMATION  
LOOSE\_UPPER\_APPROXIMATION  
LOWER\_APPROXIMATION  
LUKASIEWICZ  
LUKASIEWICZ\_AND  
LUKASIEWICZ\_IMPLIES  
LUKASIEWICZ\_OR  
MAX\_G\_SUBS\_QUERY  
MAX\_INSTANCE\_QUERY  
MAX\_KD\_SUBS\_QUERY  
MAX\_L\_SUBS\_QUERY  
MAX\_RELATED\_QUERY  
MAX\_SAT\_QUERY  
MAX\_SUBS\_QUERY  
MAX\_VAR\_QUERY  
MIN\_G\_SUBS\_QUERY

MIN\_INSTANCE\_QUERY  
MIN\_KD\_SUBS\_QUERY  
MIN\_L\_SUBS\_QUERY  
MIN\_RELATED\_QUERY  
MIN\_SAT\_QUERY  
MIN\_SUBS\_QUERY  
MIN\_VAR\_QUERY  
MODIFIED  
MUL  
NOT  
OR  
OWA  
QUASI\_SUGENO  
Q\_OWA  
RANGE  
REAL  
REFLEXIVE  
RELATED  
RIGHT\_SHOULDER  
SAT\_QUERY  
SELF  
SHOW\_ABSTRACT\_FILLERS  
SHOW\_ABSTRACT\_FILLERS\_FOR  
SHOW\_CONCEPTS  
SHOW\_CONCRETE\_FILLERS  
SHOW\_CONCRETE\_FILLERS\_FOR  
SHOW\_CONCRETE\_INSTANCE\_FOR  
SHOW\_INSTANCES  
SHOW\_LANGUAGE  
SHOW\_VARIABLES  
SIGMA\_COUNT  
SOME  
STRING

SUB  
SUGENO  
SUM  
SYMMETRIC  
TIGHT\_LOWER\_APPROXIMATION  
TIGHT\_UPPER\_APPROXIMATION  
TOP  
TRANSITIVE  
TRAPEZOIDAL  
TRIANGULAR  
TRIANGULAR\_MODIFIER  
UPPER\_APPROXIMATION  
W\_MAX  
W\_MIN  
W\_SUM  
W\_SUM\_ZERO  
ZADEH  
ZADEH\_IMPLIES

**class FuzzyLogic**

Bases: `enum.StrEnum`

Enum where members are also (and must be) strings

**\_\_repr\_\_**() → str  
Return repr(self).

**\_\_str\_\_**() → str  
Return str(self).

**CLASSICAL** = 'classical'

**LUKASIEWICZ** = 'lukasiewicz'

**ZADEH** = 'zadeh'

**class InequalityType**

Bases: `enum.StrEnum`

Enum where members are also (and must be) strings

**\_\_repr\_\_**() → str  
Return repr(self).

**\_\_str\_\_**() → str  
Return str(self).

**EQUAL** = '='



```
GREATER_THAN = '>'
```

```
LESS_THAN = '<'
```

```
class KnowledgeBaseRules(*args, **kws)
```

```
Bases: enum.Enum
```

Create a collection of name/value pairs.

Example enumeration:

```
>>> class Color(Enum):
...     RED = 1
...     BLUE = 2
...     GREEN = 3
```

Access them by:

- attribute access:

```
>>> Color.RED
<Color.RED: 1>
```

- value lookup:

```
>>> Color(1)
<Color.RED: 1>
```

- name lookup:

```
>>> Color['RED']
<Color.RED: 1>
```

Enumerations can be iterated over, and know how many members they have:

```
>>> len(Color)
3
```

```
>>> list(Color)
[<Color.RED: 1>, <Color.BLUE: 2>, <Color.GREEN: 3>]
```

Methods can be added to enumerations, and members can have their own attributes – see the documentation for details.

```
__repr__() → str
```

```
__str__() → str
```

```
RULE_ATOMIC = 0
```

```
RULE_BOTTOM = 11
```

```
RULE_CHOQUET_INTEGRAL = 30
```

```
RULE_COMPLEMENT = 1
```

```
RULE_CONCRETE = 14
```

```
RULE_DATATYPE = 18
```

```
RULE_FUZZY_NUMBER = 20
RULE_GOEDEL_ALL = 8
RULE_GOEDEL_AND = 2
RULE_GOEDEL_IMPLIES = 12
RULE_GOEDEL_OR = 4
RULE_GOEDEL_SOME = 6
RULE_HAS_VALUE = 44
RULE_LUKASIEWICZ_ALL = 9
RULE_LUKASIEWICZ_AND = 3
RULE_LUKASIEWICZ_OR = 5
RULE_LUKASIEWICZ_SOME = 7
RULE_MODIFIED = 16
RULE_NOT_CHOQUET_INTEGRAL = 31
RULE_NOT_CONCRETE = 15
RULE_NOT_DATATYPE = 19
RULE_NOT_FUZZY_NUMBER = 21
RULE_NOT_GOEDEL_IMPLIES = 13
RULE_NOT_HAS_VALUE = 45
RULE_NOT_MODIFIED = 17
RULE_NOT_OWA = 27
RULE_NOT_QUASI_SUGENO_INTEGRAL = 35
RULE_NOT_SELF = 37
RULE_NOT_SIGMA_COUNT = 49
RULE_NOT_SUGENO_INTEGRAL = 33
RULE_NOT_THRESHOLD = 25
RULE_NOT_WEIGHTED = 23
RULE_NOT_W_MAX = 41
RULE_NOT_W_MIN = 39
RULE_NOT_W_SUM = 29
RULE_NOT_W_SUM_ZERO = 43
RULE_NOT_ZADEH_IMPLIES = 47
RULE_OWA = 26
RULE_QUASI_SUGENO_INTEGRAL = 34
```

```

RULE_SELF = 36
RULE_SIGMA_COUNT = 48
RULE_SUGENO_INTEGRAL = 32
RULE_THRESHOLD = 24
RULE_TOP = 10
RULE_WEIGHTED = 22
RULE_W_MAX = 40
RULE_W_MIN = 38
RULE_W_SUM = 28
RULE_W_SUM_ZERO = 42
RULE_ZADEH_IMPLIES = 46

```

```
class LogicOperatorType(*args, **kws)
```

Bases: `enum.Enum`

Create a collection of name/value pairs.

Example enumeration:

```

>>> class Color(Enum):
...     RED = 1
...     BLUE = 2
...     GREEN = 3

```

Access them by:

- attribute access:

```

>>> Color.RED
<Color.RED: 1>

```

- value lookup:

```

>>> Color(1)
<Color.RED: 1>

```

- name lookup:

```

>>> Color['RED']
<Color.RED: 1>

```

Enumerations can be iterated over, and know how many members they have:

```

>>> len(Color)
3

```

```

>>> list(Color)
[<Color.RED: 1>, <Color.BLUE: 2>, <Color.GREEN: 3>]

```

Methods can be added to enumerations, and members can have their own attributes – see the documentation for details.

```
__repr__() → str
```

```
__str__() → str
```

```
GOEDEL = 1
```

```
KLEENE_DIENES = 2
```

```
LUKASIEWICZ = 0
```

```
ZADEH = 3
```

```
class MILPProvider
```

```
    Bases: enum.StrEnum
```

```
    Enum where members are also (and must be) strings
```

```
    static from_str(value: str) → Self
```

```
    GUROBI
```

```
    MIP
```

```
    PULP
```

```
    PULP_CPLEX
```

```
    PULP_GLPK
```

```
    PULP_HIGHS
```

```
class RepresentativeIndividualType(*args, **kws)
```

```
    Bases: enum.Enum
```

```
    Create a collection of name/value pairs.
```

```
    Example enumeration:
```

```
>>> class Color(Enum):
...     RED = 1
...     BLUE = 2
...     GREEN = 3
```

```
Access them by:
```

- attribute access:

```
>>> Color.RED
<Color.RED: 1>
```

- value lookup:

```
>>> Color(1)
<Color.RED: 1>
```

- name lookup:

```
>>> Color['RED']
<Color.RED: 1>
```

```
Enumerations can be iterated over, and know how many members they have:
```

```
>>> len(Color)
3
```

```
>>> list(Color)
[<Color.RED: 1>, <Color.BLUE: 2>, <Color.GREEN: 3>]
```

Methods can be added to enumerations, and members can have their own attributes – see the documentation for details.

```
__repr__() → str
```

```
__str__() → str
```

```
GREATER_EQUAL = 0
```

```
LESS_EQUAL = 1
```

```
class RestrictionType(*args, **kws)
```

Bases: `enum.Enum`

Create a collection of name/value pairs.

Example enumeration:

```
>>> class Color(Enum):
...     RED = 1
...     BLUE = 2
...     GREEN = 3
```

Access them by:

- attribute access:

```
>>> Color.RED
<Color.RED: 1>
```

- value lookup:

```
>>> Color(1)
<Color.RED: 1>
```

- name lookup:

```
>>> Color['RED']
<Color.RED: 1>
```

Enumerations can be iterated over, and know how many members they have:

```
>>> len(Color)
3
```

```
>>> list(Color)
[<Color.RED: 1>, <Color.BLUE: 2>, <Color.GREEN: 3>]
```

Methods can be added to enumerations, and members can have their own attributes – see the documentation for details.

```
__repr__() → str
```

```
__str__() → str
AT_LEAST_VALUE = 1
AT_MOST_VALUE = 0
EXACT_VALUE = 2
class VariableType
    Bases: enum.StrEnum
    Enum where members are also (and must be) strings
    __repr__() → str
        Return repr(self).
    __str__() → str
        Return str(self).
    BINARY
    CONTINUOUS
    INTEGER
    SEMI_CONTINUOUS
    KNOWLEDGE_BASE_SEMANTICS: FuzzyLogic
    MAXVAL: float = 2147483647000
    MAXVAL2: float = 4294967294000
    NUMBER
    RESULTS_PATH: str
    SEPARATOR: str = '-----'
    STAR_SEPARATOR: str = '*****'
```

fuzzy\_dl\_owl2.fuzzydl.util.util

Attributes

FILENAME
LOG_DIR
TODAY
logger

Classes

Util
------

## Module Contents

### class Util

```

    static debug(message: str) → None
    static error(message: str) → None
    static has_integer_value(d: float) → bool
    static info(message: str) → None
    static log2(n: float) → int
    static order(v: list[Any]) → None
    static round(x: float) → float
    static warning(message: str) → None

```

FILENAME: str

LOG\_DIR: str

TODAY: datetime.datetime

logger

fuzzy\_dl\_owl2.fuzzydl.util.utils

### Attributes

*FULL\_CLASS\_DEBUG\_PRINT*

## Functions

<i>class_debugging()</i>	Decorator to wrap all methods of a class using debugging_wrapper.
<i>debugging_wrapper</i> (cls, func)	Debugging wrapper that prints before and after the method call.
<i>recursion_unlimited</i> (func)	

## Module Contents

### class\_debugging()

Decorator to wrap all methods of a class using debugging\_wrapper.

### debugging\_wrapper(cls, func)

Debugging wrapper that prints before and after the method call.

### recursion\_unlimited(func: Callable)

**FULL\_CLASS\_DEBUG\_PRINT: bool = False**

### 7.1.1.2 fuzzy\_dl\_owl2.fuzzyowl2

#### 7.1.1.2.1 Submodules

fuzzy\_dl\_owl2.fuzzyowl2.fuzzyowl2

#### Classes

---

*FuzzyOwl2*

---

#### Module Contents

```
class FuzzyOwl2(input_file: str, output_file: str, base_iri: str =
                'http://www.semanticweb.org/ontologies/fuzzycl_ontology#')
    Bases: object

    get_atomic_concept_name(c: pyowl2.base.owl_class.OWLClass) → str

    get_atomic_data_property_name(p: pyowl2.expressions.data_property.OWLDataProperty) → str

    get_atomic_object_property_name(p: pyowl2.expressions.object_property.OWLObjectProperty) →
        str

    get_bottom_concept_name() → str

    get_bottom_data_property_name() → str

    get_bottom_object_property_name() → str

    get_class_name(c: pyowl2.abstracts.class_expression.OWLClassExpression) → str

    get_data_all_values_from_name(
        p: pyowl2.abstracts.data_property_expression.OWLDataPropertyExpression,
        range: pyowl2.abstracts.data_range.OWLDataRange,
    ) → str

    get_data_exact_cardinality_restriction(
        cardinality: int,
        p: pyowl2.abstracts.data_property_expression.OWLDataPropertyExpression,
        range: pyowl2.abstracts.data_range.OWLDataRange = None,
    ) → str

    get_data_has_value_name(
        p: pyowl2.abstracts.data_property_expression.OWLDataPropertyExpression,
        literal: pyowl2.literal.literal.OWLLiteral,
    ) → str

    get_data_max_cardinality_restriction(
        cardinality: int,
        p: pyowl2.abstracts.data_property_expression.OWLDataPropertyExpression,
        range: pyowl2.abstracts.data_range.OWLDataRange = None,
    ) → str

    get_data_min_cardinality_restriction(
        cardinality: int,
        p: pyowl2.abstracts.data_property_expression.OWLDataPropertyExpression,
        range: pyowl2.abstracts.data_range.OWLDataRange = None,
    ) → str
```



```

get_data_property_name(p:
    pyowl2.abstracts.data_property_expression.OWLDataPropertyExpression)
    → str

get_data_some_values_from_name(
    p: pyowl2.abstracts.data_property_expression.OWLDataPropertyExpression,
    range: pyowl2.abstracts.data_range.OWLDataRange,
) → str

get_individual_name(i: pyowl2.abstracts.individual.OWLIndividual) → str | None

get_object_all_values_from_name(
    p: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,
    c: pyowl2.abstracts.class_expression.OWLClassExpression,
) → str

get_object_complement_of_name(c: pyowl2.abstracts.class_expression.OWLClassExpression) → str

get_object_exact_cardinality_restriction(
    cardinality: int,
    p: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,
    c: pyowl2.abstracts.class_expression.OWLClassExpression = None,
) → str

get_object_has_self_name(p: pyowl2.abstracts.object_property_
    expression.OWLObjectPropertyExpression) → str

get_object_has_value_name(
    p: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,
    i: pyowl2.abstracts.individual.OWLIndividual,
) → str

get_object_intersection_of_name(operands:
    set[pyowl2.abstracts.class_expression.OWLClassExpression])
    → str

get_object_max_cardinality_restriction(
    cardinality: int,
    p: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,
    c: pyowl2.abstracts.class_expression.OWLClassExpression = None,
) → str

get_object_min_cardinality_restriction(
    cardinality: int,
    p: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,
    c: pyowl2.abstracts.class_expression.OWLClassExpression = None,
) → str

get_object_one_of_name(ind_set: set[pyowl2.abstracts.individual.OWLIndividual]) → str

get_object_property_name(p: pyowl2.abstracts.object_property_
    expression.OWLObjectPropertyExpression) → str

get_object_some_values_from_name(
    p: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,
    c: pyowl2.abstracts.class_expression.OWLClassExpression,
) → str

get_object_union_of_name(operands: set[pyowl2.abstracts.class_expression.OWLClassExpression])
    → str

```

```
get_short_name(e: pyowl2.abstracts.entity.OWLEntity) → str

get_top_concept_name() → str

get_top_data_property_name() → str

get_top_object_property_name() → str

process_concept_annotations() → None

process_datatype_annotations() → None

process_ontology_annotations() → None

process_ontology_axioms() → None

process_property_annotations() → None

translate_owl2ontology() → None

write_asymmetric_object_property_axiom(p: pyowl2.abstracts.object_property_-  
                                         expression.OWLObjectPropertyExpression) → None

write_choquet_concept_definition(name: str, c: fuzzy_dl_owl2.fuzzyowl2.owl_types.choquet_-  
                                         concept.ChoquetConcept) → None

write_concept_assertion_axiom(  
    i: pyowl2.abstracts.individual.OWLIndividual,  
    c: pyowl2.abstracts.class_expression.OWLClassExpression,  
    d: float,  
) → None

write_concept_declaration(c: pyowl2.abstracts.class_expression.OWLClassExpression) → None

write_data_property_assertion_axiom(  
    i: pyowl2.abstracts.individual.OWLIndividual,  
    lit: pyowl2.literal.literal.OWLLiteral,  
    p: pyowl2.abstracts.data_property_expression.OWLDataPropertyExpression,  
    d: float,  
) → None

write_data_property_declaration(dp: pyowl2.abstracts.data_property_-  
                                         expression.OWLDataPropertyExpression) → None

write_data_property_domain_axiom(  
    p: pyowl2.abstracts.data_property_expression.OWLDataPropertyExpression,  
    c: pyowl2.abstracts.class_expression.OWLClassExpression,  
) → None

write_data_property_range_axiom(  
    p: pyowl2.abstracts.data_property_expression.OWLDataPropertyExpression,  
    range: pyowl2.abstracts.data_range.OWLDataRange,  
) → None

write_different_individuals_axiom(ind_set: set[pyowl2.abstracts.individual.OWLIndividual]) →  
                                         None

write_disjoint_classes_axiom(class_set:  
                               set[pyowl2.abstracts.class_expression.OWLClassExpression]) →  
                               None

write_disjoint_data_properties_axiom(  
    class_set: set[pyowl2.abstracts.data_property_expression.OWLDataPropertyExpression],  
) → None
```

```

write_disjoint_object_properties_axiom(
    class_set: set[pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression],
) → None

write_disjoint_union_axiom(class_set:
    set[pyowl2.abstracts.class_expression.OWLClassExpression]) →
    None

write_equivalent_classes_axiom(class_set:
    set[pyowl2.abstracts.class_expression.OWLClassExpression]) →
    None

write_equivalent_data_properties_axiom(
    class_set: set[pyowl2.abstracts.data_property_expression.OWLDataPropertyExpression],
) → None

write_equivalent_object_properties_axiom(
    class_set: set[pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression],
) → None

write_functional_data_property_axiom(p: pyowl2.abstracts.object_property_
    expression.OWLObjectPropertyExpression) → None

write_functional_object_property_axiom(p: pyowl2.abstracts.object_property_
    expression.OWLObjectPropertyExpression) → None

write_fuzzy_logic(logic: str) → None

write_fuzzy_nominal_concept_definition(
    name: str,
    dat: fuzzy_dl_owl2.fuzzyowl2.owl_types.fuzzy_nominal_concept.FuzzyNominalConcept,
) → None

write_inverse_functional_object_property_axiom(
    p: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,
) → None

write_inverse_object_property_axiom(
    p1: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,
    p2: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,
) → None

write_irreflexive_object_property_axiom(p: pyowl2.abstracts.object_property_
    expression.OWLObjectPropertyExpression) → None

write_left_shoulder_function_definition(
    name: str,
    dat: fuzzy_dl_owl2.fuzzyowl2.owl_types.left_shoulder_function.LeftShoulderFunction,
) → None

write_linear_function_definition(name: str, dat: fuzzy_dl_owl2.fuzzyowl2.owl_types.linear_
    function.LinearFunction) → None

write_linear_modifier_definition(name: str, mod: fuzzy_dl_owl2.fuzzyowl2.owl_types.linear_
    modifier.LinearModifier) → None

write_modified_concept_definition(name: str, dat: fuzzy_dl_owl2.fuzzyowl2.owl_
    types.modified_concept.ModifiedConcept) → None

```

```
write_modified_function_definition(name: str, dat: fuzzy_dl_owl2.fuzzyowl2.owl_
                                     types.modified_function.ModifiedFunction) → None

write_modified_property_definition(name: str, dat: fuzzy_dl_owl2.fuzzyowl2.owl_
                                     types.modified_property.ModifiedProperty) → None

write_negative_data_property_assertion_axiom(
    i: pyowl2.abstracts.individual.OWLIndividual,
    lit: pyowl2.literal.literal.OWLLiteral,
    p: pyowl2.abstracts.data_property_expression.OWLDataPropertyExpression,
    d: float,
) → None

write_negative_object_property_assertion_axiom(
    i1: pyowl2.abstracts.individual.OWLIndividual,
    i2: pyowl2.abstracts.individual.OWLIndividual,
    p: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,
    d: float,
) → None

write_object_property_assertion_axiom(
    i1: pyowl2.abstracts.individual.OWLIndividual,
    i2: pyowl2.abstracts.individual.OWLIndividual,
    p: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,
    d: float,
) → None

write_object_property_declaration(op: pyowl2.abstracts.object_property_
                                     expression.OWLObjectPropertyExpression) → None

write_object_property_domain_axiom(
    p: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,
    c: pyowl2.abstracts.class_expression.OWLClassExpression,
) → None

write_object_property_range_axiom(
    p: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,
    c: pyowl2.abstracts.class_expression.OWLClassExpression,
) → None

write_owa_concept_definition(name: str, c:
                                fuzzy_dl_owl2.fuzzyowl2.owl_types.owa_concept.OwaConcept) →
                                None

write_qowa_concept_definition(name: str, c:
                                fuzzy_dl_owl2.fuzzyowl2.owl_types.qowa_concept.QowaConcept)
                                → None

write_quasi_sugeno_concept_definition(
    name: str,
    c: fuzzy_dl_owl2.fuzzyowl2.owl_types.quasi_sugeno_concept.QsugenoConcept,
) → None

write_reflexive_object_property_axiom(p: pyowl2.abstracts.object_property_
                                         expression.OWLObjectPropertyExpression) → None

write_right_shoulder_function_definition(
    name: str,
    dat: fuzzy_dl_owl2.fuzzyowl2.owl_types.right_shoulder_function.RightShoulderFunction,
) → None
```

**write\_same\_individual\_axiom**(*ind\_set*: *set*[*pyowl2.abstracts.individual.OWLIndividual*]) → None

**write\_sub\_data\_property\_of\_axiom**(  
     *subproperty*: *pyowl2.abstracts.data\_property\_expression.OWLDataPropertyExpression*,  
     *superproperty*: *pyowl2.abstracts.data\_property\_expression.OWLDataPropertyExpression*,  
     *d*: *float*,  
 ) → None

**write\_sub\_object\_property\_of\_axiom**(  
     *subproperty*: *pyowl2.abstracts.object\_property\_expression.OWLObjectPropertyExpression*,  
     *superproperty*: *pyowl2.abstracts.object\_property\_expression.OWLObjectPropertyExpression*,  
     *d*: *float*,  
 ) → None

**write\_sub\_property\_chain\_of\_axiom**(  
     *chain*: *list*[*pyowl2.abstracts.object\_property\_expression.OWLObjectPropertyExpression*],  
     *superproperty*: *pyowl2.abstracts.object\_property\_expression.OWLObjectPropertyExpression*,  
     *d*: *float*,  
 ) → None

**write\_subclass\_of\_axiom**(  
     *subclass*: *pyowl2.abstracts.class\_expression.OWLClassExpression*,  
     *superclass*: *pyowl2.abstracts.class\_expression.OWLClassExpression*,  
     *d*: *float*,  
 ) → None

**write\_sugeno\_concept\_definition**(*name*: *str*, *c*: *fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.sugeno\_concept.SugenoConcept*) → None

**write\_symmetric\_object\_property\_axiom**(*p*: *pyowl2.abstracts.object\_property\_expression.OWLObjectPropertyExpression*) → None

**write\_transitive\_object\_property\_axiom**(*p*: *pyowl2.abstracts.object\_property\_expression.OWLObjectPropertyExpression*) → None

**write\_trapezoidal\_function\_definition**(  
     *name*: *str*,  
     *dat*: *fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.trapezoidal\_function.TrapezoidalFunction*,  
 ) → None

**write\_triangular\_function\_definition**(  
     *name*: *str*,  
     *dat*: *fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.triangular\_function.TriangularFunction*,  
 ) → None

**write\_triangular\_modifier\_definition**(  
     *name*: *str*,  
     *mod*: *fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.triangular\_modifier.TriangularModifier*,  
 ) → None

**write\_weighted\_concept\_definition**(*name*: *str*, *c*: *fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.weighted\_concept.WeightedConcept*) → None

**write\_weighted\_max\_concept\_definition**(  
     *name*: *str*,  
     *c*: *fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.weighted\_max\_concept.WeightedMaxConcept*,  
 ) → None

**write\_weighted\_min\_concept\_definition**(  
     *name*: *str*,  
     *c*: *fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.weighted\_min\_concept.WeightedMinConcept*,  
 ) → None

```
write_weighted_sum_concept_definition(  
    name: str,  
    c: fuzzy_dl_owl2.fuzzyowl2.owl_types.weighted_sum_concept.WeightedSumConcept,  
) → None  
  
write_weighted_sum_zero_concept_definition(  
    name: str,  
    c: fuzzy_dl_owl2.fuzzyowl2.owl_types.weighted_sum_zero_concept.WeightedSumZeroConcept,  
) → None  
  
NEG_INFINITY: float = -10000.0  
  
POS_INFINITY: float = 10000.0  
  
defined_concepts: dict[str,  
    fuzzy_dl_owl2.fuzzyowl2.owl_types.concept_definition.ConceptDefinition]  
  
defined_properties: dict[str,  
    fuzzy_dl_owl2.fuzzyowl2.owl_types.concept_definition.ConceptDefinition]  
  
fuzzy_datatypes: dict[str,  
    fuzzy_dl_owl2.fuzzyowl2.owl_types.concept_definition.ConceptDefinition]  
  
fuzzy_label: pyowl2.base.annotation_property.OWLAnnotationProperty  
  
fuzzy_modifiers: dict[str,  
    fuzzy_dl_owl2.fuzzyowl2.owl_types.concept_definition.ConceptDefinition]  
  
ontologies: set[pyowl2.ontology.OWLOntology]  
  
ontology: pyowl2.ontology.OWLOntology  
  
ontology_iri  
  
ontology_path  
  
output_dl: str  
  
processed_axioms: set[str]
```

`fuzzy_dl_owl2.fuzzyowl2.fuzzyowl2_to_fuzzydl`

## Classes

<code>FuzzyOwl2ToFuzzyDL</code>
---------------------------------

## Module Contents

```
class FuzzyOwl2ToFuzzyDL(input_file: str, output_file: str, base_iri: str =  
    'http://www.semanticweb.org/ontologies/fuzzydl_ontology#')  
    Bases: fuzzy_dl_owl2.fuzzyowl2.fuzzyowl2.FuzzyOwl2  
    get_atomic_concept_name(c: pyowl2.base.owl_class.OWLClass) → str  
    get_atomic_data_property_name(p: pyowl2.expressions.data_property.OWLDataProperty) → str
```

```

get_atomic_object_property_name(p: pyowl2.expressions.object_property.OWLObjectProperty) → str

get_bottom_concept_name() → str

get_bottom_data_property_name() → str

get_bottom_object_property_name() → str

get_data_all_values_from_name(
    p: pyowl2.abstracts.data_property_expression.OWLDataPropertyExpression,
    range: pyowl2.abstracts.data_range.OWLDataRange,
) → str

get_data_exact_cardinality_restriction(
    cardinality: int,
    p: pyowl2.abstracts.data_property_expression.OWLDataPropertyExpression,
    range: pyowl2.abstracts.data_range.OWLDataRange = None,
) → str

get_data_has_value_name(
    p: pyowl2.abstracts.data_property_expression.OWLDataPropertyExpression,
    literal: pyowl2.literal.literal.OWLLiteral,
) → str

get_data_max_cardinality_restriction(
    cardinality: int,
    p: pyowl2.abstracts.data_property_expression.OWLDataPropertyExpression,
    range: pyowl2.abstracts.data_range.OWLDataRange = None,
) → str

get_data_min_cardinality_restriction(
    cardinality: int,
    p: pyowl2.abstracts.data_property_expression.OWLDataPropertyExpression,
    range: pyowl2.abstracts.data_range.OWLDataRange = None,
) → str

get_data_some_values_from_name(
    p: pyowl2.abstracts.data_property_expression.OWLDataPropertyExpression,
    range: pyowl2.abstracts.data_range.OWLDataRange,
) → str

get_individual_name(i: pyowl2.abstracts.individual.OWLIndividual) → str

get_object_all_values_from_name(
    p: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,
    c: pyowl2.abstracts.class_expression.OWLClassExpression,
) → str

get_object_complement_of_name(c: pyowl2.abstracts.class_expression.OWLClassExpression) → str

get_object_exact_cardinality_restriction(
    cardinality: int,
    p: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,
    c: pyowl2.abstracts.class_expression.OWLClassExpression = None,
) → str

get_object_has_self_name(p: pyowl2.abstracts.object_property_-
    expression.OWLObjectPropertyExpression) → str

```

```
get_object_has_value_name(  
    p: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,  
    i: pyowl2.abstracts.individual.OWLIndividual,  
) → str  
  
get_object_intersection_of_name(operands:  
    set[pyowl2.abstracts.class_expression.OWLClassExpression])  
    → str  
  
get_object_max_cardinality_restriction(  
    cardinality: int,  
    p: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,  
    c: pyowl2.abstracts.class_expression.OWLClassExpression = None,  
) → str  
  
get_object_min_cardinality_restriction(  
    cardinality: int,  
    p: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,  
    c: pyowl2.abstracts.class_expression.OWLClassExpression = None,  
) → str  
  
get_object_one_of_name(ind_set: set[pyowl2.abstracts.individual.OWLIndividual]) → str  
  
get_object_some_values_from_name(  
    p: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,  
    c: pyowl2.abstracts.class_expression.OWLClassExpression,  
) → str  
  
get_object_union_of_name(operands: set[pyowl2.abstracts.class_expression.OWLClassExpression])  
    → str  
  
get_short_name(s: pyowl2.abstracts.entity.OWLEntity | str)  
  
get_top_concept_name() → str  
  
get_top_data_property_name() → str  
  
get_top_object_property_name() → str  
  
static is_reserved_word(s: str) → bool  
  
write_asymmetric_object_property_axiom(p: pyowl2.abstracts.object_property_  
    expression.OWLObjectPropertyExpression) → None  
  
write_choquet_concept_definition(name: str, c: fuzzy_dl_owl2.fuzzyowl2.owl_types.choquet_  
    concept.ChoquetConcept) → None  
  
write_concept_assertion_axiom(  
    i: pyowl2.abstracts.individual.OWLIndividual,  
    c: pyowl2.abstracts.class_expression.OWLClassExpression,  
    d: float,  
) → None  
  
write_concept_declaration(c: pyowl2.abstracts.class_expression.OWLClassExpression) → None  
  
write_data_property_assertion_axiom(  
    i: pyowl2.abstracts.individual.OWLIndividual,  
    lit: pyowl2.literal.literal.OWLLiteral,  
    p: pyowl2.abstracts.data_property_expression.OWLDataPropertyExpression,  
    d: float,  
) → None
```



```

write_data_property_declaration(dp: pyowl2.abstracts.data_property_-
                                expression.OWLDataPropertyExpression) → None

write_data_property_domain_axiom(
    p: pyowl2.abstracts.data_property_expression.OWLDataPropertyExpression,
    c: pyowl2.abstracts.class_expression.OWLClassExpression,
) → None

write_data_property_range_axiom(
    p: pyowl2.abstracts.data_property_expression.OWLDataPropertyExpression,
    range: pyowl2.abstracts.data_range.OWLDataRange,
) → None

write_different_individuals_axiom(ind_set: set[pyowl2.abstracts.individual.OWLIndividual]) →
    None

write_disjoint_classes_axiom(class_set:
    set[pyowl2.abstracts.class_expression.OWLClassExpression]) →
    None

write_disjoint_data_properties_axiom(
    class_set: set[pyowl2.abstracts.data_property_expression.OWLDataPropertyExpression],
) → None

write_disjoint_object_properties_axiom(
    class_set: set[pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression],
) → None

write_disjoint_union_axiom(class_set:
    set[pyowl2.abstracts.class_expression.OWLClassExpression]) →
    None

write_equivalent_classes_axiom(class_set:
    set[pyowl2.abstracts.class_expression.OWLClassExpression]) →
    None

write_equivalent_data_properties_axiom(
    class_set: set[pyowl2.abstracts.data_property_expression.OWLDataPropertyExpression],
) → None

write_equivalent_object_properties_axiom(
    class_set: set[pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression],
) → None

write_functional_data_property_axiom(p: pyowl2.abstracts.object_property_-
    expression.OWLObjectPropertyExpression) → None

write_functional_object_property_axiom(p: pyowl2.abstracts.object_property_-
    expression.OWLObjectPropertyExpression) → None

write_fuzzy_logic(logic: str) → None

write_fuzzy_nominal_concept_definition(
    name: str,
    dat: fuzzy_dl_owl2.fuzzyowl2.owl_types.fuzzy_nominal_concept.FuzzyNominalConcept,
) → None

write_inverse_functional_object_property_axiom(
    p: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,
) → None

```

```
write_inverse_object_property_axiom(
    p1: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,
    p2: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,
) → None

write_irreflexive_object_property_axiom(p: pyowl2.abstracts.object_property_
    expression.OWLObjectPropertyExpression) → None

write_left_shoulders_function_definition(
    name: str,
    dat: fuzzy_dl_owl2.fuzzyowl2.owl_types.left_shoulders_function.LeftShouldersFunction,
) → None

write_linear_function_definition(name: str, dat: fuzzy_dl_owl2.fuzzyowl2.owl_types.linear_
    function.LinearFunction) → None

write_linear_modifier_definition(name: str, mod: fuzzy_dl_owl2.fuzzyowl2.owl_types.linear_
    modifier.LinearModifier) → None

write_modified_concept_definition(name: str, dat: fuzzy_dl_owl2.fuzzyowl2.owl_
    types.modified_concept.ModifiedConcept) → None

write_modified_function_definition(name: str, dat: fuzzy_dl_owl2.fuzzyowl2.owl_
    types.modified_function.ModifiedFunction) → None

write_modified_property_definition(name: str, dat: fuzzy_dl_owl2.fuzzyowl2.owl_
    types.modified_property.ModifiedProperty) → None

write_negative_data_property_assertion_axiom(
    i: pyowl2.abstracts.individual.OWLIndividual,
    lit: pyowl2.literal.literal.OWLLiteral,
    p: pyowl2.abstracts.data_property_expression.OWLDataPropertyExpression,
    d: float,
) → None

write_negative_object_property_assertion_axiom(
    i1: pyowl2.abstracts.individual.OWLIndividual,
    i2: pyowl2.abstracts.individual.OWLIndividual,
    p: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,
    d: float,
) → None

write_object_property_assertion_axiom(
    i1: pyowl2.abstracts.individual.OWLIndividual,
    i2: pyowl2.abstracts.individual.OWLIndividual,
    p: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,
    d: float,
) → None

write_object_property_declaration(op: pyowl2.abstracts.object_property_
    expression.OWLObjectPropertyExpression) → None

write_object_property_domain_axiom(
    p: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,
    c: pyowl2.abstracts.class_expression.OWLClassExpression,
) → None

write_object_property_range_axiom(
    p: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,
    c: pyowl2.abstracts.class_expression.OWLClassExpression,
) → None
```

```

write_owa_concept_definition(name: str, c:
                               fuzzy_dl_owl2.fuzzyowl2.owl_types.owa_concept.OwaConcept) →
                               None

write_qowa_concept_definition(name: str, c:
                               fuzzy_dl_owl2.fuzzyowl2.owl_types.qowa_concept.QowaConcept)
                               → None

write_quasi_sugeno_concept_definition(
    name: str,
    c: fuzzy_dl_owl2.fuzzyowl2.owl_types.quasi_sugeno_concept.QsugenoConcept,
) → None

write_reflexive_object_property_axiom(p: pyowl2.abstracts.object_property_-
                                       expression.OWLObjectPropertyExpression) → None

write_right_shoulder_function_definition(
    name: str,
    dat: fuzzy_dl_owl2.fuzzyowl2.owl_types.right_shoulder_function.RightShoulderFunction,
) → None

write_same_individual_axiom(ind_set: set[pyowl2.abstracts.individual.OWLIndividual]) → None

write_sub_data_property_of_axiom(
    subproperty: pyowl2.abstracts.data_property_expression.OWLDataPropertyExpression,
    superproperty: pyowl2.abstracts.data_property_expression.OWLDataPropertyExpression,
    d: float,
) → None

write_sub_object_property_of_axiom(
    subproperty: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,
    superproperty: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,
    d: float,
) → None

write_sub_property_chain_of_axiom(
    chain: list[pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression],
    superproperty: pyowl2.abstracts.object_property_expression.OWLObjectPropertyExpression,
    d: float,
) → None

write_subclass_of_axiom(
    subclass: pyowl2.abstracts.class_expression.OWLClassExpression,
    superclass: pyowl2.abstracts.class_expression.OWLClassExpression,
    d: float,
) → None

write_sugeno_concept_definition(name: str, c: fuzzy_dl_owl2.fuzzyowl2.owl_types.sugeno_-
                                concept.SugenoConcept) → None

write_symmetric_object_property_axiom(p: pyowl2.abstracts.object_property_-
                                       expression.OWLObjectPropertyExpression) → None

write_transitive_object_property_axiom(p: pyowl2.abstracts.object_property_-
                                       expression.OWLObjectPropertyExpression) → None

write_trapezoidal_function_definition(
    name: str,
    dat: fuzzy_dl_owl2.fuzzyowl2.owl_types.trapezoidal_function.TrapezoidalFunction,
) → None

```

```
write_triangular_function_definition(  
    name: str,  
    dat: fuzzy_dl_owl2.fuzzyowl2.owl_types.triangular_function.TriangularFunction,  
) → None  
  
write_triangular_modifier_definition(  
    name: str,  
    mod: fuzzy_dl_owl2.fuzzyowl2.owl_types.triangular_modifier.TriangularModifier,  
) → None  
  
write_weighted_concept_definition(name: str, c: fuzzy_dl_owl2.fuzzyowl2.owl_types.weighted_  
    concept.WeightedConcept) → None  
  
write_weighted_max_concept_definition(  
    name: str,  
    c: fuzzy_dl_owl2.fuzzyowl2.owl_types.weighted_max_concept.WeightedMaxConcept,  
) → None  
  
write_weighted_min_concept_definition(  
    name: str,  
    c: fuzzy_dl_owl2.fuzzyowl2.owl_types.weighted_min_concept.WeightedMinConcept,  
) → None  
  
write_weighted_sum_concept_definition(  
    name: str,  
    c: fuzzy_dl_owl2.fuzzyowl2.owl_types.weighted_sum_concept.WeightedSumConcept,  
) → None  
  
write_weighted_sum_zero_concept_definition(  
    name: str,  
    c: fuzzy_dl_owl2.fuzzyowl2.owl_types.weighted_sum_zero_concept.WeightedSumZeroConcept,  
) → None  
  
DOUBLE_MAX_VALUE: float  
  
DOUBLE_MIN_VALUE: float  
  
EPSILON: float = 0.001  
  
INTEGER_MAX_VALUE: int = 1000000000  
  
INTEGER_MIN_VALUE: int = -1000000000  
  
boolean_datatypes: set[str]  
  
data_properties: set[str]  
  
numerical_datatypes: set[str]  
  
object_properties: set[str]  
  
processed_functional_data_properties: set[str]  
  
processed_functional_object_properties: set[str]  
  
string_datatypes: set[str]
```

**fuzzy\_dl\_owl2.fuzzyowl2.owl\_types****Submodules****fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.choquet\_concept****Classes**

<i>ChoquetConcept</i>	Helper class that provides a standard way to create an ABC using
-----------------------	--

**Module Contents**

**class ChoquetConcept**(*weights: list[float], concepts: list[str]*)

Bases: *fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.concept\_definition.ConceptDefinition*

Helper class that provides a standard way to create an ABC using inheritance.

**\_\_str\_\_**() → str

**get\_concepts**() → list[str]

**get\_weights**() → list[float]

**fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.concept\_definition****Classes**

<i>ConceptDefinition</i>	Helper class that provides a standard way to create an ABC using
--------------------------	--

**Module Contents**

**class ConceptDefinition**(*type: fuzzy\_dl\_owl2.fuzzyowl2.util.constants.ConceptType*)

Bases: *abc.ABC*

Helper class that provides a standard way to create an ABC using inheritance.

**\_\_repr\_\_**() → str

**get\_type**() → *fuzzy\_dl\_owl2.fuzzyowl2.util.constants.ConceptType*

**fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.fuzzy\_datatype****Classes**

<i>FuzzyDatatype</i>	Helper class that provides a standard way to create an ABC using
----------------------	--

**Module Contents**

**class FuzzyDatatype**

Bases: *abc.ABC*

Helper class that provides a standard way to create an ABC using inheritance.

```
__repr__() → str
get_max_value() → float
get_min_value() → float
set_max_value(max: float) → None
set_min_value(min: float) → None
```

## `fuzzy_dl_owl2.fuzzyowl2.owl_types.fuzzy_modifier`

### Classes

<i>FuzzyModifier</i>	Helper class that provides a standard way to create an ABC using
----------------------	--

### Module Contents

#### **class FuzzyModifier**

Bases: `abc.ABC`

Helper class that provides a standard way to create an ABC using inheritance.

## `fuzzy_dl_owl2.fuzzyowl2.owl_types.fuzzy_nominal_concept`

### Classes

<i>FuzzyNominalConcept</i>	Helper class that provides a standard way to create an ABC using
----------------------------	--

### Module Contents

#### **class FuzzyNominalConcept**(*n: float, i: str*)

Bases: `fuzzy_dl_owl2.fuzzyowl2.owl_types.concept_definition.ConceptDefinition`

Helper class that provides a standard way to create an ABC using inheritance.

```
__str__() → str
get_degree() → float
get_individual() → str
```

## `fuzzy_dl_owl2.fuzzyowl2.owl_types.fuzzy_property`

### Classes

<i>FuzzyProperty</i>	Helper class that provides a standard way to create an ABC using
----------------------	--

### Module Contents

#### **class FuzzyProperty**

Bases: `abc.ABC`

Helper class that provides a standard way to create an ABC using inheritance.

**fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.left\_shoulder\_function****Classes**

<i>LeftShoulderFunction</i>	Helper class that provides a standard way to create an ABC using
-----------------------------	--

**Module Contents**

**class LeftShoulderFunction**(*a: float, b: float*)

Bases: *fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.fuzzy\_datatype.FuzzyDatatype*

Helper class that provides a standard way to create an ABC using inheritance.

**\_\_str\_\_**() → str

**get\_a**() → float

**get\_b**() → float

**fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.linear\_function****Classes**

<i>LinearFunction</i>	Helper class that provides a standard way to create an ABC using
-----------------------	--

**Module Contents**

**class LinearFunction**(*a: float, b: float*)

Bases: *fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.fuzzy\_datatype.FuzzyDatatype*

Helper class that provides a standard way to create an ABC using inheritance.

**\_\_str\_\_**() → str

**get\_a**() → float

**get\_b**() → float

**fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.linear\_modifier****Classes**

<i>LinearModifier</i>	Helper class that provides a standard way to create an ABC using
-----------------------	--

**Module Contents**

**class LinearModifier**(*c: float*)

Bases: *fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.fuzzy\_modifier.FuzzyModifier*

Helper class that provides a standard way to create an ABC using inheritance.

**\_\_str\_\_**() → str

**get\_c**() → float

**fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.modified\_concept****Classes**

<i>ModifiedConcept</i>	Helper class that provides a standard way to create an ABC using
------------------------	--

**Module Contents****class ModifiedConcept**(*mod: str, c: str*)Bases: *fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.concept\_definition.ConceptDefinition*

Helper class that provides a standard way to create an ABC using inheritance.

**\_\_str\_\_**() → str**get\_fuzzy\_concept**() → str**get\_fuzzy\_modifier**() → str**fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.modified\_function****Classes**

<i>ModifiedFunction</i>	Helper class that provides a standard way to create an ABC using
-------------------------	--

**Module Contents****class ModifiedFunction**(*mod: str, d: str*)Bases: *fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.fuzzy\_datatype.FuzzyDatatype*

Helper class that provides a standard way to create an ABC using inheritance.

**\_\_str\_\_**() → str**get\_d**() → str**get\_mod**() → str**fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.modified\_property****Classes**

<i>ModifiedProperty</i>	Helper class that provides a standard way to create an ABC using
-------------------------	--

**Module Contents****class ModifiedProperty**(*mod: str, prop: str*)Bases: *fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.fuzzy\_property.FuzzyProperty*

Helper class that provides a standard way to create an ABC using inheritance.

**\_\_repr\_\_**() → str**\_\_str\_\_**() → str



`get_fuzzy_modifier()` → str

`get_property()` → str

`fuzzy_dl_owl2.fuzzyowl2.owl_types.owa_concept`

## Classes

<i>OwaConcept</i>	Helper class that provides a standard way to create an ABC using
-------------------	--

## Module Contents

**class** `OwaConcept`(*weights: list[float], concepts: list[str]*)

Bases: `fuzzy_dl_owl2.fuzzyowl2.owl_types.concept_definition.ConceptDefinition`

Helper class that provides a standard way to create an ABC using inheritance.

`__str__()` → str

`get_concepts()` → list[str]

`get_weights()` → list[float]

`fuzzy_dl_owl2.fuzzyowl2.owl_types.property_definition`

## Classes

*PropertyDefinition*

## Module Contents

**class** `PropertyDefinition`(*mod: str, prop: str*)

`get_fuzzy_modifier()` → str

`get_property()` → str

`fuzzy_dl_owl2.fuzzyowl2.owl_types.qowa_concept`

## Classes

<i>QowaConcept</i>	Helper class that provides a standard way to create an ABC using
--------------------	--

## Module Contents

**class** `QowaConcept`(*q: str, concepts: list[str]*)

Bases: `fuzzy_dl_owl2.fuzzyowl2.owl_types.concept_definition.ConceptDefinition`

Helper class that provides a standard way to create an ABC using inheritance.

`__str__()` → str

`get_concepts()` → list[str]

`get_quantifier()` → str

`fuzzy_dl_owl2.fuzzyowl2.owl_types.quasi_sugeno_concept`

### Classes

<i>QsugenoConcept</i>	Helper class that provides a standard way to create an ABC using
-----------------------	--

### Module Contents

**class** `QsugenoConcept`(*weights: list[float], concepts: list[str]*)

Bases: `fuzzy_dl_owl2.fuzzyowl2.owl_types.concept_definition.ConceptDefinition`

Helper class that provides a standard way to create an ABC using inheritance.

`__str__()` → str

`get_concepts()` → list[str]

`get_weights()` → list[float]

`fuzzy_dl_owl2.fuzzyowl2.owl_types.right_shoulder_function`

### Classes

<i>RightShoulderFunction</i>	Helper class that provides a standard way to create an ABC using
------------------------------	--

### Module Contents

**class** `RightShoulderFunction`(*a: float, b: float*)

Bases: `fuzzy_dl_owl2.fuzzyowl2.owl_types.fuzzy_datatype.FuzzyDatatype`

Helper class that provides a standard way to create an ABC using inheritance.

`__str__()` → str

`get_a()` → float

`get_b()` → float

`fuzzy_dl_owl2.fuzzyowl2.owl_types.sugeno_concept`

### Classes

<i>SugenoConcept</i>	Helper class that provides a standard way to create an ABC using
----------------------	--

### Module Contents

**class** `SugenoConcept`(*weights: list[float], concepts: list[str]*)

Bases: `fuzzy_dl_owl2.fuzzyowl2.owl_types.concept_definition.ConceptDefinition`

Helper class that provides a standard way to create an ABC using inheritance.

```

__str__() → str
get_concepts() → list[str]
get_weights() → list[float]

```

## fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.trapezoidal\_function

### Classes

<i>TrapezoidalFunction</i>	Helper class that provides a standard way to create an ABC using
----------------------------	--

### Module Contents

```

class TrapezoidalFunction(a: float, b: float, c: float, d: float)
    Bases: fuzzy_dl_owl2.fuzzyowl2.owl_types.fuzzy_datatype.FuzzyDatatype
    Helper class that provides a standard way to create an ABC using inheritance.
    __str__() → str
    get_a() → float
    get_b() → float
    get_c() → float
    get_d() → float

```

## fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.triangular\_function

### Classes

<i>TriangularFunction</i>	Helper class that provides a standard way to create an ABC using
---------------------------	--

### Module Contents

```

class TriangularFunction(a: float, b: float, c: float)
    Bases: fuzzy_dl_owl2.fuzzyowl2.owl_types.fuzzy_datatype.FuzzyDatatype
    Helper class that provides a standard way to create an ABC using inheritance.
    __str__() → str
    get_a() → float
    get_b() → float
    get_c() → float

```

## fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.triangular\_modifier

### Classes

*TriangularModifier*

Helper class that provides a standard way to create an ABC using

**Module Contents**

**class TriangularModifier**(*a: float, b: float, c: float*)

Bases: *fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.fuzzy\_modifier.FuzzyModifier*

Helper class that provides a standard way to create an ABC using inheritance.

**\_\_str\_\_**() → str

**get\_a**() → float

**get\_b**() → float

**get\_c**() → float

**fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.weighted\_concept**

**Classes***WeightedConcept*

Helper class that provides a standard way to create an ABC using

**Module Contents**

**class WeightedConcept**(*n: float, c: str*)

Bases: *fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.concept\_definition.ConceptDefinition*

Helper class that provides a standard way to create an ABC using inheritance.

**\_\_str\_\_**() → str

**get\_fuzzy\_concept**() → str

**get\_number**() → float

**fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.weighted\_max\_concept**

**Classes***WeightedMaxConcept*

Helper class that provides a standard way to create an ABC using

**Module Contents**

**class WeightedMaxConcept**(*wc: list[fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.concept\_definition.ConceptDefinition]*)

Bases: *fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.concept\_definition.ConceptDefinition*

Helper class that provides a standard way to create an ABC using inheritance.

**\_\_str\_\_**() → str

**get\_weighted\_concepts**() →  
list[fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.concept\_definition.ConceptDefinition]

**fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.weighted\_min\_concept****Classes**

<i>WeightedMinConcept</i>	Helper class that provides a standard way to create an ABC using
---------------------------	--

**Module Contents**

```
class WeightedMinConcept(wc: list[fuzzy_dl_owl2.fuzzyowl2.owl_types.concept_
                                definition.ConceptDefinition])

    Bases: fuzzy_dl_owl2.fuzzyowl2.owl_types.concept_definition.ConceptDefinition

    Helper class that provides a standard way to create an ABC using inheritance.

    __str__() → str

    get_weighted_concepts() →
        list[fuzzy_dl_owl2.fuzzyowl2.owl_types.concept_definition.ConceptDefinition]
```

**fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.weighted\_sum\_concept****Classes**

<i>WeightedSumConcept</i>	Helper class that provides a standard way to create an ABC using
---------------------------	--

**Module Contents**

```
class WeightedSumConcept(wc: list[fuzzy_dl_owl2.fuzzyowl2.owl_types.concept_
                                definition.ConceptDefinition])

    Bases: fuzzy_dl_owl2.fuzzyowl2.owl_types.concept_definition.ConceptDefinition

    Helper class that provides a standard way to create an ABC using inheritance.

    __str__() → str

    get_weighted_concepts() →
        list[fuzzy_dl_owl2.fuzzyowl2.owl_types.concept_definition.ConceptDefinition]
```

**fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.weighted\_sum\_zero\_concept****Classes**

<i>WeightedSumZeroConcept</i>	Helper class that provides a standard way to create an ABC using
-------------------------------	--

**Module Contents**

```
class WeightedSumZeroConcept(wc: list[fuzzy_dl_owl2.fuzzyowl2.owl_types.concept_
                                definition.ConceptDefinition])

    Bases: fuzzy_dl_owl2.fuzzyowl2.owl_types.concept_definition.ConceptDefinition

    Helper class that provides a standard way to create an ABC using inheritance.
```

`__str__()` → str

`get_weighted_concepts()` →  
list[fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.concept\_definition.ConceptDefinition]

fuzzy\_dl\_owl2.fuzzyowl2.parser

## Submodules

fuzzy\_dl\_owl2.fuzzyowl2.parser.owl2\_parser

## Classes

---

*FuzzyOwl2Parser*

---

## Module Contents

**class** FuzzyOwl2Parser

Bases: object

**static** `get_grammatics()` → pyparsing.ParserElement

This function generate the grammatics to parse the predicate with formula “formula”.

### Parameters

**formula** (= *The predicate formula used for the parsing.*)

### Return type

The parsed result given by pyparsing.

**static** `load_config(*args)` → None

**static** `main(`

*annotation: str,*

*\*args,*

`)` → tuple[fuzzy\_dl\_owl2.fuzzydl.knowledge\_base.KnowledgeBase, list[fuzzy\_dl\_owl2.fuzzydl.query.query.Query]]

**static** `parse_string(instring: str, parse_all: bool = False, *(Keyword-only parameters separator (PEP 3102)), parseAll: bool = False)` → pyparsing.ParseResults

fuzzy\_dl\_owl2.fuzzyowl2.parser.owl2\_xml\_parser

## Classes

---

*FuzzyOwl2XMLParser*

---

## Module Contents

**class** FuzzyOwl2XMLParser

Bases: object

**static** `get_caseless_attrrib(attrib: dict[str, str], key: str)` → str | None

**static** `load_config(*args)` → None

```
static main(annotation: str, *args) → Any
```

```
static parse_string(instring: str) → pyparsing.ParseResults
```

**fuzzy\_dl\_owl2.fuzzyowl2.util**

## Submodules

**fuzzy\_dl\_owl2.fuzzyowl2.util.constants**

## Classes

<i>ConceptType</i>	Enum where members are also (and must be) strings
<i>FuzzyOWL2Keyword</i>	Create a collection of name/value pairs.

## Module Contents

### class ConceptType

Bases: `enum.StrEnum`

Enum where members are also (and must be) strings

```
__repr__() → str
```

Return repr(self).

```
__str__() → str
```

Return str(self).

**CHOQUET**

**FUZZY\_NOMINAL**

**MODIFIED\_CONCEPT**

**OWA**

**QUANTIFIED\_OWA**

**QUASI\_SUGENO**

**SUGENO**

**WEIGHTED\_CONCEPT**

**WEIGHTED\_MAX**

**WEIGHTED\_MIN**

**WEIGHTED\_SUM**

**WEIGHTED\_SUM\_ZERO**

### class FuzzyOWL2Keyword(\*args, \*\*kws)

Bases: `enum.Enum`

Create a collection of name/value pairs.

Example enumeration:

```
>>> class Color(Enum) :
...     RED = 1
...     BLUE = 2
...     GREEN = 3
```

Access them by:

- attribute access:

```
>>> Color.RED
<Color.RED: 1>
```

- value lookup:

```
>>> Color(1)
<Color.RED: 1>
```

- name lookup:

```
>>> Color['RED']
<Color.RED: 1>
```

Enumerations can be iterated over, and know how many members they have:

```
>>> len(Color)
3
```

```
>>> list(Color)
[<Color.RED: 1>, <Color.BLUE: 2>, <Color.GREEN: 3>]
```

Methods can be added to enumerations, and members can have their own attributes – see the documentation for details.

`__eq__(value: object) → bool`

`__repr__() → str`

`__str__() → str`

`get_name() → str`

`get_str_value() → str`

`get_tag_name() → str`

`get_value() → pyparsing.CaselessKeyword | pyparsing.Word`

**A**

**AXIOM**

**B**

**BASE**

**C**

**CHOQUET**

**CLOSE\_TAG**

**CONCEPT**

**CONCEPT\_NAMES**

**CRISP**



D  
DATATYPE  
DEGREE\_DEF  
DEGREE\_VALUE  
EQUAL  
FUZZY\_LABEL  
FUZZY\_LOGIC  
FUZZY\_OWL\_2  
FUZZY\_TYPE  
GEQ  
GOEDEL  
GRE  
INDIVIDUAL  
LEFT\_SHOULDER  
LEQ  
LES  
LINEAR  
LOGIC  
LUKASIEWICZ  
MODIFIED  
MODIFIER  
NAME  
NOMINAL  
ONTOLOGY  
OPEN\_TAG  
OWA  
PRODUCT  
QUANTIFIER  
QUASI\_SUGENO  
Q\_OWA  
RIGHT\_SHOULDER  
ROLE  
SINGLE\_CLOSE\_TAG

SLASH  
SUGENO  
TRAPEZOIDAL  
TRIANGULAR  
TYPE  
WEIGHT  
WEIGHTED  
WEIGHTED\_MAXIMUM  
WEIGHTED\_MINIMUM  
WEIGHTED\_SUM  
WEIGHTED\_SUMZERO  
WEIGHTS  
ZADEH

`fuzzy_dl_owl2.fuzzyowl2.util.fuzzy_xml`

## Classes

*FuzzyXML*

## Module Contents

### `class FuzzyXML`

Bases: `object`

`static build_concept_xml`(*concept\_type: str, attrib: dict[str, str] = dict()*) → `xml.etree.ElementTree.Element`

`static build_datatype_xml`(*datatype\_type: str, attrib: dict[str, str] = dict()*) → `xml.etree.ElementTree.Element`

`static build_degree_xml`(*value: int | float, attrib: dict[str, str] = dict()*) → `xml.etree.ElementTree.Element`

`static build_logic_xml`(*logic: str, attrib: dict[str, str] = dict()*) → `xml.etree.ElementTree.Element`

`static build_main_xml`(*fuzzy\_type: str*) → `xml.etree.ElementTree.Element`

`static build_modifier_xml`(*modifier\_type: str, attrib: dict[str, str] = dict()*) → `xml.etree.ElementTree.Element`

`static build_names_xml`(*concepts: list[pyowl2.abstracts.class\_expression.OWLClassExpression]*) → `xml.etree.ElementTree.Element`

`static build_weights_xml`(*weights: list[float]*) → `xml.etree.ElementTree.Element`

`static to_str`(*element: xml.etree.ElementTree.Element*) → `str`

## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

## PYTHON MODULE INDEX

### f

fuzzy\_dl\_owl2, 35  
fuzzy\_dl\_owl2.fuzzydl, 35  
fuzzy\_dl\_owl2.fuzzydl.assertion.assertion, 35  
fuzzy\_dl\_owl2.fuzzydl.assertion.atomic\_assertion, 36  
fuzzy\_dl\_owl2.fuzzydl.classification\_node, 36  
fuzzy\_dl\_owl2.fuzzydl.concept, 37  
fuzzy\_dl\_owl2.fuzzydl.concept.all\_some\_concept, 37  
fuzzy\_dl\_owl2.fuzzydl.concept.approximation\_concept, 38  
fuzzy\_dl\_owl2.fuzzydl.concept.atomic\_concept, 39  
fuzzy\_dl\_owl2.fuzzydl.concept.choquet\_integral, 40  
fuzzy\_dl\_owl2.fuzzydl.concept.concept, 41  
fuzzy\_dl\_owl2.fuzzydl.concept.concrete, 43  
fuzzy\_dl\_owl2.fuzzydl.concept.concrete.crisp\_-concrete\_concept, 43  
fuzzy\_dl\_owl2.fuzzydl.concept.concrete.fuzzy\_-concrete\_concept, 44  
fuzzy\_dl\_owl2.fuzzydl.concept.concrete.fuzzy\_number, 45  
fuzzy\_dl\_owl2.fuzzydl.concept.concrete.fuzzy\_-number.triangular\_fuzzy\_number, 45  
fuzzy\_dl\_owl2.fuzzydl.concept.concrete.left\_-concrete\_concept, 46  
fuzzy\_dl\_owl2.fuzzydl.concept.concrete.linear\_-concrete\_concept, 46  
fuzzy\_dl\_owl2.fuzzydl.concept.concrete.modified\_-concrete\_concept, 47  
fuzzy\_dl\_owl2.fuzzydl.concept.concrete.right\_-concrete\_concept, 48  
fuzzy\_dl\_owl2.fuzzydl.concept.concrete.trapezoidal\_-concrete\_concept, 48  
fuzzy\_dl\_owl2.fuzzydl.concept.concrete.triangular\_-concrete\_concept, 49  
fuzzy\_dl\_owl2.fuzzydl.concept.ext\_threshold\_concept, 50  
fuzzy\_dl\_owl2.fuzzydl.concept.has\_value\_concept, 51  
fuzzy\_dl\_owl2.fuzzydl.concept.implies\_concept, 51  
fuzzy\_dl\_owl2.fuzzydl.concept.interface, 53  
fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_-concept\_interface, 53  
fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_-concepts\_interface, 53  
fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_role\_-concept\_interface, 54  
fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_role\_-interface, 54  
fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_value\_-interface, 54  
fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_-weighted\_concepts\_interface, 54  
fuzzy\_dl\_owl2.fuzzydl.concept.modified, 55  
fuzzy\_dl\_owl2.fuzzydl.concept.modified.linearly\_-modified\_concept, 55  
fuzzy\_dl\_owl2.fuzzydl.concept.modified.modified\_-concept, 55  
fuzzy\_dl\_owl2.fuzzydl.concept.modified.triangularly\_-modified\_concept, 56  
fuzzy\_dl\_owl2.fuzzydl.concept.negated\_nominal, 57  
fuzzy\_dl\_owl2.fuzzydl.concept.operator\_concept, 57  
fuzzy\_dl\_owl2.fuzzydl.concept.owa\_concept, 61  
fuzzy\_dl\_owl2.fuzzydl.concept.qowa\_concept, 61  
fuzzy\_dl\_owl2.fuzzydl.concept.quasi\_sugeno\_integral, 62  
fuzzy\_dl\_owl2.fuzzydl.concept.self\_concept, 63  
fuzzy\_dl\_owl2.fuzzydl.concept.sigma\_concept, 63  
fuzzy\_dl\_owl2.fuzzydl.concept.sigma\_count, 64  
fuzzy\_dl\_owl2.fuzzydl.concept.string\_concept, 65  
fuzzy\_dl\_owl2.fuzzydl.concept.sugeno\_integral, 65  
fuzzy\_dl\_owl2.fuzzydl.concept.threshold\_concept, 66  
fuzzy\_dl\_owl2.fuzzydl.concept.truth\_concept, 67  
fuzzy\_dl\_owl2.fuzzydl.concept.value\_concept, 68  
fuzzy\_dl\_owl2.fuzzydl.concept.weighted\_concept, 69  
fuzzy\_dl\_owl2.fuzzydl.concept.weighted\_max\_concept, 70  
fuzzy\_dl\_owl2.fuzzydl.concept.weighted\_min\_concept, 70  
fuzzy\_dl\_owl2.fuzzydl.concept.weighted\_sum\_concept, 71  
fuzzy\_dl\_owl2.fuzzydl.concept.weighted\_sum\_zero\_-concept, 72  
fuzzy\_dl\_owl2.fuzzydl.concept.equivalence, 72  
fuzzy\_dl\_owl2.fuzzydl.concrete\_feature, 73  
fuzzy\_dl\_owl2.fuzzydl.degree, 73  
fuzzy\_dl\_owl2.fuzzydl.degree.degree, 73  
fuzzy\_dl\_owl2.fuzzydl.degree.degree\_expression, 74  
fuzzy\_dl\_owl2.fuzzydl.degree.degree\_numeric, 75  
fuzzy\_dl\_owl2.fuzzydl.degree.degree\_variable, 76  
fuzzy\_dl\_owl2.fuzzydl.domain\_axiom, 76  
fuzzy\_dl\_owl2.fuzzydl.exception, 77  
fuzzy\_dl\_owl2.fuzzydl.exception.fuzzy\_ontology\_-exception, 77  
fuzzy\_dl\_owl2.fuzzydl.exception.inconsistent\_-ontology\_exception, 77  
fuzzy\_dl\_owl2.fuzzydl.feature\_function, 77  
fuzzy\_dl\_owl2.fuzzydl.fuzzydl\_to\_owl2, 78  
fuzzy\_dl\_owl2.fuzzydl.general\_concept\_inclusion, 80  
fuzzy\_dl\_owl2.fuzzydl.individual, 81  
fuzzy\_dl\_owl2.fuzzydl.individual.created\_individual, 81  
fuzzy\_dl\_owl2.fuzzydl.individual.individual, 82  
fuzzy\_dl\_owl2.fuzzydl.individual.representative\_-individual, 83  
fuzzy\_dl\_owl2.fuzzydl.knowledge\_base, 84  
fuzzy\_dl\_owl2.fuzzydl.label, 120  
fuzzy\_dl\_owl2.fuzzydl.milp, 121  
fuzzy\_dl\_owl2.fuzzydl.milp.expression, 121  
fuzzy\_dl\_owl2.fuzzydl.milp.inequation, 122  
fuzzy\_dl\_owl2.fuzzydl.milp.milp\_helper, 123  
fuzzy\_dl\_owl2.fuzzydl.milp.show\_variables\_helper, 127  
fuzzy\_dl\_owl2.fuzzydl.milp.solution, 129  
fuzzy\_dl\_owl2.fuzzydl.milp.term, 130  
fuzzy\_dl\_owl2.fuzzydl.milp.variable, 130  
fuzzy\_dl\_owl2.fuzzydl.modifier, 132  
fuzzy\_dl\_owl2.fuzzydl.modifier.linear\_modifier, 132  
fuzzy\_dl\_owl2.fuzzydl.modifier.modifier, 133

[fuzzy\\_dl\\_owl2.fuzzydl.modifier.triangular\\_modifier](#), 134  
[fuzzy\\_dl\\_owl2.fuzzydl.parser](#), 134  
[fuzzy\\_dl\\_owl2.fuzzydl.parser.dl\\_parser](#), 134  
[fuzzy\\_dl\\_owl2.fuzzydl.primitive\\_concept\\_definition](#), 135  
[fuzzy\\_dl\\_owl2.fuzzydl.query](#), 136  
[fuzzy\\_dl\\_owl2.fuzzydl.query.all\\_instances\\_query](#), 136  
[fuzzy\\_dl\\_owl2.fuzzydl.query.bnp\\_query](#), 137  
[fuzzy\\_dl\\_owl2.fuzzydl.query.classification\\_query](#), 138  
[fuzzy\\_dl\\_owl2.fuzzydl.query.defuzzify](#), 138  
[fuzzy\\_dl\\_owl2.fuzzydl.query.defuzzify.defuzzify\\_-query](#), 138  
[fuzzy\\_dl\\_owl2.fuzzydl.query.defuzzify.lom\\_-defuzzify\\_query](#), 139  
[fuzzy\\_dl\\_owl2.fuzzydl.query.defuzzify.mom\\_-defuzzify\\_query](#), 139  
[fuzzy\\_dl\\_owl2.fuzzydl.query.defuzzify.som\\_-defuzzify\\_query](#), 140  
[fuzzy\\_dl\\_owl2.fuzzydl.query.instance\\_query](#), 140  
[fuzzy\\_dl\\_owl2.fuzzydl.query.kb\\_satisfiable\\_query](#), 140  
[fuzzy\\_dl\\_owl2.fuzzydl.query.max](#), 141  
[fuzzy\\_dl\\_owl2.fuzzydl.query.max.max\\_instance\\_query](#), 141  
[fuzzy\\_dl\\_owl2.fuzzydl.query.max.max\\_query](#), 141  
[fuzzy\\_dl\\_owl2.fuzzydl.query.max.max\\_related\\_query](#), 142  
[fuzzy\\_dl\\_owl2.fuzzydl.query.max.max\\_satisfiable\\_-query](#), 142  
[fuzzy\\_dl\\_owl2.fuzzydl.query.max.max\\_subsumes\\_query](#), 143  
[fuzzy\\_dl\\_owl2.fuzzydl.query.min](#), 143  
[fuzzy\\_dl\\_owl2.fuzzydl.query.min.min\\_instance\\_query](#), 143  
[fuzzy\\_dl\\_owl2.fuzzydl.query.min.min\\_query](#), 144  
[fuzzy\\_dl\\_owl2.fuzzydl.query.min.min\\_related\\_query](#), 144  
[fuzzy\\_dl\\_owl2.fuzzydl.query.min.min\\_satisfiable\\_-query](#), 145  
[fuzzy\\_dl\\_owl2.fuzzydl.query.min.min\\_subsumes\\_query](#), 145  
[fuzzy\\_dl\\_owl2.fuzzydl.query.query](#), 146  
[fuzzy\\_dl\\_owl2.fuzzydl.query.related\\_query](#), 146  
[fuzzy\\_dl\\_owl2.fuzzydl.query.satisfiable\\_query](#), 147  
[fuzzy\\_dl\\_owl2.fuzzydl.query.subsumption\\_query](#), 147  
[fuzzy\\_dl\\_owl2.fuzzydl.range\\_axiom](#), 148  
[fuzzy\\_dl\\_owl2.fuzzydl.relation](#), 148  
[fuzzy\\_dl\\_owl2.fuzzydl.restriction](#), 149  
[fuzzy\\_dl\\_owl2.fuzzydl.restriction.has\\_value\\_-restriction](#), 149  
[fuzzy\\_dl\\_owl2.fuzzydl.restriction.restriction](#), 149  
[fuzzy\\_dl\\_owl2.fuzzydl.role\\_parent\\_with\\_degree](#), 150  
[fuzzy\\_dl\\_owl2.fuzzydl.util](#), 150  
[fuzzy\\_dl\\_owl2.fuzzydl.util.config\\_reader](#), 150  
[fuzzy\\_dl\\_owl2.fuzzydl.util.constants](#), 151  
[fuzzy\\_dl\\_owl2.fuzzydl.util.util](#), 168  
[fuzzy\\_dl\\_owl2.fuzzydl.util.utils](#), 169  
[fuzzy\\_dl\\_owl2.fuzzyowl2](#), 170  
[fuzzy\\_dl\\_owl2.fuzzyowl2.fuzzyowl2](#), 170  
[fuzzy\\_dl\\_owl2.fuzzyowl2.fuzzyowl2\\_to\\_fuzzydl](#), 176  
[fuzzy\\_dl\\_owl2.fuzzyowl2.owl\\_types](#), 183  
[fuzzy\\_dl\\_owl2.fuzzyowl2.owl\\_types.choquet\\_concept](#), 183  
[fuzzy\\_dl\\_owl2.fuzzyowl2.owl\\_types.concept\\_definition](#), 183  
[fuzzy\\_dl\\_owl2.fuzzyowl2.owl\\_types.fuzzy\\_datatype](#), 183  
[fuzzy\\_dl\\_owl2.fuzzyowl2.owl\\_types.fuzzy\\_modifier](#), 184  
[fuzzy\\_dl\\_owl2.fuzzyowl2.owl\\_types.fuzzy\\_nominal\\_-concept](#), 184  
[fuzzy\\_dl\\_owl2.fuzzyowl2.owl\\_types.fuzzy\\_property](#), 184  
[fuzzy\\_dl\\_owl2.fuzzyowl2.owl\\_types.left\\_shoulder\\_-function](#), 185  
[fuzzy\\_dl\\_owl2.fuzzyowl2.owl\\_types.linear\\_function](#), 185  
[fuzzy\\_dl\\_owl2.fuzzyowl2.owl\\_types.linear\\_modifier](#), 185  
[fuzzy\\_dl\\_owl2.fuzzyowl2.owl\\_types.modified\\_concept](#), 186  
[fuzzy\\_dl\\_owl2.fuzzyowl2.owl\\_types.modified\\_function](#), 186  
[fuzzy\\_dl\\_owl2.fuzzyowl2.owl\\_types.modified\\_property](#), 186  
[fuzzy\\_dl\\_owl2.fuzzyowl2.owl\\_types.owa\\_concept](#), 187  
[fuzzy\\_dl\\_owl2.fuzzyowl2.owl\\_types.property\\_-definition](#), 187  
[fuzzy\\_dl\\_owl2.fuzzyowl2.owl\\_types.qowa\\_concept](#), 187  
[fuzzy\\_dl\\_owl2.fuzzyowl2.owl\\_types.quasi\\_sugeno\\_-concept](#), 188  
[fuzzy\\_dl\\_owl2.fuzzyowl2.owl\\_types.right\\_shoulder\\_-function](#), 188  
[fuzzy\\_dl\\_owl2.fuzzyowl2.owl\\_types.sugeno\\_concept](#), 188  
[fuzzy\\_dl\\_owl2.fuzzyowl2.owl\\_types.trapezoidal\\_-function](#), 189  
[fuzzy\\_dl\\_owl2.fuzzyowl2.owl\\_types.triangular\\_-function](#), 189  
[fuzzy\\_dl\\_owl2.fuzzyowl2.owl\\_types.triangular\\_modifier](#), 189  
[fuzzy\\_dl\\_owl2.fuzzyowl2.owl\\_types.weighted\\_concept](#), 190  
[fuzzy\\_dl\\_owl2.fuzzyowl2.owl\\_types.weighted\\_max\\_-concept](#), 190  
[fuzzy\\_dl\\_owl2.fuzzyowl2.owl\\_types.weighted\\_min\\_-concept](#), 191  
[fuzzy\\_dl\\_owl2.fuzzyowl2.owl\\_types.weighted\\_sum\\_-concept](#), 191  
[fuzzy\\_dl\\_owl2.fuzzyowl2.owl\\_types.weighted\\_sum\\_-zero\\_concept](#), 191  
[fuzzy\\_dl\\_owl2.fuzzyowl2.parser](#), 192  
[fuzzy\\_dl\\_owl2.fuzzyowl2.parser.owl2\\_parser](#), 192  
[fuzzy\\_dl\\_owl2.fuzzyowl2.parser.owl2\\_xml\\_parser](#), 192  
[fuzzy\\_dl\\_owl2.fuzzyowl2.util](#), 193  
[fuzzy\\_dl\\_owl2.fuzzyowl2.util.constants](#), 193  
[fuzzy\\_dl\\_owl2.fuzzyowl2.util.fuzzy\\_xml](#), 196

## Symbols

- \_\_add\_\_ () (*Expression method*), 121
- \_\_add\_\_ () (*Term method*), 130
- \_\_add\_\_ () (*TriangularFuzzyNumber method*), 45
- \_\_and\_\_ () (*ApproximationConcept method*), 39
- \_\_and\_\_ () (*AtomicConcept method*), 40
- \_\_and\_\_ () (*ChoquetIntegral method*), 41
- \_\_and\_\_ () (*Concept method*), 41
- \_\_and\_\_ () (*CrispConcreteConcept method*), 44
- \_\_and\_\_ () (*ExtThresholdConcept method*), 50
- \_\_and\_\_ () (*HasValueConcept method*), 51
- \_\_and\_\_ () (*ImpliesConcept method*), 52
- \_\_and\_\_ () (*LeftConcreteConcept method*), 46
- \_\_and\_\_ () (*LinearConcreteConcept method*), 47
- \_\_and\_\_ () (*LinearModifier method*), 132
- \_\_and\_\_ () (*LinearlyModifiedConcept method*), 55
- \_\_and\_\_ () (*ModifiedConcept method*), 56
- \_\_and\_\_ () (*ModifiedConcreteConcept method*), 47
- \_\_and\_\_ () (*NegatedNominal method*), 57
- \_\_and\_\_ () (*OperatorConcept method*), 58
- \_\_and\_\_ () (*OwaConcept method*), 61
- \_\_and\_\_ () (*QowaConcept method*), 61
- \_\_and\_\_ () (*QsugenoIntegral method*), 62
- \_\_and\_\_ () (*RightConcreteConcept method*), 48
- \_\_and\_\_ () (*SelfConcept method*), 63
- \_\_and\_\_ () (*SigmaConcept method*), 63
- \_\_and\_\_ () (*SugenoIntegral method*), 66
- \_\_and\_\_ () (*ThresholdConcept method*), 66
- \_\_and\_\_ () (*TrapezoidalConcreteConcept method*), 49
- \_\_and\_\_ () (*TriangularConcreteConcept method*), 49
- \_\_and\_\_ () (*TriangularFuzzyNumber method*), 45
- \_\_and\_\_ () (*TriangularModifier method*), 134
- \_\_and\_\_ () (*TriangularlyModifiedConcept method*), 56
- \_\_and\_\_ () (*TruthConcept method*), 67
- \_\_and\_\_ () (*ValueConcept method*), 68
- \_\_and\_\_ () (*WeightedConcept method*), 69
- \_\_and\_\_ () (*WeightedMaxConcept method*), 70
- \_\_and\_\_ () (*WeightedMinConcept method*), 70
- \_\_and\_\_ () (*WeightedSumConcept method*), 71
- \_\_and\_\_ () (*WeightedSumZeroConcept method*), 72
- \_\_eq\_\_ () (*Assertion method*), 35
- \_\_eq\_\_ () (*AtomicConcept method*), 40
- \_\_eq\_\_ () (*Concept method*), 41
- \_\_eq\_\_ () (*CreatedIndividual method*), 81
- \_\_eq\_\_ () (*Degree method*), 73
- \_\_eq\_\_ () (*DegreeExpression method*), 74
- \_\_eq\_\_ () (*DegreeNumeric method*), 75
- \_\_eq\_\_ () (*DegreeVariable method*), 76
- \_\_eq\_\_ () (*Expression method*), 121
- \_\_eq\_\_ () (*FuzzyDLKeyword method*), 158
- \_\_eq\_\_ () (*FuzzyOWL2Keyword method*), 194
- \_\_eq\_\_ () (*GeneralConceptInclusion method*), 80
- \_\_eq\_\_ () (*ImpliesConcept method*), 52
- \_\_eq\_\_ () (*Individual method*), 82
- \_\_eq\_\_ () (*Inequation method*), 122
- \_\_eq\_\_ () (*Label method*), 120
- \_\_eq\_\_ () (*OperatorConcept method*), 58
- \_\_eq\_\_ () (*PrimitiveConceptDefinition method*), 136
- \_\_eq\_\_ () (*Term method*), 130
- \_\_eq\_\_ () (*Thing method*), 42
- \_\_eq\_\_ () (*TriangularFuzzyNumber method*), 45
- \_\_eq\_\_ () (*TruthConcept method*), 67
- \_\_eq\_\_ () (*Variable method*), 131
- \_\_ge\_\_ () (*CreatedIndividual method*), 81
- \_\_ge\_\_ () (*GeneralConceptInclusion method*), 80
- \_\_ge\_\_ () (*PrimitiveConceptDefinition method*), 136
- \_\_ge\_\_ () (*Thing method*), 42
- \_\_gt\_\_ () (*CreatedIndividual method*), 81
- \_\_gt\_\_ () (*GeneralConceptInclusion method*), 80
- \_\_gt\_\_ () (*PrimitiveConceptDefinition method*), 136
- \_\_gt\_\_ () (*Thing method*), 42
- \_\_hash\_\_ () (*AllSomeConcept method*), 37
- \_\_hash\_\_ () (*ApproximationConcept method*), 39
- \_\_hash\_\_ () (*AtomicConcept method*), 40
- \_\_hash\_\_ () (*ChoquetIntegral method*), 41
- \_\_hash\_\_ () (*ClassificationNode method*), 36
- \_\_hash\_\_ () (*CreatedIndividual method*), 81
- \_\_hash\_\_ () (*CrispConcreteConcept method*), 44
- \_\_hash\_\_ () (*Expression method*), 121
- \_\_hash\_\_ () (*ExtThresholdConcept method*), 50
- \_\_hash\_\_ () (*GeneralConceptInclusion method*), 80
- \_\_hash\_\_ () (*HasValueConcept method*), 51
- \_\_hash\_\_ () (*ImpliesConcept method*), 52
- \_\_hash\_\_ () (*Inequation method*), 122
- \_\_hash\_\_ () (*LeftConcreteConcept method*), 46
- \_\_hash\_\_ () (*LinearConcreteConcept method*), 47
- \_\_hash\_\_ () (*LinearModifier method*), 132
- \_\_hash\_\_ () (*LinearlyModifiedConcept method*), 55
- \_\_hash\_\_ () (*ModifiedConcreteConcept method*), 47
- \_\_hash\_\_ () (*NegatedNominal method*), 57
- \_\_hash\_\_ () (*OperatorConcept method*), 58
- \_\_hash\_\_ () (*OwaConcept method*), 61
- \_\_hash\_\_ () (*PrimitiveConceptDefinition method*), 136
- \_\_hash\_\_ () (*QowaConcept method*), 62
- \_\_hash\_\_ () (*QsugenoIntegral method*), 62
- \_\_hash\_\_ () (*RightConcreteConcept method*), 48
- \_\_hash\_\_ () (*SelfConcept method*), 63
- \_\_hash\_\_ () (*SigmaConcept method*), 64
- \_\_hash\_\_ () (*SigmaCount method*), 64
- \_\_hash\_\_ () (*Solution method*), 129
- \_\_hash\_\_ () (*StringConcept method*), 65
- \_\_hash\_\_ () (*SugenoIntegral method*), 66
- \_\_hash\_\_ () (*Term method*), 130
- \_\_hash\_\_ () (*ThresholdConcept method*), 66
- \_\_hash\_\_ () (*TrapezoidalConcreteConcept method*), 49
- \_\_hash\_\_ () (*TriangularConcreteConcept method*), 49
- \_\_hash\_\_ () (*TriangularFuzzyNumber method*), 45
- \_\_hash\_\_ () (*TriangularModifier method*), 134
- \_\_hash\_\_ () (*TriangularlyModifiedConcept method*), 56
- \_\_hash\_\_ () (*TruthConcept method*), 67
- \_\_hash\_\_ () (*ValueConcept method*), 68
- \_\_hash\_\_ () (*Variable method*), 131
- \_\_hash\_\_ () (*WeightedConcept method*), 69
- \_\_hash\_\_ () (*WeightedMaxConcept method*), 70
- \_\_hash\_\_ () (*WeightedMinConcept method*), 70
- \_\_hash\_\_ () (*WeightedSumConcept method*), 71

\_\_hash\_\_() (WeightedSumZeroConcept method), 72  
 \_\_iand\_\_() (Concept method), 41  
 \_\_invert\_\_() (AtomicConcept method), 40  
 \_\_invert\_\_() (Thing method), 42  
 \_\_ior\_\_() (Concept method), 41  
 \_\_irshift\_\_() (Concept method), 41  
 \_\_le\_\_() (CreatedIndividual method), 81  
 \_\_le\_\_() (GeneralConceptInclusion method), 80  
 \_\_le\_\_() (PrimitiveConceptDefinition method), 136  
 \_\_le\_\_() (Thing method), 42  
 \_\_lt\_\_() (CreatedIndividual method), 81  
 \_\_lt\_\_() (GeneralConceptInclusion method), 80  
 \_\_lt\_\_() (PrimitiveConceptDefinition method), 136  
 \_\_lt\_\_() (Thing method), 42  
 \_\_mul\_\_() (Expression method), 121  
 \_\_mul\_\_() (Term method), 130  
 \_\_mul\_\_() (TriangularFuzzyNumber method), 45  
 \_\_ne\_\_() (Assertion method), 35  
 \_\_ne\_\_() (AtomicConcept method), 40  
 \_\_ne\_\_() (Concept method), 41  
 \_\_ne\_\_() (CreatedIndividual method), 81  
 \_\_ne\_\_() (Degree method), 73  
 \_\_ne\_\_() (Expression method), 121  
 \_\_ne\_\_() (GeneralConceptInclusion method), 80  
 \_\_ne\_\_() (Individual method), 82  
 \_\_ne\_\_() (Inequation method), 122  
 \_\_ne\_\_() (Label method), 120  
 \_\_ne\_\_() (OperatorConcept method), 58  
 \_\_ne\_\_() (PrimitiveConceptDefinition method), 136  
 \_\_ne\_\_() (Term method), 130  
 \_\_ne\_\_() (Thing method), 42  
 \_\_ne\_\_() (TriangularFuzzyNumber method), 45  
 \_\_ne\_\_() (TruthConcept method), 67  
 \_\_ne\_\_() (Variable method), 131  
 \_\_neg\_\_() (AllSomeConcept method), 37  
 \_\_neg\_\_() (ApproximationConcept method), 39  
 \_\_neg\_\_() (AtomicConcept method), 40  
 \_\_neg\_\_() (ChoquetIntegral method), 41  
 \_\_neg\_\_() (CrispConcreteConcept method), 44  
 \_\_neg\_\_() (Expression method), 121  
 \_\_neg\_\_() (ExtThresholdConcept method), 50  
 \_\_neg\_\_() (HasValueConcept method), 51  
 \_\_neg\_\_() (ImpliesConcept method), 52  
 \_\_neg\_\_() (LeftConcreteConcept method), 46  
 \_\_neg\_\_() (LinearConcreteConcept method), 47  
 \_\_neg\_\_() (LinearModifier method), 132  
 \_\_neg\_\_() (LinearlyModifiedConcept method), 55  
 \_\_neg\_\_() (ModifiedConcept method), 56  
 \_\_neg\_\_() (ModifiedConcreteConcept method), 47  
 \_\_neg\_\_() (NegatedNominal method), 57  
 \_\_neg\_\_() (OperatorConcept method), 58  
 \_\_neg\_\_() (OwaConcept method), 61  
 \_\_neg\_\_() (QowaConcept method), 62  
 \_\_neg\_\_() (QsugenoIntegral method), 62  
 \_\_neg\_\_() (RightConcreteConcept method), 48  
 \_\_neg\_\_() (SelfConcept method), 63  
 \_\_neg\_\_() (SigmaConcept method), 64  
 \_\_neg\_\_() (StringConcept method), 65  
 \_\_neg\_\_() (SugenoIntegral method), 66  
 \_\_neg\_\_() (Term method), 130  
 \_\_neg\_\_() (Thing method), 42  
 \_\_neg\_\_() (ThresholdConcept method), 67  
 \_\_neg\_\_() (TrapezoidalConcreteConcept method), 49  
 \_\_neg\_\_() (TriangularConcreteConcept method), 49  
 \_\_neg\_\_() (TriangularFuzzyNumber method), 45  
 \_\_neg\_\_() (TriangularModifier method), 134  
 \_\_neg\_\_() (TriangularlyModifiedConcept method), 56  
 \_\_neg\_\_() (TruthConcept method), 67  
 \_\_neg\_\_() (ValueConcept method), 68  
 \_\_neg\_\_() (WeightedConcept method), 69  
 \_\_neg\_\_() (WeightedMaxConcept method), 70  
 \_\_neg\_\_() (WeightedMinConcept method), 71  
 \_\_neg\_\_() (WeightedSumConcept method), 71  
 \_\_neg\_\_() (WeightedSumZeroConcept method), 72  
 \_\_radd\_\_() (Expression method), 121  
 \_\_repr\_\_() (Assertion method), 35  
 \_\_repr\_\_() (AtomicConcept method), 40  
 \_\_repr\_\_() (BlockingDynamicType method), 152  
 \_\_repr\_\_() (ClassificationNode method), 36  
 \_\_repr\_\_() (ConceptDefinition method), 183  
 \_\_repr\_\_() (ConceptType method), 153, 193  
 \_\_repr\_\_() (ConcreteFeature method), 73  
 \_\_repr\_\_() (ConcreteFeatureType method), 156  
 \_\_repr\_\_() (CreatedIndividualBlockingType method), 156  
 \_\_repr\_\_() (Degree method), 73  
 \_\_repr\_\_() (Expression method), 121  
 \_\_repr\_\_() (FeatureFunction method), 77  
 \_\_repr\_\_() (FeatureFunctionType method), 157  
 \_\_repr\_\_() (FuzzyDLKeyword method), 158  
 \_\_repr\_\_() (FuzzyDatatype method), 183  
 \_\_repr\_\_() (FuzzyLogic method), 162  
 \_\_repr\_\_() (FuzzyOWL2Keyword method), 194  
 \_\_repr\_\_() (GeneralConceptInclusion method), 80  
 \_\_repr\_\_() (Individual method), 82  
 \_\_repr\_\_() (InequalityType method), 162  
 \_\_repr\_\_() (Inequation method), 123  
 \_\_repr\_\_() (KnowledgeBaseRules method), 163  
 \_\_repr\_\_() (LogicOperatorType method), 165  
 \_\_repr\_\_() (ModifiedConcept method), 56  
 \_\_repr\_\_() (ModifiedProperty method), 186  
 \_\_repr\_\_() (Modifier method), 133  
 \_\_repr\_\_() (PrimitiveConceptDefinition method), 136  
 \_\_repr\_\_() (Relation method), 148  
 \_\_repr\_\_() (RepresentativeIndividualType method), 167  
 \_\_repr\_\_() (Restriction method), 149  
 \_\_repr\_\_() (RestrictionType method), 167  
 \_\_repr\_\_() (SigmaCount method), 64  
 \_\_repr\_\_() (Solution method), 129  
 \_\_repr\_\_() (Term method), 130  
 \_\_repr\_\_() (Thing method), 42  
 \_\_neg\_\_() (WeightedSumZeroConcept method), 72  
 \_\_or\_\_() (ApproximationConcept method), 39  
 \_\_or\_\_() (AtomicConcept method), 40  
 \_\_or\_\_() (ChoquetIntegral method), 41  
 \_\_or\_\_() (Concept method), 41  
 \_\_or\_\_() (CrispConcreteConcept method), 44  
 \_\_or\_\_() (ExtThresholdConcept method), 50  
 \_\_or\_\_() (HasValueConcept method), 51  
 \_\_or\_\_() (ImpliesConcept method), 52  
 \_\_or\_\_() (LeftConcreteConcept method), 46  
 \_\_or\_\_() (LinearConcreteConcept method), 47  
 \_\_or\_\_() (LinearModifier method), 132  
 \_\_or\_\_() (LinearlyModifiedConcept method), 55  
 \_\_or\_\_() (ModifiedConcept method), 56  
 \_\_or\_\_() (ModifiedConcreteConcept method), 47  
 \_\_or\_\_() (NegatedNominal method), 57  
 \_\_or\_\_() (OperatorConcept method), 58  
 \_\_or\_\_() (OwaConcept method), 61  
 \_\_or\_\_() (QowaConcept method), 62  
 \_\_or\_\_() (QsugenoIntegral method), 62  
 \_\_or\_\_() (RightConcreteConcept method), 48  
 \_\_or\_\_() (SelfConcept method), 63  
 \_\_or\_\_() (SigmaConcept method), 64  
 \_\_or\_\_() (SugenoIntegral method), 66  
 \_\_or\_\_() (ThresholdConcept method), 67  
 \_\_or\_\_() (TrapezoidalConcreteConcept method), 49  
 \_\_or\_\_() (TriangularConcreteConcept method), 49  
 \_\_or\_\_() (TriangularFuzzyNumber method), 45  
 \_\_or\_\_() (TriangularModifier method), 134  
 \_\_or\_\_() (TriangularlyModifiedConcept method), 56  
 \_\_or\_\_() (TruthConcept method), 68  
 \_\_or\_\_() (ValueConcept method), 68  
 \_\_or\_\_() (WeightedConcept method), 69  
 \_\_or\_\_() (WeightedMaxConcept method), 70  
 \_\_or\_\_() (WeightedMinConcept method), 71  
 \_\_or\_\_() (WeightedSumConcept method), 71  
 \_\_or\_\_() (WeightedSumZeroConcept method), 72  
 \_\_radd\_\_() (Expression method), 121  
 \_\_repr\_\_() (Assertion method), 35  
 \_\_repr\_\_() (AtomicConcept method), 40  
 \_\_repr\_\_() (BlockingDynamicType method), 152  
 \_\_repr\_\_() (ClassificationNode method), 36  
 \_\_repr\_\_() (ConceptDefinition method), 183  
 \_\_repr\_\_() (ConceptType method), 153, 193  
 \_\_repr\_\_() (ConcreteFeature method), 73  
 \_\_repr\_\_() (ConcreteFeatureType method), 156  
 \_\_repr\_\_() (CreatedIndividualBlockingType method), 156  
 \_\_repr\_\_() (Degree method), 73  
 \_\_repr\_\_() (Expression method), 121  
 \_\_repr\_\_() (FeatureFunction method), 77  
 \_\_repr\_\_() (FeatureFunctionType method), 157  
 \_\_repr\_\_() (FuzzyDLKeyword method), 158  
 \_\_repr\_\_() (FuzzyDatatype method), 183  
 \_\_repr\_\_() (FuzzyLogic method), 162  
 \_\_repr\_\_() (FuzzyOWL2Keyword method), 194  
 \_\_repr\_\_() (GeneralConceptInclusion method), 80  
 \_\_repr\_\_() (Individual method), 82  
 \_\_repr\_\_() (InequalityType method), 162  
 \_\_repr\_\_() (Inequation method), 123  
 \_\_repr\_\_() (KnowledgeBaseRules method), 163  
 \_\_repr\_\_() (LogicOperatorType method), 165  
 \_\_repr\_\_() (ModifiedConcept method), 56  
 \_\_repr\_\_() (ModifiedProperty method), 186  
 \_\_repr\_\_() (Modifier method), 133  
 \_\_repr\_\_() (PrimitiveConceptDefinition method), 136  
 \_\_repr\_\_() (Relation method), 148  
 \_\_repr\_\_() (RepresentativeIndividualType method), 167  
 \_\_repr\_\_() (Restriction method), 149  
 \_\_repr\_\_() (RestrictionType method), 167  
 \_\_repr\_\_() (SigmaCount method), 64  
 \_\_repr\_\_() (Solution method), 129  
 \_\_repr\_\_() (Term method), 130  
 \_\_repr\_\_() (Thing method), 42



\_\_repr\_\_ () (Variable method), 131  
 \_\_repr\_\_ () (VariableType method), 168  
 \_\_rmul\_\_ () (Expression method), 121  
 \_\_rmul\_\_ () (Term method), 130  
 \_\_rshift\_\_ () (AtomicConcept method), 40  
 \_\_rshift\_\_ () (Concept method), 41  
 \_\_rshift\_\_ () (TruthConcept method), 68  
 \_\_rsub\_\_ () (Expression method), 121  
 \_\_str\_\_ () (AllInstancesQuery method), 137  
 \_\_str\_\_ () (Assertion method), 35  
 \_\_str\_\_ () (AtomicAssertion method), 36  
 \_\_str\_\_ () (BlockingDynamicType method), 152  
 \_\_str\_\_ () (BnpQuery method), 137  
 \_\_str\_\_ () (ChoquetConcept method), 183  
 \_\_str\_\_ () (ClassificationNode method), 36  
 \_\_str\_\_ () (ClassificationQuery method), 138  
 \_\_str\_\_ () (Concept method), 42  
 \_\_str\_\_ () (ConceptType method), 153, 193  
 \_\_str\_\_ () (ConcreteFeature method), 73  
 \_\_str\_\_ () (ConcreteFeatureType method), 156  
 \_\_str\_\_ () (CreatedIndividual method), 81  
 \_\_str\_\_ () (CreatedIndividualBlockingType method), 157  
 \_\_str\_\_ () (Degree method), 73  
 \_\_str\_\_ () (DegreeExpression method), 74  
 \_\_str\_\_ () (DegreeNumeric method), 75  
 \_\_str\_\_ () (DegreeVariable method), 76  
 \_\_str\_\_ () (Expression method), 121  
 \_\_str\_\_ () (FeatureFunction method), 77  
 \_\_str\_\_ () (FeatureFunctionType method), 157  
 \_\_str\_\_ () (FuzzyDLKeyword method), 158  
 \_\_str\_\_ () (FuzzyLogic method), 162  
 \_\_str\_\_ () (FuzzyNominalConcept method), 184  
 \_\_str\_\_ () (FuzzyOWL2Keyword method), 194  
 \_\_str\_\_ () (GeneralConceptInclusion method), 80  
 \_\_str\_\_ () (Individual method), 82  
 \_\_str\_\_ () (InequalityType method), 162  
 \_\_str\_\_ () (Inequation method), 123  
 \_\_str\_\_ () (KbSatisfiableQuery method), 141  
 \_\_str\_\_ () (KnowledgeBaseRules method), 163  
 \_\_str\_\_ () (Label method), 120  
 \_\_str\_\_ () (LeftShoulderFunction method), 185  
 \_\_str\_\_ () (LinearFunction method), 185  
 \_\_str\_\_ () (LinearModifier method), 185  
 \_\_str\_\_ () (LogicOperatorType method), 166  
 \_\_str\_\_ () (LomDefuzzifyQuery method), 139  
 \_\_str\_\_ () (MaxInstanceQuery method), 141  
 \_\_str\_\_ () (MaxQuery method), 142  
 \_\_str\_\_ () (MaxRelatedQuery method), 142  
 \_\_str\_\_ () (MaxSatisfiableQuery method), 143  
 \_\_str\_\_ () (MaxSubsumesQuery method), 143  
 \_\_str\_\_ () (MinInstanceQuery method), 144  
 \_\_str\_\_ () (MinQuery method), 144  
 \_\_str\_\_ () (MinRelatedQuery method), 145  
 \_\_str\_\_ () (MinSatisfiableQuery method), 145  
 \_\_str\_\_ () (MinSubsumesQuery method), 146  
 \_\_str\_\_ () (ModifiedConcept method), 56, 186  
 \_\_str\_\_ () (ModifiedFunction method), 186  
 \_\_str\_\_ () (ModifiedProperty method), 186  
 \_\_str\_\_ () (Modifier method), 133  
 \_\_str\_\_ () (MomDefuzzifyQuery method), 139  
 \_\_str\_\_ () (OwaConcept method), 187  
 \_\_str\_\_ () (PrimitiveConceptDefinition method), 136  
 \_\_str\_\_ () (QowaConcept method), 187  
 \_\_str\_\_ () (QsugenoConcept method), 188  
 \_\_str\_\_ () (Query method), 146  
 \_\_str\_\_ () (Relation method), 148  
 \_\_str\_\_ () (RepresentativeIndividualType method), 167  
 \_\_str\_\_ () (Restriction method), 149  
 \_\_str\_\_ () (RestrictionType method), 167  
 \_\_str\_\_ () (RightShoulderFunction method), 188  
 \_\_str\_\_ () (SigmaCount method), 65  
 \_\_str\_\_ () (Solution method), 129  
 \_\_str\_\_ () (SomDefuzzifyQuery method), 140

\_\_str\_\_ () (SugenoConcept method), 188  
 \_\_str\_\_ () (Term method), 130  
 \_\_str\_\_ () (TrapezoidalFunction method), 189  
 \_\_str\_\_ () (TriangularFunction method), 189  
 \_\_str\_\_ () (TriangularModifier method), 190  
 \_\_str\_\_ () (Variable method), 131  
 \_\_str\_\_ () (VariableType method), 168  
 \_\_str\_\_ () (WeightedConcept method), 190  
 \_\_str\_\_ () (WeightedMaxConcept method), 190  
 \_\_str\_\_ () (WeightedMinConcept method), 191  
 \_\_str\_\_ () (WeightedSumConcept method), 191  
 \_\_str\_\_ () (WeightedSumZeroConcept method), 191  
 \_\_sub\_\_ () (Expression method), 121  
 \_\_sub\_\_ () (Term method), 130  
 \_\_sub\_\_ () (TriangularFuzzyNumber method), 45  
 \_\_truediv\_\_ () (Expression method), 121  
 \_\_truediv\_\_ () (Term method), 130  
 \_\_truediv\_\_ () (TriangularFuzzyNumber method), 45

## A

a (CrispConcreteConcept property), 44  
 a (DefuzzifyQuery attribute), 138  
 A (FuzzyOWL2Keyword attribute), 194  
 a (LeftConcreteConcept property), 46  
 a (LinearConcreteConcept property), 47  
 a (LinearModifier property), 133  
 a (RightConcreteConcept property), 48  
 a (TrapezoidalConcreteConcept property), 49  
 a (TriangularConcreteConcept property), 50  
 a (TriangularModifier property), 134  
 ABOX\_EXPANDED (KnowledgeBase attribute), 113  
 ABSORPTION\_OPERATORS (OperatorConcept attribute), 60  
 abstract\_fillers (ShowVariablesHelper attribute), 129  
 abstract\_roles (KnowledgeBase attribute), 114  
 acyclic\_tbox (KnowledgeBase attribute), 114  
 add () (TriangularFuzzyNumber static method), 45  
 add\_abstract\_filler\_to\_show () (ShowVariablesHelper method), 128  
 add\_assertion () (KnowledgeBase method), 91  
 add\_assertions () (KnowledgeBase method), 91  
 add\_atomic\_concepts\_disjoint () (KnowledgeBase method), 91  
 add\_axiom\_to\_A\_equiv\_C () (KnowledgeBase method), 91  
 add\_axiom\_to\_A\_is\_a\_C () (KnowledgeBase method), 91  
 add\_axiom\_to\_C\_is\_a\_A () (KnowledgeBase method), 91  
 add\_axiom\_to\_C\_is\_a\_D () (KnowledgeBase method), 92  
 add\_axiom\_to\_C\_is\_a\_X () (KnowledgeBase method), 92  
 add\_axiom\_to\_do\_A\_is\_a\_X () (KnowledgeBase method), 92  
 add\_axiom\_to\_inc () (KnowledgeBase method), 92  
 add\_axioms\_to\_tg () (KnowledgeBase method), 92  
 add\_cardinality\_list () (MILPHelper method), 123  
 add\_concept () (Individual method), 82  
 add\_concept () (KnowledgeBase method), 92  
 add\_concept\_to\_show () (ShowVariablesHelper method), 128  
 add\_concepts\_disjoint () (KnowledgeBase method), 92  
 add\_concrete\_filler\_to\_show () (ShowVariablesHelper method), 128  
 add\_concrete\_restriction () (Individual method), 82  
 add\_constant () (Expression static method), 121  
 add\_contradiction () (MILPHelper method), 123  
 add\_created\_individual () (KnowledgeBase method), 93  
 add\_crisp\_concept () (MILPHelper method), 123  
 add\_crisp\_role () (MILPHelper method), 123  
 add\_datatype\_restriction () (KnowledgeBase method), 93  
 add\_disjoint\_union\_concept () (KnowledgeBase method), 93  
 add\_entity\_annotation () (FuzzydlToOwl2 method), 78  
 add\_equality () (MILPHelper method), 124  
 add\_equivalence\_relation () (KnowledgeBase method), 93  
 add\_equivalent\_concepts () (KnowledgeBase method), 93  
 add\_equivalent\_roles () (KnowledgeBase method), 93  
 add\_expressions () (Expression static method), 121  
 add\_fuzzy\_number () (KnowledgeBase method), 93  
 add\_gci () (KnowledgeBase method), 93  
 add\_individual () (KnowledgeBase method), 94



add\_individual\_to\_concept() (*KnowledgeBase* method), 94  
 add\_individual\_to\_show() (*ShowVariablesHelper* method), 128  
 add\_input\_edge() (*ClassificationNode* method), 36  
 add\_inverse\_roles() (*KnowledgeBase* method), 94  
 add\_label() (*ClassificationNode* method), 36  
 add\_labels\_with\_nodes() (*KnowledgeBase* method), 94  
 add\_modifier() (*KnowledgeBase* method), 94  
 add\_mutually\_disjoint() (*KnowledgeBase* method), 94  
 add\_negated\_datatype\_restriction() (*KnowledgeBase* method), 94  
 add\_negated\_equations() (*KnowledgeBase* method), 94  
 add\_new\_constraint() (*MILPHelper* method), 124  
 add\_not\_self\_restriction() (*IndividualHandler* static method), 90  
 add\_ontology\_annotation() (*FuzzyclToOwl2* method), 78  
 add\_output\_edge() (*ClassificationNode* method), 36  
 add\_parent\_recursively() (*KnowledgeBase* method), 94  
 add\_relation() (*IndividualHandler* static method), 90  
 add\_relation() (*KnowledgeBase* method), 94  
 add\_relation\_with\_role\_parent() (*KnowledgeBase* method), 95  
 add\_relation\_with\_role\_parent\_in\_lukasiewicz() (*KnowledgeBase* method), 95  
 add\_restriction() (*IndividualHandler* static method), 90  
 add\_showed\_variable() (*Solution* method), 129  
 add\_similarity\_relation() (*KnowledgeBase* method), 95  
 add\_simple\_inverse\_roles() (*KnowledgeBase* method), 95  
 add\_string\_feature() (*MILPHelper* method), 124  
 add\_string\_value() (*MILPHelper* method), 124  
 add\_subsumption() (*KnowledgeBase* method), 95  
 add\_tdef\_links() (*KnowledgeBase* method), 95  
 add\_tdr\_links() (*KnowledgeBase* method), 95  
 add\_term() (*Expression* method), 121  
 add\_term\_() (*Expression* static method), 122  
 add\_tinc\_links() (*KnowledgeBase* method), 95  
 add\_tmp\_feature() (*KnowledgeBase* method), 95  
 add\_to\_expression() (*Degree* method), 74  
 add\_to\_expression() (*DegreeExpression* method), 74  
 add\_to\_expression() (*DegreeNumeric* method), 75  
 add\_to\_expression() (*DegreeVariable* method), 76  
 add\_to\_nominal\_list() (*Individual* method), 82  
 add\_variable() (*ShowVariablesHelper* method), 128  
 ALL (*ConceptType* attribute), 153  
 ALL (*FuzzyDLKeyword* attribute), 158  
 All (in module *fuzzy\_dl\_owl2.fuzzycl.concept.all\_some\_concept*), 38  
 all() (*AllSomeConcept* static method), 37  
 ALL\_INSTANCES\_QUERY (*FuzzyDLKeyword* attribute), 158  
 ALL\_OPERATORS (*OperatorConcept* attribute), 60  
 AllInstancesQuery (class in *fuzzy\_dl\_owl2.fuzzycl.query.all\_instances\_query*), 137  
 AllSomeConcept (class in *fuzzy\_dl\_owl2.fuzzycl.concept.all\_some\_concept*), 37  
 AND (*ConceptType* attribute), 153  
 AND (*FuzzyDLKeyword* attribute), 158  
 And (in module *fuzzy\_dl\_owl2.fuzzycl.concept.operator\_concept*), 60  
 and\_() (*LukasiewiczSolver* static method), 116  
 and\_() (*OperatorConcept* static method), 58  
 and\_() (*ZadehSolver* static method), 118  
 and\_equation() (*LukasiewiczSolver* static method), 117  
 and\_equation() (*ZadehSolver* static method), 118  
 and\_geq\_equation() (*LukasiewiczSolver* static method), 117  
 and\_geq\_equation() (*ZadehSolver* static method), 118  
 and\_leq\_equation() (*LukasiewiczSolver* static method), 117  
 and\_leq\_equation() (*ZadehSolver* static method), 118  
 and\_negated\_equation() (*ZadehSolver* static method), 119  
 AND\_OPERATORS (*OperatorConcept* attribute), 60  
 annotate\_gci() (*FuzzyclToOwl2* method), 78  
 annotate\_pcd() (*FuzzyclToOwl2* method), 78  
 annotation\_property\_iri() (*FuzzyclToOwl2* method), 78  
 ANYWHERE\_DOUBLE\_BLOCKING (*BlockingDynamicType* attribute), 152  
 ANYWHERE\_DOUBLE\_BLOCKING (*ConfigReader* attribute), 150  
 ANYWHERE\_SET\_BLOCKING (*BlockingDynamicType* attribute), 152  
 ANYWHERE\_SIMPLE\_BLOCKING (*ConfigReader* attribute), 150  
 ANYWHERE\_SUBSET\_BLOCKING (*BlockingDynamicType* attribute), 152  
 applied\_trans\_role\_rules (*KnowledgeBase* attribute), 114  
 apply\_at\_least\_value\_rule() (*DatatypeReasoner* static method), 87  
 apply\_at\_most\_value\_rule() (*DatatypeReasoner* static method), 87  
 apply\_exact\_value\_rule() (*DatatypeReasoner* static method), 87  
 apply\_not\_at\_least\_value\_rule() (*DatatypeReasoner* static method), 87  
 apply\_not\_at\_most\_value\_rule() (*DatatypeReasoner* static method), 87  
 apply\_not\_exact\_value\_rule() (*DatatypeReasoner* static method), 88  
 apply\_not\_rule() (*DatatypeReasoner* static method), 88  
 apply\_rule() (*DatatypeReasoner* static method), 88  
 ApproximationConcept (class in *fuzzy\_dl\_owl2.fuzzycl.concept.approximation\_concept*), 38  
 Assertion (class in *fuzzy\_dl\_owl2.fuzzycl.assertion.assertion*), 35  
 assertions (*KnowledgeBase* attribute), 114  
 AT\_LEAST\_VALUE (*ConceptType* attribute), 153  
 AT\_LEAST\_VALUE (*RestrictionType* attribute), 168  
 at\_least\_value() (*ValueConcept* static method), 68  
 AT\_MOST\_VALUE (*ConceptType* attribute), 153  
 AT\_MOST\_VALUE (*RestrictionType* attribute), 168  
 at\_most\_value() (*ValueConcept* static method), 68  
 ATOMIC (*ConceptType* attribute), 153  
 ATOMIC (*FeatureFunctionType* attribute), 157  
 atomic\_concepts (*KnowledgeBase* attribute), 114  
 AtomicAssertion (class in *fuzzy\_dl\_owl2.fuzzycl.assertion.atomic\_assertion*), 36  
 AtomicConcept (class in *fuzzy\_dl\_owl2.fuzzycl.concept.atomic\_concept*), 40  
 AXIOM (*FuzzyOWL2Keyword* attribute), 194  
 axioms\_A\_equiv\_C (*KnowledgeBase* attribute), 114  
 axioms\_A\_is\_a\_B (*KnowledgeBase* attribute), 114  
 axioms\_A\_is\_a\_C (*KnowledgeBase* attribute), 114  
 axioms\_C\_equiv\_D (*KnowledgeBase* attribute), 114  
 axioms\_C\_is\_a\_A (*KnowledgeBase* attribute), 114  
 axioms\_C\_is\_a\_D (*KnowledgeBase* attribute), 114  
 axioms\_to\_do\_A\_is\_a\_B (*KnowledgeBase* attribute), 114  
 axioms\_to\_do\_A\_is\_a\_C (*KnowledgeBase* attribute), 114  
 axioms\_to\_do\_C\_is\_a\_A (*KnowledgeBase* attribute), 114  
 axioms\_to\_do\_C\_is\_a\_D (*KnowledgeBase* attribute), 114  
 axioms\_to\_do\_tmp\_A\_is\_a\_C (*KnowledgeBase* attribute), 114  
 axioms\_to\_do\_tmp\_C\_is\_a\_A (*KnowledgeBase* attribute), 114  
 axioms\_to\_do\_tmp\_C\_is\_a\_D (*KnowledgeBase* attribute), 114

## B

b (*CrispConcreteConcept* property), 44  
 B (*FuzzyOWL2Keyword* attribute), 194  
 b (*LeftConcreteConcept* property), 46  
 b (*LinearConcreteConcept* property), 47  
 b (*LinearModifier* property), 133  
 b (*RightConcreteConcept* property), 48  
 b (*TrapezoidalConcreteConcept* property), 49  
 b (*TriangularConcreteConcept* property), 50  
 b (*TriangularModifier* property), 134  
 BASE (*FuzzyOWL2Keyword* attribute), 194  
 BINARY (*FuzzyDLKeyword* attribute), 158  
 BINARY (*VariableType* attribute), 168  
 BINARY\_OPERATORS (*OperatorConcept* attribute), 60  
 BinaryVar (in module *fuzzy\_dl\_owl2.fuzzycl.milp.variable*), 132  
 BLOCKED (*CreatedIndividualBlockingType* attribute), 157  
 blocked\_assertions (*KnowledgeBase* attribute), 114  
 blocked\_exist\_assertions (*KnowledgeBase* attribute), 114  
 blocking\_dynamic (*KnowledgeBase* attribute), 114  
 blocking\_type (*KnowledgeBase* attribute), 114  
 BlockingDynamicType (class in *fuzzy\_dl\_owl2.fuzzycl.util.constants*), 151

BNP\_QUERY (FuzzyDLKeyword attribute), 159  
 BnpQuery (class in fuzzy\_dl\_owl2.fuzzycl.query.bnp\_query), 137  
 BOOLEAN (ConcreteFeatureType attribute), 156  
 BOOLEAN (FuzzyDLKeyword attribute), 159  
 boolean\_datatypes (FuzzyOwl2ToFuzzyDL attribute), 182  
 BOTTOM (ConceptType attribute), 153  
 BOTTOM (FuzzyDLKeyword attribute), 159  
 BOTTOM (in module fuzzy\_dl\_owl2.fuzzycl.concept.truth\_concept), 68  
 build\_concept\_xml() (FuzzyXML static method), 196  
 build\_datatype\_xml() (FuzzyXML static method), 196  
 build\_degree\_xml() (FuzzyXML static method), 196  
 build\_logic\_xml() (FuzzyXML static method), 196  
 build\_main\_xml() (FuzzyXML static method), 196  
 build\_modifier\_xml() (FuzzyXML static method), 196  
 build\_names\_xml() (FuzzyXML static method), 196  
 build\_weights\_xml() (FuzzyXML static method), 196

## C

c (AtomicAssertion attribute), 36  
 c (BnpQuery attribute), 137  
 C (FuzzyOWL2Keyword attribute), 194  
 c (LinearModifier property), 133  
 c (TrapezoidalConcreteConcept property), 49  
 c (TriangularConcreteConcept property), 50  
 c (TriangularModifier property), 134  
 c1 (ConceptEquivalence attribute), 72  
 c1 (SubsumptionQuery attribute), 147  
 c2 (ConceptEquivalence attribute), 72  
 c2 (SubsumptionQuery attribute), 147  
 cardinalities (MILPHelper attribute), 127  
 change\_variable\_names() (MILPHelper method), 124  
 check\_fuzzy\_number\_concept\_exists() (KnowledgeBase method), 95  
 check\_if\_replacement\_is\_needed() (MILPHelper method), 124  
 check\_individual\_exists() (KnowledgeBase method), 95  
 check\_role() (KnowledgeBase method), 95  
 check\_trans\_role\_applied() (KnowledgeBase method), 96  
 CHOQUET (ConceptType attribute), 193  
 CHOQUET (FuzzyDLKeyword attribute), 159  
 CHOQUET (FuzzyOWL2Keyword attribute), 194  
 CHOQUET\_INTEGRAL (ConceptType attribute), 153  
 ChoquetConcept (class in  
   fuzzy\_dl\_owl2.fuzzycl.owl\_types.choquet\_concept),  
   183  
 ChoquetIntegral (class in  
   fuzzy\_dl\_owl2.fuzzycl.concept.choquet\_integral), 41  
 class\_debugging() (in module fuzzy\_dl\_owl2.fuzzycl.util.utils),  
   169  
 class\_iri() (FuzzyclToOwl2 method), 78  
 classic\_cnf() (Thing method), 42  
 classic\_dnf() (Thing method), 42  
 CLASSICAL (FuzzyDLKeyword attribute), 159  
 CLASSICAL (FuzzyLogic attribute), 162  
 ClassicalSolver (class in  
   fuzzy\_dl\_owl2.fuzzycl.knowledge\_base), 84  
 ClassificationNode (class in  
   fuzzy\_dl\_owl2.fuzzycl.classification\_node), 36  
 ClassificationQuery (class in  
   fuzzy\_dl\_owl2.fuzzycl.query.classification\_query), 138  
 CLASSIFIED (KnowledgeBase attribute), 113  
 classify() (KnowledgeBase method), 96  
 clone() (AllSomeConcept method), 38  
 clone() (ApproximationConcept method), 39  
 clone() (Assertion method), 35  
 clone() (AtomicConcept method), 40  
 clone() (ChoquetIntegral method), 41  
 clone() (ConceptEquivalence method), 72  
 clone() (ConcreteFeature method), 73  
 clone() (CreatedIndividual method), 81  
 clone() (CrispConcreteConcept method), 44  
 clone() (Degree method), 74  
 clone() (DegreeExpression method), 74

clone() (DegreeNumeric method), 75  
 clone() (DegreeVariable method), 76  
 clone() (Expression method), 122  
 clone() (ExtThresholdConcept method), 50  
 clone() (GeneralConceptInclusion method), 80  
 clone() (HasValueConcept method), 51  
 clone() (ImpliesConcept method), 52  
 clone() (Individual method), 83  
 clone() (Inequation method), 123  
 clone() (KnowledgeBase method), 96  
 clone() (LeftConcreteConcept method), 46  
 clone() (LinearConcreteConcept method), 47  
 clone() (LinearlyModifiedConcept method), 55  
 clone() (LinearModifier method), 132  
 clone() (MILPHelper method), 124  
 clone() (ModifiedConcreteConcept method), 47  
 clone() (Modifier method), 133  
 clone() (NegatedNominal method), 57  
 clone() (OperatorConcept method), 58  
 clone() (OwaConcept method), 61  
 clone() (PrimitiveConceptDefinition method), 136  
 clone() (QowaConcept method), 62  
 clone() (QsugenIntegral method), 62  
 clone() (Relation method), 148  
 clone() (Restriction method), 149  
 clone() (RightConcreteConcept method), 48  
 clone() (SelfConcept method), 63  
 clone() (ShowVariablesHelper method), 128  
 clone() (SigmaConcept method), 64  
 clone() (SigmaCount method), 65  
 clone() (StringConcept method), 65  
 clone() (SugenIntegral method), 66  
 clone() (Term method), 130  
 clone() (Thing method), 42  
 clone() (ThresholdConcept method), 67  
 clone() (TrapezoidalConcreteConcept method), 49  
 clone() (TriangularConcreteConcept method), 49  
 clone() (TriangularFuzzyNumber method), 45  
 clone() (TriangularlyModifiedConcept method), 56  
 clone() (TriangularModifier method), 134  
 clone() (TruthConcept method), 68  
 clone() (ValueConcept method), 69  
 clone() (Variable method), 131  
 clone() (WeightedConcept method), 69  
 clone() (WeightedMaxConcept method), 70  
 clone() (WeightedMinConcept method), 71  
 clone() (WeightedSumConcept method), 71  
 clone() (WeightedSumZeroConcept method), 72  
 clone\_attributes() (Individual method), 83  
 clone\_special\_attributes() (CreatedIndividual method), 81  
 clone\_withoutabox() (KnowledgeBase method), 96  
 CLOSE\_TAG (FuzzyOWL2Keyword attribute), 194  
 common\_part\_add\_restriction() (IndividualHandler static  
   method), 90  
 COMPLEMENT (ConceptType attribute), 153  
 COMPLEMENT\_LAW\_OPERATORS (OperatorConcept attribute), 60  
 compute\_atomic\_concepts() (AllSomeConcept method), 38  
 compute\_atomic\_concepts() (ApproximationConcept method),  
   39  
 compute\_atomic\_concepts() (AtomicConcept method), 40  
 compute\_atomic\_concepts() (ChoquetIntegral method), 41  
 compute\_atomic\_concepts() (ExtThresholdConcept method), 50  
 compute\_atomic\_concepts() (FuzzyConcreteConcept method),  
   44  
 compute\_atomic\_concepts() (HasValueConcept method), 51  
 compute\_atomic\_concepts() (ImpliesConcept method), 52  
 compute\_atomic\_concepts() (ModifiedConcept method), 56  
 compute\_atomic\_concepts() (OperatorConcept method), 58  
 compute\_atomic\_concepts() (OwaConcept method), 61  
 compute\_atomic\_concepts() (SelfConcept method), 63  
 compute\_atomic\_concepts() (SigmaConcept method), 64  
 compute\_atomic\_concepts() (StringConcept method), 65  
 compute\_atomic\_concepts() (SugenIntegral method), 66

- `compute_atomic_concepts()` (*Thing* method), 42
- `compute_atomic_concepts()` (*ThresholdConcept* method), 67
- `compute_atomic_concepts()` (*TruthConcept* method), 68
- `compute_atomic_concepts()` (*ValueConcept* method), 69
- `compute_atomic_concepts()` (*WeightedConcept* method), 69
- `compute_atomic_concepts()` (*WeightedMaxConcept* method), 70
- `compute_atomic_concepts()` (*WeightedMinConcept* method), 71
- `compute_atomic_concepts()` (*WeightedSumConcept* method), 71
- `compute_atomic_concepts()` (*WeightedSumZeroConcept* method), 72
- `compute_blocking_type()` (*KnowledgeBase* method), 96
- `compute_language()` (*KnowledgeBase* method), 96
- `compute_name()` (*AllSomeConcept* method), 38
- `compute_name()` (*ApproximationConcept* method), 39
- `compute_name()` (*AtomicConcept* method), 40
- `compute_name()` (*ChoquetIntegral* method), 41
- `compute_name()` (*CrispConcreteConcept* method), 44
- `compute_name()` (*ExtThresholdConcept* method), 50
- `compute_name()` (*FuzzyConcreteConcept* method), 44
- `compute_name()` (*HasValueConcept* method), 51
- `compute_name()` (*ImpliesConcept* method), 52
- `compute_name()` (*LeftConcreteConcept* method), 46
- `compute_name()` (*LinearConcreteConcept* method), 47
- `compute_name()` (*LinearModifier* method), 132
- `compute_name()` (*ModifiedConcept* method), 56
- `compute_name()` (*ModifiedConcreteConcept* method), 47
- `compute_name()` (*Modifier* method), 133
- `compute_name()` (*NegatedNominal* method), 57
- `compute_name()` (*OperatorConcept* method), 58
- `compute_name()` (*OwaConcept* method), 61
- `compute_name()` (*QowaConcept* method), 62
- `compute_name()` (*QsugenoIntegral* method), 62
- `compute_name()` (*RightConcreteConcept* method), 48
- `compute_name()` (*SelfConcept* method), 63
- `compute_name()` (*SigmaConcept* method), 64
- `compute_name()` (*StringConcept* method), 65
- `compute_name()` (*SugenoIntegral* method), 66
- `compute_name()` (*Thing* method), 42
- `compute_name()` (*ThresholdConcept* method), 67
- `compute_name()` (*TrapezoidalConcreteConcept* method), 49
- `compute_name()` (*TriangularConcreteConcept* method), 49
- `compute_name()` (*TriangularFuzzyNumber* method), 45
- `compute_name()` (*TriangularModifier* method), 134
- `compute_name()` (*TruthConcept* method), 68
- `compute_name()` (*ValueConcept* method), 69
- `compute_name()` (*WeightedConcept* method), 69
- `compute_name()` (*WeightedMaxConcept* method), 70
- `compute_name()` (*WeightedMinConcept* method), 71
- `compute_name()` (*WeightedSumConcept* method), 71
- `compute_name()` (*WeightedSumZeroConcept* method), 72
- `compute_variables_old_calculus()` (*KnowledgeBase* method), 96
- `compute_weights()` (*QowaConcept* method), 62
- `conc` (*AllInstancesQuery* attribute), 137
- `conc` (*DefuzzifyQuery* attribute), 138
- `conc` (*InstanceQuery* attribute), 140
- `concept` (*Assertion* attribute), 36
- `Concept` (class in *fuzzy\_dl\_owl2.fuzzycl.concept.concept*), 41
- `concept` (*DomainAxiom* attribute), 76
- `CONCEPT` (*FuzzyOWL2Keyword* attribute), 194
- `concept` (*Label* attribute), 120
- `concept` (*RangeAxiom* attribute), 148
- `concept` (*Restriction* attribute), 150
- `concept` (*SigmaConcept* attribute), 64
- `concept` (*SigmaCount* attribute), 65
- `concept_absorption()` (*KnowledgeBase* method), 96
- `concept_exists()` (*KnowledgeBase* method), 96
- `concept_individual_list` (*KnowledgeBase* attribute), 114
- `CONCEPT_NAMES` (*FuzzyOWL2Keyword* attribute), 194
- `ConceptDefinition` (class in *fuzzy\_dl\_owl2.fuzzycl.owl\_types.concept\_definition*), 183
- `ConceptEquivalence` (class in *fuzzy\_dl\_owl2.fuzzycl.concept\_equivalence*), 72
- `concepts` (*FuzzyclToOwl2* attribute), 79
- `concepts` (*HasConceptsInterface* property), 53
- `concepts` (*OperatorConcept* property), 60
- `concepts` (*ShowVariablesHelper* attribute), 129
- `ConceptType` (class in *fuzzy\_dl\_owl2.fuzzycl.util.constants*), 152
- `ConceptType` (class in *fuzzy\_dl\_owl2.fuzzyclowl2.util.constants*), 193
- `CONCRETE` (*ConceptType* attribute), 153
- `CONCRETE_COMPLEMENT` (*ConceptType* attribute), 153
- `concrete_concept` (*SigmaConcept* attribute), 64
- `concrete_concepts` (*KnowledgeBase* attribute), 114
- `concrete_features` (*KnowledgeBase* attribute), 115
- `concrete_fillers` (*ShowVariablesHelper* attribute), 129
- `concrete_fuzzy_concepts` (*KnowledgeBase* attribute), 115
- `concrete_role_restrictions` (*Individual* attribute), 83
- `concrete_roles` (*KnowledgeBase* attribute), 115
- `ConcreteFeature` (class in *fuzzy\_dl\_owl2.fuzzycl.concrete\_feature*), 73
- `ConcreteFeatureType` (class in *fuzzy\_dl\_owl2.fuzzycl.util.constants*), 155
- `ConfigReader` (class in *fuzzy\_dl\_owl2.fuzzycl.util.config\_reader*), 150
- `CONSISTENT_KB` (*Solution* attribute), 130
- `CONSTRAINTS` (*FuzzyDLKeyword* attribute), 159
- `constraints` (*MILPHelper* attribute), 127
- `contains_negated_subconcept()` (*Thing* static method), 42
- `contains_subconcept()` (*Thing* static method), 42
- `CONTINUOUS` (*VariableType* attribute), 168
- `convert_strings_into_integers()` (*KnowledgeBase* method), 96
- `create_inequality_with_degree_rhs()` (*Degree* method), 74
- `create_inequality_with_degree_rhs()` (*DegreeExpression* method), 74
- `create_inequality_with_degree_rhs()` (*DegreeNumeric* method), 75
- `create_inequality_with_degree_rhs()` (*DegreeVariable* method), 76
- `create_roles_with_all_parents()` (*KnowledgeBase* method), 96
- `create_roles_with_trans_children()` (*KnowledgeBase* method), 96
- `CreatedIndividual` (class in *fuzzy\_dl\_owl2.fuzzycl.individual.created\_individual*), 81
- `CreatedIndividualBlockingType` (class in *fuzzy\_dl\_owl2.fuzzycl.util.constants*), 156
- `CreatedIndividualHandler` (class in *fuzzy\_dl\_owl2.fuzzycl.knowledge\_base*), 85
- `CRISP` (*FuzzyDLKeyword* attribute), 159
- `CRISP` (*FuzzyOWL2Keyword* attribute), 194
- `CRISP_CONCEPT` (*FuzzyDLKeyword* attribute), 159
- `crisp_concepts` (*MILPHelper* attribute), 127
- `CRISP_ROLE` (*FuzzyDLKeyword* attribute), 159
- `crisp_roles` (*MILPHelper* attribute), 127
- `CrispConcreteConcept` (class in *fuzzy\_dl\_owl2.fuzzycl.concept.concrete.crisp\_concrete\_concept*), 44
- `curr_concept` (*HasConceptInterface* property), 53

## D

- `D` (*FuzzyOWL2Keyword* attribute), 194
- `d` (*TrapezoidalConcreteConcept* property), 49
- `data_properties` (*FuzzyOwl2ToFuzzyDL* attribute), 182
- `data_property_iri()` (*FuzzyclToOwl2* method), 78
- `DATATYPE` (*FuzzyOWL2Keyword* attribute), 195
- `datatype_filler` (*Variable* attribute), 131
- `datatype_iri()` (*FuzzyclToOwl2* method), 78
- `DatatypeReasoner` (class in *fuzzy\_dl\_owl2.fuzzycl.knowledge\_base*), 87
- `datatypes` (*FuzzyclToOwl2* attribute), 79
- `de_morgan()` (*OperatorConcept* method), 58
- `de_morgan()` (*Thing* method), 42
- `debug()` (*Util* static method), 169



DEBUG\_PRINT (*ConfigReader* attribute), 150  
 debugging\_wrapper() (in module *fuzzy\_dl\_owl2.fuzzycl.util.utils*), 169  
 DEFAULT\_NAME (*Concept* attribute), 42  
 DEFAULT\_NAME (*Individual* attribute), 83  
 define\_atomic\_concept() (*KnowledgeBase* method), 96  
 define\_boolean\_concrete\_feature() (*KnowledgeBase* method), 96  
 DEFINE\_CONCEPT (*FuzzyDLKeyword* attribute), 159  
 define\_concept() (*KnowledgeBase* method), 97  
 define\_concreate\_feature() (*KnowledgeBase* method), 97  
 define\_equivalent\_concepts() (*KnowledgeBase* method), 97  
 DEFINE\_FUZZY\_CONCEPT (*FuzzyDLKeyword* attribute), 159  
 DEFINE\_FUZZY\_EQUIVALENCE (*FuzzyDLKeyword* attribute), 159  
 DEFINE\_FUZZY\_LOGIC (*FuzzyDLKeyword* attribute), 159  
 DEFINE\_FUZZY\_NUMBER (*FuzzyDLKeyword* attribute), 159  
 DEFINE\_FUZZY\_NUMBER\_RANGE (*FuzzyDLKeyword* attribute), 159  
 DEFINE\_FUZZY\_SIMILARITY (*FuzzyDLKeyword* attribute), 159  
 define\_integer\_concrete\_feature() (*KnowledgeBase* method), 97  
 DEFINE\_MODIFIER (*FuzzyDLKeyword* attribute), 159  
 DEFINE\_PRIMITIVE\_CONCEPT (*FuzzyDLKeyword* attribute), 159  
 define\_real\_concrete\_feature() (*KnowledgeBase* method), 97  
 define\_string\_concrete\_feature() (*KnowledgeBase* method), 97  
 define\_synonym() (*KnowledgeBase* method), 97  
 define\_synonyms() (*KnowledgeBase* method), 97  
 DEFINE\_TRUTH\_CONSTANT (*FuzzyDLKeyword* attribute), 159  
 defined (*PrimitiveConceptDefinition* attribute), 136  
 defined\_concepts (*FuzzyOwl2* attribute), 176  
 defined\_properties (*FuzzyOwl2* attribute), 176  
 definition (*PrimitiveConceptDefinition* attribute), 136  
 definition\_absorption() (*KnowledgeBase* method), 97  
 definition\_absorption\_to\_do() (*KnowledgeBase* method), 97  
 DEFUZZIFY\_LOM\_QUERY (*FuzzyDLKeyword* attribute), 159  
 DEFUZZIFY\_MOM\_QUERY (*FuzzyDLKeyword* attribute), 159  
 DEFUZZIFY\_SOM\_QUERY (*FuzzyDLKeyword* attribute), 159  
 DefuzzifyQuery (class in *fuzzy\_dl\_owl2.fuzzycl.query.defuzzify.defuzzify\_query*), 138  
 degree (*Assertion* attribute), 36  
 degree (*AtomicAssertion* attribute), 36  
 Degree (class in *fuzzy\_dl\_owl2.fuzzycl.degree.degree*), 73  
 degree (*GeneralConceptInclusion* attribute), 80  
 degree (*PrimitiveConceptDefinition* attribute), 136  
 degree (*Relation* attribute), 148  
 degree (*Restriction* attribute), 150  
 degree (*RoleParentWithDegree* attribute), 150  
 DEGREE\_DEF (*FuzzyOWL2Keyword* attribute), 195  
 degree\_if\_not\_one() (*KnowledgeBase* method), 98  
 DEGREE\_VALUE (*FuzzyOWL2Keyword* attribute), 195  
 DegreeExpression (class in *fuzzy\_dl\_owl2.fuzzycl.degree.degree\_expression*), 74  
 DegreeNumeric (class in *fuzzy\_dl\_owl2.fuzzycl.degree.degree\_numeric*), 75  
 degrees (*AllInstancesQuery* attribute), 137  
 DegreeVariable (class in *fuzzy\_dl\_owl2.fuzzycl.degree.degree\_variable*), 76  
 directly\_blocked\_children (*KnowledgeBase* attribute), 115  
 DISJOINT (*FuzzyDLKeyword* attribute), 159  
 DISJOINT\_UNION (*FuzzyDLKeyword* attribute), 159  
 disjoint\_variables (*KnowledgeBase* attribute), 115  
 disjoint\_with\_defined\_concept() (*KnowledgeBase* method), 98  
 distribute() (*OperatorConcept* method), 58  
 distribute() (*Thing* method), 42  
 DISTRIBUTIVE\_OPERATORS (*OperatorConcept* attribute), 60  
 divided\_by() (*TriangularFuzzyNumber* static method), 45  
 DLParser (class in *fuzzy\_dl\_owl2.fuzzycl.parser.dl\_parser*), 135  
 DOMAIN (*FuzzyDLKeyword* attribute), 159  
 domain\_restrictions (*KnowledgeBase* attribute), 115  
 DomainAxiom (class in *fuzzy\_dl\_owl2.fuzzycl.domain\_axiom*), 76  
 DOUBLE\_BLOCKING (*BlockingDynamicType* attribute), 152

DOUBLE\_MAX\_VALUE (*FuzzyOwl2ToFuzzyDL* attribute), 182  
 DOUBLE\_MIN\_VALUE (*FuzzyOwl2ToFuzzyDL* attribute), 182

## E

EPSILON (*ConfigReader* attribute), 150  
 EPSILON (*FuzzyOwl2ToFuzzyDL* attribute), 182  
 EQUAL (*FuzzyOWL2Keyword* attribute), 195  
 EQUAL (*InequalityType* attribute), 162  
 equal\_to() (*Inequation* static method), 123  
 EQUALS (*FuzzyDLKeyword* attribute), 159  
 equals() (*Assertion* method), 35  
 EqualTo (in module *fuzzy\_dl\_owl2.fuzzycl.milp.inequation*), 123  
 EQUIVALENT\_CONCEPTS (*FuzzyDLKeyword* attribute), 159  
 EQUIVALENT\_NAMES (*ClassificationNode* attribute), 37  
 error() (*Util* static method), 169  
 EXACT\_VALUE (*ConceptType* attribute), 153  
 EXACT\_VALUE (*RestrictionType* attribute), 168  
 exact\_value() (*ValueConcept* static method), 69  
 exist\_assertions (*KnowledgeBase* attribute), 115  
 exist\_data\_property() (*FuzzyclToOwl2* method), 78  
 exist\_object\_property() (*FuzzyclToOwl2* method), 78  
 exists\_nominal\_variable() (*MILPHelper* method), 124  
 exists\_prime\_concept\_definition() (*KnowledgeBase* method), 98  
 exists\_variable() (*MILPHelper* method), 125  
 exit\_condition() (*KnowledgeBase* method), 98  
 exit\_condition\_A\_is\_a\_X() (*KnowledgeBase* method), 98  
 exit\_condition\_C\_is\_a\_X() (*KnowledgeBase* method), 98  
 expr (*DegreeExpression* attribute), 75  
 expr (*Inequation* attribute), 123  
 Expression (class in *fuzzy\_dl\_owl2.fuzzycl.milp.expression*), 121  
 EXT\_NEG\_THRESHOLD (*ConceptType* attribute), 153  
 EXT\_POS\_THRESHOLD (*ConceptType* attribute), 153  
 extended\_neg\_threshold() (*ExtThresholdConcept* static method), 50  
 extended\_pos\_threshold() (*ExtThresholdConcept* static method), 50  
 ExtendedNegThreshold (in module *fuzzy\_dl\_owl2.fuzzycl.concept.ext\_threshold\_concept*), 51  
 ExtendedPosThreshold (in module *fuzzy\_dl\_owl2.fuzzycl.concept.ext\_threshold\_concept*), 51  
 ExtThresholdConcept (class in *fuzzy\_dl\_owl2.fuzzycl.concept.ext\_threshold\_concept*), 50

## F

f (*RepresentativeIndividual* attribute), 84  
 f\_name (*DefuzzifyQuery* attribute), 139  
 f\_name (*RepresentativeIndividual* attribute), 84  
 FEATURE\_DIV (*FuzzyDLKeyword* attribute), 159  
 FEATURE\_MUL (*FuzzyDLKeyword* attribute), 159  
 FEATURE\_SUB (*FuzzyDLKeyword* attribute), 159  
 FEATURE\_SUM (*FuzzyDLKeyword* attribute), 159  
 FeatureFunction (class in *fuzzy\_dl\_owl2.fuzzycl.feature\_function*), 77  
 FeatureFunctionType (class in *fuzzy\_dl\_owl2.fuzzycl.util.constants*), 157  
 FILENAME (in module *fuzzy\_dl\_owl2.fuzzycl.parser.dl\_parser*), 135  
 FILENAME (in module *fuzzy\_dl\_owl2.fuzzycl.util.util*), 169  
 fillers\_to\_show (*Individual* attribute), 83  
 form\_inv\_role\_inc\_axioms() (*KnowledgeBase* method), 98  
 form\_inv\_role\_relations() (*KnowledgeBase* method), 98  
 form\_inv\_trans\_roles() (*KnowledgeBase* method), 98  
 FREE (*FuzzyDLKeyword* attribute), 159  
 FreeVar (in module *fuzzy\_dl\_owl2.fuzzycl.milp.variable*), 132  
 from\_str() (*MILPProvider* static method), 166  
 FULL\_CLASS\_DEBUG\_PRINT (in module *fuzzy\_dl\_owl2.fuzzycl.util.utils*), 169  
 FUNCTIONAL (*FuzzyDLKeyword* attribute), 159  
 functional\_roles (*KnowledgeBase* attribute), 115

fuzzy\_datatypes (*FuzzyOwl2 attribute*), 176  
 fuzzy\_dl\_owl2  
   module, 35  
 fuzzy\_dl\_owl2.fuzzydl  
   module, 35  
 fuzzy\_dl\_owl2.fuzzydl.assertion  
   module, 35  
 fuzzy\_dl\_owl2.fuzzydl.assertion.assertion  
   module, 35  
 fuzzy\_dl\_owl2.fuzzydl.assertion.atomic\_assertion  
   module, 36  
 fuzzy\_dl\_owl2.fuzzydl.classification\_node  
   module, 36  
 fuzzy\_dl\_owl2.fuzzydl.concept  
   module, 37  
 fuzzy\_dl\_owl2.fuzzydl.concept.all\_some\_concept  
   module, 37  
 fuzzy\_dl\_owl2.fuzzydl.concept.approximation\_concept  
   module, 38  
 fuzzy\_dl\_owl2.fuzzydl.concept.atomic\_concept  
   module, 39  
 fuzzy\_dl\_owl2.fuzzydl.concept.choquet\_integral  
   module, 40  
 fuzzy\_dl\_owl2.fuzzydl.concept.concept  
   module, 41  
 fuzzy\_dl\_owl2.fuzzydl.concept.concrete  
   module, 43  
 fuzzy\_dl\_owl2.fuzzydl.concept.concrete.crisp\_-  
   concrete\_concept  
   module, 43  
 fuzzy\_dl\_owl2.fuzzydl.concept.concrete.fuzzy\_-  
   concrete\_concept  
   module, 44  
 fuzzy\_dl\_owl2.fuzzydl.concept.concrete.fuzzy\_number  
   module, 45  
 fuzzy\_dl\_owl2.fuzzydl.concept.concrete.fuzzy\_-  
   number.triangular\_fuzzy\_number  
   module, 45  
 fuzzy\_dl\_owl2.fuzzydl.concept.concrete.left\_-  
   concrete\_concept  
   module, 46  
 fuzzy\_dl\_owl2.fuzzydl.concept.concrete.linear\_-  
   concrete\_concept  
   module, 46  
 fuzzy\_dl\_owl2.fuzzydl.concept.concrete.modified\_-  
   concrete\_concept  
   module, 47  
 fuzzy\_dl\_owl2.fuzzydl.concept.concrete.right\_-  
   concrete\_concept  
   module, 48  
 fuzzy\_dl\_owl2.fuzzydl.concept.concrete.trapezoidal\_-  
   concrete\_concept  
   module, 48  
 fuzzy\_dl\_owl2.fuzzydl.concept.concrete.triangular\_-  
   concrete\_concept  
   module, 49  
 fuzzy\_dl\_owl2.fuzzydl.concept.ext\_threshold\_concept  
   module, 50  
 fuzzy\_dl\_owl2.fuzzydl.concept.has\_value\_concept  
   module, 51  
 fuzzy\_dl\_owl2.fuzzydl.concept.implies\_concept  
   module, 51  
 fuzzy\_dl\_owl2.fuzzydl.concept.interface  
   module, 53  
 fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_-  
   concept\_interface  
   module, 53  
 fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_-  
   concepts\_interface  
   module, 53  
 fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_role\_-  
   concept\_interface  
   module, 54  
 fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_role\_-  
   interface  
   module, 54  
 fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_value\_-  
   interface  
   module, 54  
 fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_-  
   weighted\_concepts\_interface  
   module, 54  
 fuzzy\_dl\_owl2.fuzzydl.concept.modified  
   module, 55  
 fuzzy\_dl\_owl2.fuzzydl.concept.modified.linearly\_-  
   modified\_concept  
   module, 55  
 fuzzy\_dl\_owl2.fuzzydl.concept.modified.modified\_-  
   concept  
   module, 55  
 fuzzy\_dl\_owl2.fuzzydl.concept.modified.triangularly\_-  
   modified\_concept  
   module, 56  
 fuzzy\_dl\_owl2.fuzzydl.concept.negated\_nominal  
   module, 57  
 fuzzy\_dl\_owl2.fuzzydl.concept.operator\_concept  
   module, 57  
 fuzzy\_dl\_owl2.fuzzydl.concept.owa\_concept  
   module, 61  
 fuzzy\_dl\_owl2.fuzzydl.concept.qowa\_concept  
   module, 61  
 fuzzy\_dl\_owl2.fuzzydl.concept.quasi\_sugeno\_integral  
   module, 62  
 fuzzy\_dl\_owl2.fuzzydl.concept.self\_concept  
   module, 63  
 fuzzy\_dl\_owl2.fuzzydl.concept.sigma\_concept  
   module, 63  
 fuzzy\_dl\_owl2.fuzzydl.concept.sigma\_count  
   module, 64  
 fuzzy\_dl\_owl2.fuzzydl.concept.string\_concept  
   module, 65  
 fuzzy\_dl\_owl2.fuzzydl.concept.sugeno\_integral  
   module, 65  
 fuzzy\_dl\_owl2.fuzzydl.concept.threshold\_concept  
   module, 66  
 fuzzy\_dl\_owl2.fuzzydl.concept.truth\_concept  
   module, 67  
 fuzzy\_dl\_owl2.fuzzydl.concept.value\_concept  
   module, 68  
 fuzzy\_dl\_owl2.fuzzydl.concept.weighted\_concept  
   module, 69  
 fuzzy\_dl\_owl2.fuzzydl.concept.weighted\_max\_concept  
   module, 70  
 fuzzy\_dl\_owl2.fuzzydl.concept.weighted\_min\_concept  
   module, 70  
 fuzzy\_dl\_owl2.fuzzydl.concept.weighted\_sum\_concept  
   module, 71  
 fuzzy\_dl\_owl2.fuzzydl.concept.weighted\_sum\_zero\_-  
   concept  
   module, 72  
 fuzzy\_dl\_owl2.fuzzydl.concept\_equivalence  
   module, 72  
 fuzzy\_dl\_owl2.fuzzydl.concrete\_feature  
   module, 73  
 fuzzy\_dl\_owl2.fuzzydl.degree  
   module, 73  
 fuzzy\_dl\_owl2.fuzzydl.degree.degree  
   module, 73  
 fuzzy\_dl\_owl2.fuzzydl.degree.degree\_expression  
   module, 74  
 fuzzy\_dl\_owl2.fuzzydl.degree.degree\_numeric  
   module, 75  
 fuzzy\_dl\_owl2.fuzzydl.degree.degree\_variable  
   module, 76  
 fuzzy\_dl\_owl2.fuzzydl.domain\_axiom

- module, 76
- fuzzy\_dl\_owl2.fuzzydl.exception
  - module, 77
- fuzzy\_dl\_owl2.fuzzydl.exception.fuzzy\_ontology\_exception
  - module, 77
- fuzzy\_dl\_owl2.fuzzydl.exception.inconsistent\_ontology\_exception
  - module, 77
- fuzzy\_dl\_owl2.fuzzydl.feature\_function
  - module, 77
- fuzzy\_dl\_owl2.fuzzydl.fuzzydl\_to\_owl2
  - module, 78
- fuzzy\_dl\_owl2.fuzzydl.general\_concept\_inclusion
  - module, 80
- fuzzy\_dl\_owl2.fuzzydl.individual
  - module, 81
- fuzzy\_dl\_owl2.fuzzydl.individual.created\_individual
  - module, 81
- fuzzy\_dl\_owl2.fuzzydl.individual.individual
  - module, 82
- fuzzy\_dl\_owl2.fuzzydl.individual.representative\_individual
  - module, 83
- fuzzy\_dl\_owl2.fuzzydl.knowledge\_base
  - module, 84
- fuzzy\_dl\_owl2.fuzzydl.label
  - module, 120
- fuzzy\_dl\_owl2.fuzzydl.milp
  - module, 121
- fuzzy\_dl\_owl2.fuzzydl.milp.expression
  - module, 121
- fuzzy\_dl\_owl2.fuzzydl.milp.inequation
  - module, 122
- fuzzy\_dl\_owl2.fuzzydl.milp.milp\_helper
  - module, 123
- fuzzy\_dl\_owl2.fuzzydl.milp.show\_variables\_helper
  - module, 127
- fuzzy\_dl\_owl2.fuzzydl.milp.solution
  - module, 129
- fuzzy\_dl\_owl2.fuzzydl.milp.term
  - module, 130
- fuzzy\_dl\_owl2.fuzzydl.milp.variable
  - module, 130
- fuzzy\_dl\_owl2.fuzzydl.modifier
  - module, 132
- fuzzy\_dl\_owl2.fuzzydl.modifier.linear\_modifier
  - module, 132
- fuzzy\_dl\_owl2.fuzzydl.modifier.modifier
  - module, 133
- fuzzy\_dl\_owl2.fuzzydl.modifier.triangular\_modifier
  - module, 134
- fuzzy\_dl\_owl2.fuzzydl.parser
  - module, 134
- fuzzy\_dl\_owl2.fuzzydl.parser.dl\_parser
  - module, 134
- fuzzy\_dl\_owl2.fuzzydl.primitive\_concept\_definition
  - module, 135
- fuzzy\_dl\_owl2.fuzzydl.query
  - module, 136
- fuzzy\_dl\_owl2.fuzzydl.query.all\_instances\_query
  - module, 136
- fuzzy\_dl\_owl2.fuzzydl.query.bnp\_query
  - module, 137
- fuzzy\_dl\_owl2.fuzzydl.query.classification\_query
  - module, 138
- fuzzy\_dl\_owl2.fuzzydl.query.defuzzify
  - module, 138
- fuzzy\_dl\_owl2.fuzzydl.query.defuzzify.defuzzify\_query
  - module, 138
- fuzzy\_dl\_owl2.fuzzydl.query.defuzzify.lom\_defuzzify\_query
  - module, 139
- fuzzy\_dl\_owl2.fuzzydl.query.defuzzify.mom\_defuzzify\_query
  - module, 139
- fuzzy\_dl\_owl2.fuzzydl.query.defuzzify.som\_defuzzify\_query
  - module, 140
- fuzzy\_dl\_owl2.fuzzydl.query.instance\_query
  - module, 140
- fuzzy\_dl\_owl2.fuzzydl.query.kb\_satisfiable\_query
  - module, 140
- fuzzy\_dl\_owl2.fuzzydl.query.max
  - module, 141
- fuzzy\_dl\_owl2.fuzzydl.query.max.max\_instance\_query
  - module, 141
- fuzzy\_dl\_owl2.fuzzydl.query.max.max\_query
  - module, 141
- fuzzy\_dl\_owl2.fuzzydl.query.max.max\_related\_query
  - module, 142
- fuzzy\_dl\_owl2.fuzzydl.query.max.max\_satisfiable\_query
  - module, 142
- fuzzy\_dl\_owl2.fuzzydl.query.max.max\_subsumes\_query
  - module, 143
- fuzzy\_dl\_owl2.fuzzydl.query.min
  - module, 143
- fuzzy\_dl\_owl2.fuzzydl.query.min.min\_instance\_query
  - module, 143
- fuzzy\_dl\_owl2.fuzzydl.query.min.min\_query
  - module, 144
- fuzzy\_dl\_owl2.fuzzydl.query.min.min\_related\_query
  - module, 144
- fuzzy\_dl\_owl2.fuzzydl.query.min.min\_satisfiable\_query
  - module, 145
- fuzzy\_dl\_owl2.fuzzydl.query.min.min\_subsumes\_query
  - module, 145
- fuzzy\_dl\_owl2.fuzzydl.query.query
  - module, 146
- fuzzy\_dl\_owl2.fuzzydl.query.related\_query
  - module, 146
- fuzzy\_dl\_owl2.fuzzydl.query.satisfiable\_query
  - module, 147
- fuzzy\_dl\_owl2.fuzzydl.query.subsumption\_query
  - module, 147
- fuzzy\_dl\_owl2.fuzzydl.range\_axiom
  - module, 148
- fuzzy\_dl\_owl2.fuzzydl.relation
  - module, 148
- fuzzy\_dl\_owl2.fuzzydl.restriction
  - module, 149
- fuzzy\_dl\_owl2.fuzzydl.restriction.has\_value\_restriction
  - module, 149
- fuzzy\_dl\_owl2.fuzzydl.restriction.restriction
  - module, 149
- fuzzy\_dl\_owl2.fuzzydl.role\_parent\_with\_degree
  - module, 150
- fuzzy\_dl\_owl2.fuzzydl.util
  - module, 150
- fuzzy\_dl\_owl2.fuzzydl.util.config\_reader
  - module, 150
- fuzzy\_dl\_owl2.fuzzydl.util.constants
  - module, 151
- fuzzy\_dl\_owl2.fuzzydl.util.util
  - module, 168
- fuzzy\_dl\_owl2.fuzzydl.util.utils
  - module, 169
- fuzzy\_dl\_owl2.fuzzyowl2
  - module, 170
- fuzzy\_dl\_owl2.fuzzyowl2.fuzzyowl2
  - module, 170
- fuzzy\_dl\_owl2.fuzzyowl2.fuzzyowl2\_to\_fuzzydl
  - module, 170

module, 176  
 fuzzy\_dl\_owl2.fuzzyowl2.owl\_types  
   module, 183  
 fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.choquet\_concept  
   module, 183  
 fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.concept\_definition  
   module, 183  
 fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.fuzzy\_datatype  
   module, 183  
 fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.fuzzy\_modifier  
   module, 184  
 fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.fuzzy\_nominal\_-  
   concept  
     module, 184  
 fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.fuzzy\_property  
   module, 184  
 fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.left\_shoulder\_-  
   function  
     module, 185  
 fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.linear\_function  
   module, 185  
 fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.linear\_modifier  
   module, 185  
 fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.modified\_concept  
   module, 186  
 fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.modified\_function  
   module, 186  
 fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.modified\_property  
   module, 186  
 fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.owa\_concept  
   module, 187  
 fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.property\_-  
   definition  
     module, 187  
 fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.qowa\_concept  
   module, 187  
 fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.quasi\_sugeno\_-  
   concept  
     module, 188  
 fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.right\_shoulder\_-  
   function  
     module, 188  
 fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.sugeno\_concept  
   module, 188  
 fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.trapezoidal\_-  
   function  
     module, 189  
 fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.triangular\_-  
   function  
     module, 189  
 fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.triangular\_modifier  
   module, 189  
 fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.weighted\_concept  
   module, 190  
 fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.weighted\_max\_-  
   concept  
     module, 190  
 fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.weighted\_min\_-  
   concept  
     module, 191  
 fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.weighted\_sum\_-  
   concept  
     module, 191  
 fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.weighted\_sum\_-  
   zero\_concept  
     module, 191  
 fuzzy\_dl\_owl2.fuzzyowl2.parser  
   module, 192  
 fuzzy\_dl\_owl2.fuzzyowl2.parser.owl2\_parser  
   module, 192  
 fuzzy\_dl\_owl2.fuzzyowl2.parser.owl2\_xml\_parser  
   module, 192  
 fuzzy\_dl\_owl2.fuzzyowl2.util

module, 193  
 fuzzy\_dl\_owl2.fuzzyowl2.util.constants  
   module, 193  
 fuzzy\_dl\_owl2.fuzzyowl2.util.fuzzy\_xml  
   module, 196  
 fuzzy\_label (FuzzyOwl2 attribute), 176  
 FUZZY\_LABEL (FuzzyOWL2Keyword attribute), 195  
 FUZZY\_LOGIC (FuzzyOWL2Keyword attribute), 195  
 fuzzy\_modifiers (FuzzyOwl2 attribute), 176  
 FUZZY\_NOMINAL (ConceptType attribute), 193  
 FUZZY\_NUMBER (ConceptType attribute), 153  
 FUZZY\_NUMBER\_COMPLEMENT (ConceptType attribute), 153  
 fuzzy\_numbers (KnowledgeBase attribute), 115  
 FUZZY\_OWL\_2 (FuzzyOWL2Keyword attribute), 195  
 FUZZY\_TYPE (FuzzyOWL2Keyword attribute), 195  
 FuzzyConcreteConcept (class in fuzzy\_dl\_-  
   owl2.fuzzydl.concept.concrete.fuzzy\_concrete\_concept),  
   44  
 FuzzyDatatype (class in  
   fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.fuzzy\_datatype), 183  
 FuzzyDLKeyword (class in fuzzy\_dl\_owl2.fuzzydl.util.constants),  
   158  
 FuzzydlToOwl2 (class in fuzzy\_dl\_owl2.fuzzydl.fuzzydl\_to\_owl2),  
   78  
 fuzzyLabel (FuzzydlToOwl2 attribute), 79  
 FuzzyLogic (class in fuzzy\_dl\_owl2.fuzzydl.util.constants), 162  
 FuzzyModifier (class in  
   fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.fuzzy\_modifier), 184  
 FuzzyNominalConcept (class in fuzzy\_dl\_owl2.fuzzyowl2.owl\_-  
   types.fuzzy\_nominal\_concept), 184  
 FuzzyOntologyException, 77  
 FuzzyOwl2 (class in fuzzy\_dl\_owl2.fuzzyowl2.fuzzyowl2), 170  
 FuzzyOWL2Keyword (class in  
   fuzzy\_dl\_owl2.fuzzyowl2.util.constants), 193  
 FuzzyOwl2Parser (class in  
   fuzzy\_dl\_owl2.fuzzyowl2.parser.owl2\_parser), 192  
 FuzzyOwl2ToFuzzyDL (class in  
   fuzzy\_dl\_owl2.fuzzyowl2.fuzzyowl2\_to\_fuzzydl), 176  
 FuzzyOwl2XMLParser (class in  
   fuzzy\_dl\_owl2.fuzzyowl2.parser.owl2\_xml\_parser), 192  
 FuzzyProperty (class in  
   fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.fuzzy\_property), 184  
 FuzzyXML (class in fuzzy\_dl\_owl2.fuzzyowl2.util.fuzzy\_xml), 196

## G

gci\_transform\_define\_atomic\_concept() (KnowledgeBase  
   method), 98  
 gci\_transformation() (KnowledgeBase method), 98  
 gci\_transformation\_add\_axiom\_to\_C\_is\_a\_X()  
   (KnowledgeBase method), 98  
 gci\_transformations\_A\_is\_a\_C() (KnowledgeBase method), 98  
 gci\_transformations\_C\_is\_a\_A() (KnowledgeBase method), 99  
 gci\_transformations\_C\_is\_a\_D() (KnowledgeBase method), 99  
 GeneralConceptInclusion (class in  
   fuzzy\_dl\_owl2.fuzzydl.general\_concept\_inclusion), 80  
 GEQ (FuzzyOWL2Keyword attribute), 195  
 geq\_equation() (DatatypeReasoner static method), 88  
 get\_a() (LeftShoulderFunction method), 185  
 get\_a() (LinearFunction method), 185  
 get\_a() (RightShoulderFunction method), 188  
 get\_a() (TrapezoidalFunction method), 189  
 get\_a() (TriangularFunction method), 189  
 get\_a() (TriangularModifier method), 190  
 get\_A\_t\_C() (KnowledgeBase method), 99  
 get\_annotations\_for\_axiom() (FuzzydlToOwl2 method), 79  
 get\_atom() (OperatorConcept method), 58  
 get\_atomic\_concept\_name() (FuzzyOwl2 method), 170  
 get\_atomic\_concept\_name() (FuzzyOwl2ToFuzzyDL method),  
   176  
 get\_atomic\_concepts() (AtomicConcept method), 40  
 get\_atomic\_concepts() (Thing method), 42  
 get\_atomic\_concepts() (TruthConcept method), 68  
 get\_atomic\_concepts\_names() (Thing method), 42



[get\\_atomic\\_data\\_property\\_name\(\)](#) (*FuzzyOwl2 method*), 170  
[get\\_atomic\\_data\\_property\\_name\(\)](#) (*FuzzyOwl2ToFuzzyDL method*), 176  
[get\\_atomic\\_object\\_property\\_name\(\)](#) (*FuzzyOwl2 method*), 170  
[get\\_atomic\\_object\\_property\\_name\(\)](#) (*FuzzyOwl2ToFuzzyDL method*), 176  
[get\\_atoms\(\)](#) (*AllSomeConcept method*), 38  
[get\\_atoms\(\)](#) (*AtomicConcept method*), 40  
[get\\_atoms\(\)](#) (*OperatorConcept method*), 58  
[get\\_atoms\(\)](#) (*Thing method*), 42  
[get\\_atoms\(\)](#) (*TruthConcept method*), 68  
[get\\_b\(\)](#) (*LeftShoulderFunction method*), 185  
[get\\_b\(\)](#) (*LinearFunction method*), 185  
[get\\_b\(\)](#) (*RightShoulderFunction method*), 188  
[get\\_b\(\)](#) (*TrapezoidalFunction method*), 189  
[get\\_b\(\)](#) (*TriangularFunction method*), 189  
[get\\_b\(\)](#) (*TriangularModifier method*), 190  
[get\\_base\(\)](#) (*FuzzydlToOwl2 method*), 79  
[get\\_best\\_non\\_fuzzy\\_performance\(\)](#) (*TriangularFuzzyNumber method*), 45  
[get\\_binary\\_variable\(\)](#) (*Variable static method*), 131  
[get\\_bottom\(\)](#) (*TruthConcept static method*), 68  
[get\\_bottom\\_concept\\_name\(\)](#) (*FuzzyOwl2 method*), 170  
[get\\_bottom\\_concept\\_name\(\)](#) (*FuzzyOwl2ToFuzzyDL method*), 177  
[get\\_bottom\\_data\\_property\\_name\(\)](#) (*FuzzyOwl2 method*), 170  
[get\\_bottom\\_data\\_property\\_name\(\)](#) (*FuzzyOwl2ToFuzzyDL method*), 177  
[get\\_bottom\\_object\\_property\\_name\(\)](#) (*FuzzyOwl2 method*), 170  
[get\\_bottom\\_object\\_property\\_name\(\)](#) (*FuzzyOwl2ToFuzzyDL method*), 177  
[get\\_bounds\(\)](#) (*DatatypeReasoner static method*), 88  
[get\\_c\(\)](#) (*LinearModifier method*), 185  
[get\\_c\(\)](#) (*TrapezoidalFunction method*), 189  
[get\\_c\(\)](#) (*TriangularFunction method*), 189  
[get\\_c\(\)](#) (*TriangularModifier method*), 190  
[get\\_c1\(\)](#) (*ConceptEquivalence method*), 72  
[get\\_c2\(\)](#) (*ConceptEquivalence method*), 72  
[get\\_caseless\\_attrrib\(\)](#) (*FuzzyOwl2XMLParser static method*), 192  
[get\\_class\(\)](#) (*FuzzydlToOwl2 method*), 79  
[get\\_class\\_name\(\)](#) (*FuzzyOwl2 method*), 170  
[get\\_classification\\_node\(\)](#) (*KnowledgeBase method*), 99  
[get\\_clauses\(\)](#) (*AtomicConcept method*), 40  
[get\\_clauses\(\)](#) (*OperatorConcept method*), 58  
[get\\_clauses\(\)](#) (*Thing method*), 42  
[get\\_coeff\(\)](#) (*Term method*), 130  
[get\\_concept\(\)](#) (*Assertion method*), 35  
[get\\_concept\(\)](#) (*KnowledgeBase method*), 99  
[get\\_concept\(\)](#) (*Restriction method*), 149  
[get\\_concept\(\)](#) (*SigmaConcept method*), 64  
[get\\_concept\(\)](#) (*SigmaCount method*), 65  
[get\\_concept\\_from\\_number\(\)](#) (*KnowledgeBase method*), 99  
[get\\_concept\\_name\(\)](#) (*AtomicAssertion method*), 36  
[get\\_concepts\(\)](#) (*ChoquetConcept method*), 183  
[get\\_concepts\(\)](#) (*Individual method*), 83  
[get\\_concepts\(\)](#) (*OwaConcept method*), 187  
[get\\_concepts\(\)](#) (*QowaConcept method*), 187  
[get\\_concepts\(\)](#) (*QsugenoConcept method*), 188  
[get\\_concepts\(\)](#) (*SugenoConcept method*), 189  
[get\\_constant\(\)](#) (*Expression method*), 122  
[get\\_constant\(\)](#) (*Inequation method*), 123  
[get\\_constant\\_term\(\)](#) (*Expression method*), 122  
[get\\_continuous\\_variable\(\)](#) (*Variable static method*), 131  
[get\\_correct\\_version\\_of\\_individual\(\)](#) (*KnowledgeBase method*), 99  
[get\\_created\\_individual\\_and\\_variables\(\)](#) (*DatatypeReasoner static method*), 88  
[get\\_d\(\)](#) (*ModifiedFunction method*), 186  
[get\\_d\(\)](#) (*TrapezoidalFunction method*), 189  
[get\\_data\\_all\\_values\\_from\\_name\(\)](#) (*FuzzyOwl2 method*), 170  
[get\\_data\\_all\\_values\\_from\\_name\(\)](#) (*FuzzyOwl2ToFuzzyDL method*), 177  
[get\\_data\\_exact\\_cardinality\\_restriction\(\)](#) (*FuzzyOwl2 method*), 170  
[get\\_data\\_exact\\_cardinality\\_restriction\(\)](#) (*FuzzyOwl2ToFuzzyDL method*), 177  
[get\\_data\\_has\\_value\\_name\(\)](#) (*FuzzyOwl2 method*), 170  
[get\\_data\\_has\\_value\\_name\(\)](#) (*FuzzyOwl2ToFuzzyDL method*), 177  
[get\\_data\\_max\\_cardinality\\_restriction\(\)](#) (*FuzzyOwl2 method*), 170  
[get\\_data\\_max\\_cardinality\\_restriction\(\)](#) (*FuzzyOwl2ToFuzzyDL method*), 177  
[get\\_data\\_min\\_cardinality\\_restriction\(\)](#) (*FuzzyOwl2 method*), 170  
[get\\_data\\_min\\_cardinality\\_restriction\(\)](#) (*FuzzyOwl2ToFuzzyDL method*), 177  
[get\\_data\\_property\(\)](#) (*FuzzydlToOwl2 method*), 79  
[get\\_data\\_property\\_name\(\)](#) (*FuzzyOwl2 method*), 170  
[get\\_data\\_some\\_values\\_from\\_name\(\)](#) (*FuzzyOwl2 method*), 171  
[get\\_data\\_some\\_values\\_from\\_name\(\)](#) (*FuzzyOwl2ToFuzzyDL method*), 177  
[get\\_datatype\\_filler\\_type\(\)](#) (*Variable method*), 131  
[get\\_defined\\_concept\(\)](#) (*PrimitiveConceptDefinition method*), 136  
[get\\_definition\(\)](#) (*PrimitiveConceptDefinition method*), 136  
[get\\_degree\(\)](#) (*AtomicAssertion method*), 36  
[get\\_degree\(\)](#) (*Degree static method*), 74  
[get\\_degree\(\)](#) (*DegreeExpression static method*), 74  
[get\\_degree\(\)](#) (*DegreeNumeric static method*), 75  
[get\\_degree\(\)](#) (*DegreeVariable static method*), 76  
[get\\_degree\(\)](#) (*FuzzyNominalConcept method*), 184  
[get\\_degree\(\)](#) (*GeneralConceptInclusion method*), 80  
[get\\_degree\(\)](#) (*PrimitiveConceptDefinition method*), 136  
[get\\_degree\(\)](#) (*Relation method*), 148  
[get\\_degree\(\)](#) (*Restriction method*), 149  
[get\\_degree\(\)](#) (*RoleParentWithDegree method*), 150  
[get\\_degrees\(\)](#) (*AllInstancesQuery method*), 137  
[get\\_depth\(\)](#) (*CreatedIndividual method*), 81  
[get\\_expression\(\)](#) (*DegreeExpression method*), 74  
[get\\_feature\(\)](#) (*DatatypeReasoner static method*), 88  
[get\\_feature\\_name\(\)](#) (*RepresentativeIndividual method*), 84  
[get\\_features\(\)](#) (*FeatureFunction method*), 77  
[get\\_full\\_name\(\)](#) (*ClassificationNode method*), 37  
[get\\_fuzzy\\_concept\(\)](#) (*ModifiedConcept method*), 186  
[get\\_fuzzy\\_concept\(\)](#) (*SigmaConcept method*), 64  
[get\\_fuzzy\\_concept\(\)](#) (*WeightedConcept method*), 190  
[get\\_fuzzy\\_modifier\(\)](#) (*ModifiedConcept method*), 186  
[get\\_fuzzy\\_modifier\(\)](#) (*ModifiedProperty method*), 186  
[get\\_fuzzy\\_modifier\(\)](#) (*PropertyDefinition method*), 187  
[get\\_fuzzy\\_number\(\)](#) (*RepresentativeIndividual method*), 84  
[get\\_grammatics\(\)](#) (*DLParser static method*), 135  
[get\\_grammatics\(\)](#) (*FuzzyOwl2Parser static method*), 192  
[get\\_immediate\\_predecessors\(\)](#) (*ClassificationNode method*), 37  
[get\\_immediate\\_successors\(\)](#) (*ClassificationNode method*), 37  
[get\\_inclusion\\_degree\(\)](#) (*KnowledgeBase method*), 99  
[get\\_individual\(\)](#) (*Assertion method*), 35  
[get\\_individual\(\)](#) (*FuzzydlToOwl2 method*), 79  
[get\\_individual\(\)](#) (*FuzzyNominalConcept method*), 184  
[get\\_individual\(\)](#) (*HasValueRestriction method*), 149  
[get\\_individual\(\)](#) (*KnowledgeBase method*), 99  
[get\\_individual\(\)](#) (*RepresentativeIndividual method*), 84  
[get\\_individual\(\)](#) (*SigmaCount method*), 65  
[get\\_individual\\_name\(\)](#) (*FuzzyOwl2 method*), 171  
[get\\_individual\\_name\(\)](#) (*FuzzyOwl2ToFuzzyDL method*), 177  
[get\\_individuals\(\)](#) (*AllInstancesQuery method*), 137  
[get\\_individuals\(\)](#) (*KnowledgeBase method*), 99  
[get\\_individuals\(\)](#) (*SigmaConcept method*), 64  
[get\\_individuals\(\)](#) (*SigmaCount method*), 65  
[get\\_integer\\_id\(\)](#) (*CreatedIndividual method*), 81  
[get\\_integer\\_variable\(\)](#) (*Variable static method*), 131  
[get\\_inverses\\_of\\_inverse\\_role\(\)](#) (*KnowledgeBase method*), 99



get\_k1() (*ConcreteFeature method*), 73  
 get\_k2() (*ConcreteFeature method*), 73  
 get\_kb() (*DLParser static method*), 135  
 get\_labels() (*ShowVariablesHelper method*), 128  
 get\_language() (*KnowledgeBase method*), 99  
 get\_logic() (*KnowledgeBase method*), 99  
 get\_lower\_bound() (*Variable method*), 131  
 get\_lower\_limit() (*Assertion method*), 35  
 get\_max\_value() (*FuzzyDatatype method*), 184  
 get\_membership\_degree() (*CrispConcreteConcept method*), 44  
 get\_membership\_degree() (*FuzzyConcreteConcept method*), 44  
 get\_membership\_degree() (*LeftConcreteConcept method*), 46  
 get\_membership\_degree() (*LinearConcreteConcept method*), 47  
 get\_membership\_degree() (*LinearModifier method*), 132  
 get\_membership\_degree() (*ModifiedConcreteConcept method*), 47  
 get\_membership\_degree() (*Modifier method*), 133  
 get\_membership\_degree() (*RightConcreteConcept method*), 48  
 get\_membership\_degree() (*TrapezoidalConcreteConcept method*), 49  
 get\_membership\_degree() (*TriangularConcreteConcept method*), 50  
 get\_membership\_degree() (*TriangularModifier method*), 134  
 get\_min\_value() (*FuzzyDatatype method*), 184  
 get\_mod() (*ModifiedFunction method*), 186  
 get\_name() (*ConcreteFeature method*), 73  
 get\_name() (*FuzzyDLKeyword method*), 158  
 get\_name() (*FuzzyOWL2Keyword method*), 194  
 get\_name() (*ShowVariablesHelper method*), 128  
 get\_name\_for\_integer() (*MILPHelper method*), 125  
 get\_name\_without\_degree() (*Assertion method*), 36  
 get\_name\_without\_degree() (*HasValueRestriction method*), 149  
 get\_name\_without\_degree() (*Relation method*), 148  
 get\_name\_without\_degree() (*Restriction method*), 149  
 get\_named\_individuals() (*KnowledgeBase method*), 99  
 get\_negated\_nominal\_variable() (*MILPHelper method*), 125  
 get\_new\_atomic\_class() (*FuzzydlToOwl2 method*), 79  
 get\_new\_atomic\_concept() (*KnowledgeBase method*), 99  
 get\_new\_concrete\_individual() (*KnowledgeBase method*), 99  
 get\_new\_individual() (*KnowledgeBase method*), 99  
 get\_new\_individual\_common\_code() (*KnowledgeBase method*), 100  
 get\_new\_variable() (*MILPHelper method*), 125  
 get\_new\_variable() (*Variable static method*), 131  
 get\_nominal\_list() (*Individual method*), 83  
 get\_nominal\_variable() (*MILPHelper method*), 125  
 get\_number() (*FeatureFunction method*), 77  
 get\_number() (*WeightedConcept method*), 190  
 get\_number\_for\_assertion() (*MILPHelper method*), 125  
 get\_number\_from\_concept() (*KnowledgeBase method*), 100  
 get\_number\_of\_domain\_restrictions() (*KnowledgeBase method*), 100  
 get\_number\_of\_range\_restrictions() (*KnowledgeBase method*), 100  
 get\_numerical\_value() (*DegreeNumeric method*), 75  
 get\_obj\_expression() (*DefuzzifyQuery method*), 138  
 get\_obj\_expression() (*LomDefuzzifyQuery method*), 139  
 get\_obj\_expression() (*MomDefuzzifyQuery method*), 139  
 get\_obj\_expression() (*SomDefuzzifyQuery method*), 140  
 get\_object\_all\_values\_from\_name() (*FuzzyOwl2 method*), 171  
 get\_object\_all\_values\_from\_name() (*FuzzyOwl2ToFuzzyDL method*), 177  
 get\_object\_complement\_of\_name() (*FuzzyOwl2 method*), 171  
 get\_object\_complement\_of\_name() (*FuzzyOwl2ToFuzzyDL method*), 177  
 get\_object\_exact\_cardinality\_restriction() (*FuzzyOwl2 method*), 171  
 get\_object\_exact\_cardinality\_restriction() (*FuzzyOwl2ToFuzzyDL method*), 177  
 get\_object\_has\_self\_name() (*FuzzyOwl2 method*), 171  
 get\_object\_has\_self\_name() (*FuzzyOwl2ToFuzzyDL method*), 177  
 get\_object\_has\_value\_name() (*FuzzyOwl2 method*), 171  
 get\_object\_has\_value\_name() (*FuzzyOwl2ToFuzzyDL method*), 177  
 get\_object\_individual() (*Relation method*), 148  
 get\_object\_intersection\_of\_name() (*FuzzyOwl2 method*), 171  
 get\_object\_intersection\_of\_name() (*FuzzyOwl2ToFuzzyDL method*), 178  
 get\_object\_max\_cardinality\_restriction() (*FuzzyOwl2 method*), 171  
 get\_object\_max\_cardinality\_restriction() (*FuzzyOwl2ToFuzzyDL method*), 178  
 get\_object\_min\_cardinality\_restriction() (*FuzzyOwl2 method*), 171  
 get\_object\_min\_cardinality\_restriction() (*FuzzyOwl2ToFuzzyDL method*), 178  
 get\_object\_one\_of\_name() (*FuzzyOwl2 method*), 171  
 get\_object\_one\_of\_name() (*FuzzyOwl2ToFuzzyDL method*), 178  
 get\_object\_property() (*FuzzydlToOwl2 method*), 79  
 get\_object\_property\_name() (*FuzzyOwl2 method*), 171  
 get\_object\_some\_values\_from\_name() (*FuzzyOwl2 method*), 171  
 get\_object\_some\_values\_from\_name() (*FuzzyOwl2ToFuzzyDL method*), 178  
 get\_object\_union\_of\_name() (*FuzzyOwl2 method*), 171  
 get\_object\_union\_of\_name() (*FuzzyOwl2ToFuzzyDL method*), 178  
 get\_one() (*DegreeNumeric static method*), 75  
 get\_ordered\_permutation() (*MILPHelper method*), 125  
 get\_output\_edges() (*ClassificationNode method*), 37  
 get\_parent() (*CreatedIndividual method*), 81  
 get\_parent() (*RoleParentWithDegree method*), 150  
 get\_parent\_name() (*CreatedIndividual method*), 81  
 get\_property() (*ModifiedProperty method*), 187  
 get\_property() (*PropertyDefinition method*), 187  
 get\_quantifier() (*QowaConcept method*), 187  
 get\_representative() (*CreatedIndividualHandler static method*), 85  
 get\_representative\_if\_exists() (*CreatedIndividual method*), 81  
 get\_representative\_if\_exists() (*Individual method*), 83  
 get\_role() (*SigmaConcept method*), 64  
 get\_role() (*SigmaCount method*), 65  
 get\_role\_name() (*CreatedIndividual method*), 82  
 get\_role\_name() (*Relation method*), 148  
 get\_role\_name() (*Restriction method*), 149  
 get\_roles() (*AllSomeConcept method*), 38  
 get\_roles() (*ApproximationConcept method*), 39  
 get\_roles() (*AtomicConcept method*), 40  
 get\_roles() (*ChoquetIntegral method*), 41  
 get\_roles() (*ExtThresholdConcept method*), 50  
 get\_roles() (*FuzzyConcreteConcept method*), 44  
 get\_roles() (*HasValueConcept method*), 51  
 get\_roles() (*ImpliesConcept method*), 52  
 get\_roles() (*ModifiedConcept method*), 56  
 get\_roles() (*OperatorConcept method*), 58  
 get\_roles() (*OwaConcept method*), 61  
 get\_roles() (*SelfConcept method*), 63  
 get\_roles() (*SigmaConcept method*), 64  
 get\_roles() (*StringConcept method*), 65  
 get\_roles() (*SugenoIntegral method*), 66  
 get\_roles() (*Thing method*), 42  
 get\_roles() (*ThresholdConcept method*), 67  
 get\_roles() (*TruthConcept method*), 68  
 get\_roles() (*ValueConcept method*), 69  
 get\_roles() (*WeightedConcept method*), 69  
 get\_roles() (*WeightedMaxConcept method*), 70  
 get\_roles() (*WeightedMinConcept method*), 71  
 get\_roles() (*WeightedSumConcept method*), 71  
 get\_roles() (*WeightedSumZeroConcept method*), 72  
 get\_semi\_continuous\_variable() (*Variable static method*), 131  
 get\_short\_name() (*FuzzyOwl2 method*), 171  
 get\_short\_name() (*FuzzyOwl2ToFuzzyDL method*), 178

get\_showed\_variables() (Solution method), 129  
 get\_solution() (Solution method), 129  
 get\_str\_value() (FuzzyOWL2Keyword method), 194  
 get\_string\_type() (Inequation method), 123  
 get\_subject\_individual() (Relation method), 148  
 get\_subsumed() (GeneralConceptInclusion method), 80  
 get\_subsumer() (GeneralConceptInclusion method), 80  
 get\_subsumption\_flags() (KnowledgeBase method), 100  
 get\_tag\_name() (FuzzyOWL2Keyword method), 194  
 get\_terms() (Expression method), 122  
 get\_terms() (Inequation method), 123  
 get\_tmp\_feature() (KnowledgeBase method), 100  
 get\_top() (TruthConcept static method), 68  
 get\_top\_concept\_name() (FuzzyOwl2 method), 172  
 get\_top\_concept\_name() (FuzzyOwl2ToFuzzyDL method), 178  
 get\_top\_data\_property\_name() (FuzzyOwl2 method), 172  
 get\_top\_data\_property\_name() (FuzzyOwl2ToFuzzyDL method), 178  
 get\_top\_object\_property\_name() (FuzzyOwl2 method), 172  
 get\_top\_object\_property\_name() (FuzzyOwl2ToFuzzyDL method), 178  
 get\_total\_time() (Query method), 146  
 get\_truth\_constants() (KnowledgeBase method), 100  
 get\_type() (Assertion method), 36  
 get\_type() (ConceptDefinition method), 183  
 get\_type() (ConcreteFeature method), 73  
 get\_type() (FeatureFunction method), 77  
 get\_type() (GeneralConceptInclusion method), 80  
 get\_type() (Inequation method), 123  
 get\_type() (PrimitiveConceptDefinition method), 136  
 get\_type() (RepresentativeIndividual method), 84  
 get\_type() (Variable method), 131  
 get\_upper\_bound() (Variable method), 131  
 get\_value() (FuzzyDLKeyword method), 158  
 get\_value() (FuzzyOWL2Keyword method), 194  
 get\_var() (Term method), 130  
 get\_variable() (DegreeVariable method), 76  
 get\_variable() (MLPHelper method), 125  
 get\_variable() (SigmaCount method), 65  
 get\_variables() (ShowVariablesHelper method), 128  
 get\_weighted\_concepts() (WeightedMaxConcept method), 190  
 get\_weighted\_concepts() (WeightedMinConcept method), 191  
 get\_weighted\_concepts() (WeightedSumConcept method), 191  
 get\_weighted\_concepts() (WeightedSumZeroConcept method), 192  
 get\_weights() (ChoquetConcept method), 183  
 get\_weights() (OwaConcept method), 187  
 get\_weights() (QsugenoConcept method), 188  
 get\_weights() (SugenoConcept method), 189  
 get\_xb() (DatatypeReasoner static method), 88  
 global\_abstract\_fillers (ShowVariablesHelper attribute), 129  
 global\_concrete\_fillers (ShowVariablesHelper attribute), 129  
 GOEDEL (FuzzyOWL2Keyword attribute), 195  
 GOEDEL (LogicOperatorType attribute), 166  
 GOEDEL\_AND (ConceptType attribute), 153  
 GOEDEL\_AND (FuzzyDLKeyword attribute), 160  
 goedel\_and() (OperatorConcept static method), 58  
 goedel\_cnf() (Thing method), 43  
 goedel\_dnf() (Thing method), 43  
 GOEDEL\_IMPLIES (ConceptType attribute), 153  
 GOEDEL\_IMPLIES (FuzzyDLKeyword attribute), 160  
 goedel\_implies() (ImpliesConcept static method), 52  
 goedel\_implies() (KnowledgeBase method), 100  
 goedel\_implies\_equation() (ZadehSolver static method), 119  
 goedel\_not\_equation() (ZadehSolver static method), 119  
 GOEDEL\_OR (ConceptType attribute), 153  
 GOEDEL\_OR (FuzzyDLKeyword attribute), 160  
 goedel\_or() (OperatorConcept static method), 58  
 GoedelAnd (in module  
     fuzzy\_dl\_owl2.fuzzydl.concept.operator\_concept), 60  
 GoedelImplies (in module  
     fuzzy\_dl\_owl2.fuzzydl.concept.implies\_concept), 53

GoedelOr (in module  
     fuzzy\_dl\_owl2.fuzzydl.concept.operator\_concept), 60  
 GRE (FuzzyOWL2Keyword attribute), 195  
 GREATER\_EQUAL (RepresentativeIndividualType attribute), 167  
 GREATER\_THAN (InequalityType attribute), 162  
 GREATER\_THAN\_OR\_EQUAL\_TO (FuzzyDLKeyword attribute), 160  
 greater\_then() (Inequation static method), 123  
 GreaterThan (in module fuzzy\_dl\_owl2.fuzzydl.milp.inequation), 123  
 GUROBI (MILPProvider attribute), 166

## H

has\_defined\_range() (TriangularFuzzyNumber static method), 45  
 has\_functional\_abstract\_roles() (KnowledgeBase method), 100  
 has\_integer\_value() (Util static method), 169  
 has\_name() (ClassificationNode method), 37  
 has\_nominal\_variable() (MLPHelper method), 126  
 has\_nominals() (Thing method), 43  
 has\_nominals\_inabox() (KnowledgeBase method), 100  
 has\_nominals\_inabox() (KnowledgeBase method), 100  
 has\_only\_crisp\_sub\_concepts() (KnowledgeBase method), 100  
 HAS\_VALUE (ConceptType attribute), 153  
 HAS\_VALUE (FuzzyDLKeyword attribute), 160  
 has\_value() (HasValueConcept static method), 51  
 has\_variable() (MLPHelper method), 126  
 HasConceptInterface (class in fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_concept\_interface), 53  
 HasConceptsInterface (class in fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_concepts\_interface), 53  
 HasRoleConceptInterface (class in  
     fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_role\_concept\_interface), 54  
 HasRoleInterface (class in fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_role\_interface), 54  
 HasValueConcept (class in  
     fuzzy\_dl\_owl2.fuzzydl.concept.has\_value\_concept), 51  
 HasValueInterface (class in fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_value\_interface), 54  
 HasValueRestriction (class in  
     fuzzy\_dl\_owl2.fuzzydl.restriction.has\_value\_restriction), 149  
 HasWeightedConceptsInterface (class in  
     fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_weighted\_concepts\_interface), 55

implication (PrimitiveConceptDefinition attribute), 136  
 IMPLIES (FuzzyDLKeyword attribute), 160  
 implies() (KnowledgeBase method), 100  
 IMPLIES\_ROLE (FuzzyDLKeyword attribute), 160  
 ImpliesConcept (class in  
     fuzzy\_dl\_owl2.fuzzydl.concept.implies\_concept), 52  
 INCONSISTENT\_KB (Solution attribute), 130  
 InconsistentOntologyException, 77  
 increment\_constant() (Expression method), 122  
 ind (InstanceQuery attribute), 140  
 ind (RepresentativeIndividual attribute), 84  
 ind1 (MaxRelatedQuery attribute), 142  
 ind1 (MinRelatedQuery attribute), 145  
 ind1 (RelatedQuery attribute), 147  
 ind2 (MaxRelatedQuery attribute), 142  
 ind2 (MinRelatedQuery attribute), 145  
 ind2 (RelatedQuery attribute), 147  
 ind\_a (Relation attribute), 148  
 ind\_b (Relation attribute), 148  
 ind\_name (HasValueRestriction attribute), 149  
 ind\_name (NegatedNominal property), 57  
 individual (Assertion attribute), 36

- Individual (class in fuzzy\_dl\_owl2.fuzzydl.individual.individual), 82
- INDIVIDUAL (FuzzyOWL2Keyword attribute), 195
- individual (SigmaCount attribute), 65
- individual\_iri() (FuzzyDLToOwl2 method), 79
- individual\_set\_intersection\_of() (CreatedIndividual method), 82
- IndividualHandler (class in fuzzy\_dl\_owl2.fuzzydl.knowledge\_base), 90
- individuals (AllInstancesQuery attribute), 137
- individuals (KnowledgeBase attribute), 115
- individuals (ShowVariablesHelper attribute), 129
- individuals (SigmaConcept attribute), 64
- individuals (SigmaCount attribute), 65
- InequalityType (class in fuzzy\_dl\_owl2.fuzzydl.util.constants), 162
- Inequation (class in fuzzy\_dl\_owl2.fuzzydl.milp.inequation), 122
- info() (Util static method), 169
- initial\_time (Query attribute), 146
- INPUT\_EDGES (ClassificationNode attribute), 37
- input\_FDL (FuzzyDLToOwl2 attribute), 79
- INSTANCE (FuzzyDLKeyword attribute), 160
- InstanceQuery (class in fuzzy\_dl\_owl2.fuzzydl.query.instance\_query), 140
- INTEGER (ConcreteFeatureType attribute), 156
- INTEGER (FuzzyDLKeyword attribute), 160
- INTEGER (VariableType attribute), 168
- INTEGER\_MAX\_VALUE (FuzzyOwl2ToFuzzyDL attribute), 182
- INTEGER\_MIN\_VALUE (FuzzyOwl2ToFuzzyDL attribute), 182
- IntegerVar (in module fuzzy\_dl\_owl2.fuzzydl.milp.variable), 132
- INVERSE (FuzzyDLKeyword attribute), 160
- INVERSE\_APPROXIMATION (ApproximationConcept attribute), 39
- INVERSE\_FUNCTIONAL (FuzzyDLKeyword attribute), 160
- inverse\_functional\_roles (KnowledgeBase attribute), 115
- inverse\_roles (KnowledgeBase attribute), 115
- iri() (FuzzyDLToOwl2 method), 79
- is\_and() (OperatorConcept static method), 58
- is\_assertion\_processed() (KnowledgeBase method), 101
- is\_atomic() (AtomicConcept method), 40
- is\_atomic() (Concept method), 42
- is\_atomic() (OperatorConcept method), 58
- is\_atomic() (TruthConcept method), 68
- is\_atomic\_crisp\_concept() (KnowledgeBase method), 101
- is\_blockable() (CreatedIndividual method), 82
- is\_blockable() (Individual method), 83
- is\_blocked() (CreatedIndividualHandler static method), 85
- is\_classified() (KnowledgeBase method), 101
- is\_complemented\_atomic() (AllSomeConcept method), 38
- is\_complemented\_atomic() (AtomicConcept method), 40
- is\_complemented\_atomic() (Concept method), 42
- is\_complemented\_atomic() (OperatorConcept method), 58
- is\_complemented\_atomic() (TruthConcept method), 68
- is\_concrete() (AtomicConcept method), 40
- is\_concrete() (CreatedIndividual method), 82
- is\_concrete() (FuzzyConcreteConcept method), 44
- is\_concrete() (ModifiedConcept method), 56
- is\_concrete() (OperatorConcept method), 58
- is\_concrete() (Thing method), 43
- is\_concrete() (TriangularFuzzyNumber method), 45
- is\_concrete\_type() (KnowledgeBase method), 101
- is\_consistent\_kb() (KbSatisfiableQuery method), 141
- is\_consistent\_kb() (Solution method), 130
- is\_crisp\_concept() (KnowledgeBase method), 101
- is\_crisp\_concept() (MILPHelper method), 126
- is\_crisp\_role() (KnowledgeBase method), 101
- is\_crisp\_role() (MILPHelper method), 126
- is\_directly\_anywhere\_pairwise\_blocked() (CreatedIndividualHandler static method), 85
- is\_directly\_anywhere\_simple\_blocked() (CreatedIndividualHandler static method), 85
- is\_directly\_blocked() (CreatedIndividualHandler static method), 85
- is\_directly\_pairwise\_blocked() (CreatedIndividualHandler static method), 85
- is\_directly\_simple\_blocked() (CreatedIndividualHandler static method), 86
- is\_indirectly\_anywhere\_pairwise\_blocked() (CreatedIndividualHandler static method), 86
- is\_indirectly\_anywhere\_simple\_blocked() (CreatedIndividualHandler static method), 86
- is\_indirectly\_blocked() (CreatedIndividualHandler static method), 86
- is\_indirectly\_pairwise\_blocked() (CreatedIndividualHandler static method), 86
- is\_indirectly\_simple\_blocked() (CreatedIndividualHandler static method), 86
- is\_lazy\_unfoldable() (KnowledgeBase method), 101
- is\_loaded() (KnowledgeBase method), 101
- is\_nominal\_variable() (MILPHelper method), 126
- is\_not\_at\_least\_value() (OperatorConcept static method), 58
- is\_not\_at\_most\_value() (OperatorConcept static method), 59
- is\_not\_choquet() (OperatorConcept static method), 59
- is\_not\_concrete() (OperatorConcept static method), 59
- is\_not\_exact\_value() (OperatorConcept static method), 59
- is\_not\_ext\_neg\_threshold() (OperatorConcept static method), 59
- is\_not\_ext\_pos\_threshold() (OperatorConcept static method), 59
- is\_not\_fuzzy\_number() (OperatorConcept static method), 59
- is\_not\_goedel\_implies() (OperatorConcept static method), 59
- is\_not\_has\_value() (OperatorConcept static method), 59
- is\_not\_modified() (OperatorConcept static method), 59
- is\_not\_neg\_threshold() (OperatorConcept static method), 59
- is\_not\_owa() (OperatorConcept static method), 59
- is\_not\_pos\_threshold() (OperatorConcept static method), 59
- is\_not\_qowa() (OperatorConcept static method), 59
- is\_not\_quasi\_sugeno() (OperatorConcept static method), 59
- is\_not\_self() (OperatorConcept static method), 59
- is\_not\_sigma\_concept() (OperatorConcept static method), 59
- is\_not\_sugeno() (OperatorConcept static method), 59
- is\_not\_type() (OperatorConcept static method), 59
- is\_not\_weighted() (OperatorConcept static method), 59
- is\_not\_weighted\_max() (OperatorConcept static method), 59
- is\_not\_weighted\_min() (OperatorConcept static method), 59
- is\_not\_weighted\_sum() (OperatorConcept static method), 59
- is\_not\_weighted\_sum\_zero() (OperatorConcept static method), 59
- is\_not\_zadeh\_implies() (OperatorConcept static method), 59
- is\_nothing() (ClassificationNode method), 37
- is\_number() (TriangularFuzzyNumber method), 45
- is\_number\_not\_one() (Degree method), 74
- is\_number\_not\_one() (DegreeExpression method), 74
- is\_number\_not\_one() (DegreeNumeric method), 75
- is\_number\_not\_one() (DegreeVariable method), 76
- is\_number\_zero() (Degree method), 74
- is\_number\_zero() (DegreeExpression method), 75
- is\_number\_zero() (DegreeNumeric method), 75
- is\_number\_zero() (DegreeVariable method), 76
- is\_numeric() (Degree method), 74
- is\_numeric() (DegreeExpression method), 75
- is\_numeric() (DegreeNumeric method), 75
- is\_numeric() (DegreeVariable method), 76
- is\_or() (OperatorConcept static method), 59
- is\_redundant\_A\_is\_a\_C() (KnowledgeBase method), 101
- is\_redundant\_gci() (KnowledgeBase method), 101
- is\_reserved\_word() (FuzzyOwl2ToFuzzyDL static method), 178
- is\_simplified() (OperatorConcept method), 59
- is\_simplified() (Thing method), 43
- is\_tbox\_acyclic() (KnowledgeBase method), 102
- is\_thing() (ClassificationNode method), 37
- is\_zero() (Inequation method), 123

## K

- k1 (CrispConcreteConcept attribute), 44
- k1 (FuzzyConcreteConcept property), 44



**k1** (*LeftConcreteConcept* attribute), 46  
**k1** (*LinearConcreteConcept* attribute), 47  
**k1** (*ModifiedConcreteConcept* attribute), 48  
**k1** (*RightConcreteConcept* attribute), 48  
**k1** (*TrapezoidalConcreteConcept* attribute), 49  
**k1** (*TriangularConcreteConcept* attribute), 50  
**K1** (*TriangularFuzzyNumber* attribute), 46  
**k2** (*CrispConcreteConcept* attribute), 44  
**k2** (*FuzzyConcreteConcept* property), 44  
**k2** (*LeftConcreteConcept* attribute), 46  
**k2** (*LinearConcreteConcept* attribute), 47  
**k2** (*ModifiedConcreteConcept* attribute), 48  
**k2** (*RightConcreteConcept* attribute), 48  
**k2** (*TrapezoidalConcreteConcept* attribute), 49  
**k2** (*TriangularConcreteConcept* attribute), 50  
**K2** (*TriangularFuzzyNumber* attribute), 46  
**kb** (*DLParser* attribute), 135  
**KB\_LOADED** (*KnowledgeBase* attribute), 113  
**KB\_UNSAT** (*KnowledgeBase* attribute), 113  
**KbSatisfiableQuery** (class in *fuzzy\_dl\_owl2.fuzzycl.query.kb\_satisfiable\_query*), 140  
**KLEENE\_DIENES** (*LogicOperatorType* attribute), 166  
**KLEENE\_DIENES\_IMPLIES** (*FuzzyDLKeyword* attribute), 160  
**kleene\_dienes\_implies()** (*ImpliesConcept* static method), 52  
**kleene\_dienes\_implies()** (*KnowledgeBase* method), 102  
**kleene\_dienes\_implies\_equation()** (*ZadehSolver* static method), 119  
**KleeneDienesImplies** (in module *fuzzy\_dl\_owl2.fuzzycl.concept.implies\_concept*), 53  
**KNOWLEDGE\_BASE\_SEMANTICS** (in module *fuzzy\_dl\_owl2.fuzzycl.util.constants*), 168  
**KnowledgeBase** (class in *fuzzy\_dl\_owl2.fuzzycl.knowledge\_base*), 91  
**KnowledgeBaseRules** (class in *fuzzy\_dl\_owl2.fuzzycl.util.constants*), 163

**L**

**Label** (class in *fuzzy\_dl\_owl2.fuzzycl.label*), 120  
**labels\_for\_fillers** (*ShowVariablesHelper* attribute), 129  
**labels\_with\_nodes** (*KnowledgeBase* attribute), 115  
**language** (*KnowledgeBase* attribute), 115  
**lazy\_unfondable** (*KnowledgeBase* attribute), 115  
**LEFT\_SHOULDER** (*FuzzyDLKeyword* attribute), 160  
**LEFT\_SHOULDER** (*FuzzyOWL2Keyword* attribute), 195  
**LeftConcreteConcept** (class in *fuzzy\_dl\_owl2.fuzzycl.concept.concrete.left\_concrete\_concept*), 46  
**LeftShoulderFunction** (class in *fuzzy\_dl\_owl2.fuzzycl.owl\_types.left\_shoulder\_function*), 185  
**LEQ** (*FuzzyOWL2Keyword* attribute), 195  
**LES** (*FuzzyOWL2Keyword* attribute), 195  
**LESS\_EQUAL** (*RepresentativeIndividualType* attribute), 167  
**LESS\_THAN** (*InequalityType* attribute), 163  
**less\_than()** (*Inequation* static method), 123  
**LESS\_THAN\_OR\_EQUAL\_TO** (*FuzzyDLKeyword* attribute), 160  
**LessThan** (in module *fuzzy\_dl\_owl2.fuzzycl.milp.inequation*), 123  
**LINEAR** (*FuzzyDLKeyword* attribute), 160  
**LINEAR** (*FuzzyOWL2Keyword* attribute), 195  
**LINEAR\_MODIFIER** (*FuzzyDLKeyword* attribute), 160  
**LinearConcreteConcept** (class in *fuzzy\_dl\_owl2.fuzzycl.concept.concrete.linear\_concrete\_concept*), 47  
**LinearFunction** (class in *fuzzy\_dl\_owl2.fuzzycl.owl\_types.linear\_function*), 185  
**LinearlyModifiedConcept** (class in *fuzzy\_dl\_owl2.fuzzycl.concept.modified.linearly\_modified\_concept*), 55  
**LinearModifier** (class in *fuzzy\_dl\_owl2.fuzzycl.modifier.linear\_modifier*), 132  
**LinearModifier** (class in *fuzzy\_dl\_owl2.fuzzycl.owl\_types.linear\_modifier*), 185

**list\_of\_concepts** (*Individual* attribute), 83  
**load\_config()** (*DLParser* static method), 135  
**load\_config()** (*FuzzyOwl2Parser* static method), 192  
**load\_config()** (*FuzzyOwl2XMLParser* static method), 192  
**load\_parameters()** (*ConfigReader* static method), 150  
**log2()** (*Util* static method), 169  
**LOG\_DIR** (in module *fuzzy\_dl\_owl2.fuzzycl.parser.dl\_parser*), 135  
**LOG\_DIR** (in module *fuzzy\_dl\_owl2.fuzzycl.util.util*), 169  
**logger** (in module *fuzzy\_dl\_owl2.fuzzycl.util.util*), 169  
**LOGIC** (*FuzzyOWL2Keyword* attribute), 195  
**LogicOperatorType** (class in *fuzzy\_dl\_owl2.fuzzycl.util.constants*), 165  
**LomDefuzzifyQuery** (class in *fuzzy\_dl\_owl2.fuzzycl.query.defuzzify.lom\_defuzzify\_query*), 139  
**LOOSE\_LOWER\_APPROX** (*ConceptType* attribute), 153  
**loose\_lower\_approx()** (*ApproximationConcept* static method), 39  
**LOOSE\_LOWER\_APPROXIMATION** (*FuzzyDLKeyword* attribute), 160  
**LOOSE\_UPPER\_APPROX** (*ConceptType* attribute), 154  
**loose\_upper\_approx()** (*ApproximationConcept* static method), 39  
**LOOSE\_UPPER\_APPROXIMATION** (*FuzzyDLKeyword* attribute), 160  
**LooseLowerApprox** (in module *fuzzy\_dl\_owl2.fuzzycl.concept.approximation\_concept*), 39  
**LooseUpperApprox** (in module *fuzzy\_dl\_owl2.fuzzycl.concept.approximation\_concept*), 39  
**LOWER\_APPROX** (*ConceptType* attribute), 154  
**lower\_approx()** (*ApproximationConcept* static method), 39  
**LOWER\_APPROXIMATION** (*FuzzyDLKeyword* attribute), 160  
**lower\_bound** (*Variable* attribute), 131  
**LowerApprox** (in module *fuzzy\_dl\_owl2.fuzzycl.concept.approximation\_concept*), 39  
**LUKASIEWICZ** (*FuzzyDLKeyword* attribute), 160  
**LUKASIEWICZ** (*FuzzyLogic* attribute), 162  
**LUKASIEWICZ** (*FuzzyOWL2Keyword* attribute), 195  
**LUKASIEWICZ** (*LogicOperatorType* attribute), 166  
**LUKASIEWICZ\_AND** (*ConceptType* attribute), 154  
**LUKASIEWICZ\_AND** (*FuzzyDLKeyword* attribute), 160  
**lukasiewicz\_and()** (*OperatorConcept* static method), 60  
**lukasiewicz\_cnf()** (*Thing* method), 43  
**lukasiewicz\_dnf()** (*Thing* method), 43  
**LUKASIEWICZ\_IMPLIES** (*FuzzyDLKeyword* attribute), 160  
**lukasiewicz\_implies()** (*ImpliesConcept* static method), 52  
**lukasiewicz\_implies()** (*KnowledgeBase* method), 102  
**LUKASIEWICZ\_OR** (*ConceptType* attribute), 154  
**LUKASIEWICZ\_OR** (*FuzzyDLKeyword* attribute), 160  
**lukasiewicz\_or()** (*OperatorConcept* static method), 60  
**LukasiewiczAnd** (in module *fuzzy\_dl\_owl2.fuzzycl.concept.operator\_concept*), 60  
**LukasiewiczImplies** (in module *fuzzy\_dl\_owl2.fuzzycl.concept.implies\_concept*), 53  
**LukasiewiczOr** (in module *fuzzy\_dl\_owl2.fuzzycl.concept.operator\_concept*), 61  
**LukasiewiczSolver** (class in *fuzzy\_dl\_owl2.fuzzycl.knowledge\_base*), 116

## M

**main()** (*DLParser* static method), 135  
**main()** (*FuzzyOwl2Parser* static method), 192  
**main()** (*FuzzyOwl2XMLParser* static method), 192  
**main()** (in module *fuzzy\_dl\_owl2.fuzzycl.fuzzycl\_to\_owl2*), 80  
**mark\_indirectly\_blocked()** (*CreatedIndividual* method), 82  
**mark\_indirectly\_simple\_unchecked()** (*CreatedIndividualHandler* static method), 86  
**mark\_process\_assertion()** (*KnowledgeBase* method), 102  
**match\_concept\_labels()** (*CreatedIndividualHandler* static method), 86  
**match\_set\_concept\_labels()** (*CreatedIndividualHandler* static method), 86

`match_subset_concept_labels()` (*CreatedIndividualHandler* static method), 87  
`matching_individual()` (*CreatedIndividualHandler* static method), 87  
`max_depth` (*KnowledgeBase* attribute), 115  
`MAX_G_SUBS_QUERY` (*FuzzyDLKeyword* attribute), 160  
`MAX_INDIVIDUALS` (*ConfigReader* attribute), 150  
`MAX_INSTANCE_QUERY` (*FuzzyDLKeyword* attribute), 160  
`MAX_KD_SUBS_QUERY` (*FuzzyDLKeyword* attribute), 160  
`MAX_L_SUBS_QUERY` (*FuzzyDLKeyword* attribute), 160  
`MAX_RELATED_QUERY` (*FuzzyDLKeyword* attribute), 160  
`MAX_SAT_QUERY` (*FuzzyDLKeyword* attribute), 160  
`MAX_SUBS_QUERY` (*FuzzyDLKeyword* attribute), 160  
`MAX_VAR_QUERY` (*FuzzyDLKeyword* attribute), 160  
`MaxInstanceQuery` (class in *fuzzy\_dl\_owl2.fuzzydl.query.max.max\_instance\_query*), 141  
`MaxQuery` (class in *fuzzy\_dl\_owl2.fuzzydl.query.max.max\_query*), 141  
`MaxRelatedQuery` (class in *fuzzy\_dl\_owl2.fuzzydl.query.max.max\_related\_query*), 142  
`MaxSatisfiableQuery` (class in *fuzzy\_dl\_owl2.fuzzydl.query.max.max\_satisfiable\_query*), 143  
`MaxSubsumesQuery` (class in *fuzzy\_dl\_owl2.fuzzydl.query.max.max\_subsumes\_query*), 143  
`MAXVAL` (in module *fuzzy\_dl\_owl2.fuzzydl.util.constants*), 168  
`MAXVAL2` (in module *fuzzy\_dl\_owl2.fuzzydl.util.constants*), 168  
`merge()` (*KnowledgeBase* method), 102  
`merge_fillers()` (*KnowledgeBase* method), 103  
`milp` (*KnowledgeBase* attribute), 115  
`MILP_PROVIDER` (*ConfigReader* attribute), 150  
`MILPHelper` (class in *fuzzy\_dl\_owl2.fuzzydl.milp.milp\_helper*), 123  
`MILPProvider` (class in *fuzzy\_dl\_owl2.fuzzydl.util.constants*), 166  
`MIN_G_SUBS_QUERY` (*FuzzyDLKeyword* attribute), 160  
`MIN_INSTANCE_QUERY` (*FuzzyDLKeyword* attribute), 160  
`MIN_KD_SUBS_QUERY` (*FuzzyDLKeyword* attribute), 161  
`MIN_L_SUBS_QUERY` (*FuzzyDLKeyword* attribute), 161  
`MIN_RELATED_QUERY` (*FuzzyDLKeyword* attribute), 161  
`MIN_SAT_QUERY` (*FuzzyDLKeyword* attribute), 161  
`MIN_SUBS_QUERY` (*FuzzyDLKeyword* attribute), 161  
`MIN_VAR_QUERY` (*FuzzyDLKeyword* attribute), 161  
`MinInstanceQuery` (class in *fuzzy\_dl\_owl2.fuzzydl.query.min.min\_instance\_query*), 144  
`MinQuery` (class in *fuzzy\_dl\_owl2.fuzzydl.query.min.min\_query*), 144  
`MinRelatedQuery` (class in *fuzzy\_dl\_owl2.fuzzydl.query.min.min\_related\_query*), 144  
`MinSatisfiableQuery` (class in *fuzzy\_dl\_owl2.fuzzydl.query.min.min\_satisfiable\_query*), 145  
`MinSubsumesQuery` (class in *fuzzy\_dl\_owl2.fuzzydl.query.min.min\_subsumes\_query*), 145  
`minus()` (*TriangularFuzzyNumber* static method), 46  
`MIP` (*MILPProvider* attribute), 166  
`MODIFIED` (*ConceptType* attribute), 154  
`MODIFIED` (*FuzzyDLKeyword* attribute), 161  
`MODIFIED` (*FuzzyOWL2Keyword* attribute), 195  
`modified` (*ModifiedConcreteConcept* property), 48  
`MODIFIED_COMPLEMENT` (*ConceptType* attribute), 154  
`MODIFIED_CONCEPT` (*ConceptType* attribute), 193  
`ModifiedConcept` (class in *fuzzy\_dl\_owl2.fuzzydl.concept.modified.modified\_concept*), 56  
`ModifiedConcept` (class in *fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.modified\_concept*), 186  
`ModifiedConcreteConcept` (class in *fuzzy\_dl\_owl2.fuzzydl.concept.concrete.modified\_concrete\_concept*), 47  
`ModifiedFunction` (class in *fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.modified\_function*), 186  
`ModifiedProperty` (class in *fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.modified\_property*), 186  
`Modifier` (class in *fuzzy\_dl\_owl2.fuzzydl.modifier.modifier*), 133  
`MODIFIER` (*FuzzyOWL2Keyword* attribute), 195  
`modifier` (*ModifiedConcept* property), 56  
`modifier` (*ModifiedConcreteConcept* property), 48  
`modifiers` (*FuzzydlToOwl2* attribute), 79  
`modifiers` (*KnowledgeBase* attribute), 115  
`modify()` (*LinearModifier* method), 132  
`modify()` (*Modifier* method), 133  
`modify()` (*TriangularModifier* method), 134  
module  
  *fuzzy\_dl\_owl2*, 35  
  *fuzzy\_dl\_owl2.fuzzydl*, 35  
  *fuzzy\_dl\_owl2.fuzzydl.assertion*, 35  
  *fuzzy\_dl\_owl2.fuzzydl.assertion.assertion*, 35  
  *fuzzy\_dl\_owl2.fuzzydl.assertion.atomic\_assertion*, 36  
  *fuzzy\_dl\_owl2.fuzzydl.classification\_node*, 36  
  *fuzzy\_dl\_owl2.fuzzydl.concept*, 37  
  *fuzzy\_dl\_owl2.fuzzydl.concept.all\_some\_concept*, 37  
  *fuzzy\_dl\_owl2.fuzzydl.concept.approximation\_concept*, 38  
  *fuzzy\_dl\_owl2.fuzzydl.concept.atomic\_concept*, 39  
  *fuzzy\_dl\_owl2.fuzzydl.concept.choquet\_integral*, 40  
  *fuzzy\_dl\_owl2.fuzzydl.concept.concept*, 41  
  *fuzzy\_dl\_owl2.fuzzydl.concept.concrete*, 43  
  *fuzzy\_dl\_owl2.fuzzydl.concept.concrete.crisp\_concrete\_concept*, 43  
  *fuzzy\_dl\_owl2.fuzzydl.concept.concrete.fuzzy\_concrete\_concept*, 44  
  *fuzzy\_dl\_owl2.fuzzydl.concept.concrete.fuzzy\_number*, 45  
  *fuzzy\_dl\_owl2.fuzzydl.concept.concrete.fuzzy\_number.triangular\_fuzzy\_number*, 45  
  *fuzzy\_dl\_owl2.fuzzydl.concept.concrete.left\_concrete\_concept*, 46  
  *fuzzy\_dl\_owl2.fuzzydl.concept.concrete.linear\_concrete\_concept*, 46  
  *fuzzy\_dl\_owl2.fuzzydl.concept.concrete.modified\_concrete\_concept*, 47  
  *fuzzy\_dl\_owl2.fuzzydl.concept.concrete.right\_concrete\_concept*, 48  
  *fuzzy\_dl\_owl2.fuzzydl.concept.concrete.trapezoidal\_concrete\_concept*, 48  
  *fuzzy\_dl\_owl2.fuzzydl.concept.concrete.triangular\_concrete\_concept*, 49  
  *fuzzy\_dl\_owl2.fuzzydl.concept.ext\_threshold\_concept*, 50  
  *fuzzy\_dl\_owl2.fuzzydl.concept.has\_value\_concept*, 51  
  *fuzzy\_dl\_owl2.fuzzydl.concept.implies\_concept*, 51  
  *fuzzy\_dl\_owl2.fuzzydl.concept.interface*, 53  
  *fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_concept\_interface*, 53  
  *fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_concepts\_interface*, 53  
  *fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_role\_concept\_interface*, 54  
  *fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_role\_interface*, 54  
  *fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_value\_interface*, 54

fuzzy\_dl\_owl2.fuzzydl.concept.interface.has\_  
     weighted\_concepts\_interface, 54  
 fuzzy\_dl\_owl2.fuzzydl.concept.modified, 55  
 fuzzy\_dl\_  
     owl2.fuzzydl.concept.modified.linearly\_  
         modified\_concept, 55  
 fuzzy\_dl\_  
     owl2.fuzzydl.concept.modified.modified\_  
         concept, 55  
 fuzzy\_dl\_  
     owl2.fuzzydl.concept.modified.triangularly\_  
         modified\_concept, 56  
 fuzzy\_dl\_owl2.fuzzydl.concept.negated\_nominal, 57  
 fuzzy\_dl\_owl2.fuzzydl.concept.operator\_concept,  
     57  
 fuzzy\_dl\_owl2.fuzzydl.concept.owa\_concept, 61  
 fuzzy\_dl\_owl2.fuzzydl.concept.qowa\_concept, 61  
 fuzzy\_dl\_owl2.fuzzydl.concept.quasi\_sugeno\_  
     integral, 62  
 fuzzy\_dl\_owl2.fuzzydl.concept.self\_concept, 63  
 fuzzy\_dl\_owl2.fuzzydl.concept.sigma\_concept, 63  
 fuzzy\_dl\_owl2.fuzzydl.concept.sigma\_count, 64  
 fuzzy\_dl\_owl2.fuzzydl.concept.string\_concept, 65  
 fuzzy\_dl\_owl2.fuzzydl.concept.sugeno\_integral, 65  
 fuzzy\_dl\_owl2.fuzzydl.concept.threshold\_concept,  
     66  
 fuzzy\_dl\_owl2.fuzzydl.concept.truth\_concept, 67  
 fuzzy\_dl\_owl2.fuzzydl.concept.value\_concept, 68  
 fuzzy\_dl\_owl2.fuzzydl.concept.weighted\_concept,  
     69  
 fuzzy\_dl\_owl2.fuzzydl.concept.weighted\_max\_  
     concept, 70  
 fuzzy\_dl\_owl2.fuzzydl.concept.weighted\_min\_  
     concept, 70  
 fuzzy\_dl\_owl2.fuzzydl.concept.weighted\_sum\_  
     concept, 71  
 fuzzy\_dl\_owl2.fuzzydl.concept.weighted\_sum\_  
     zero\_concept, 72  
 fuzzy\_dl\_owl2.fuzzydl.concept.equivalence, 72  
 fuzzy\_dl\_owl2.fuzzydl.concrete\_feature, 73  
 fuzzy\_dl\_owl2.fuzzydl.degree, 73  
 fuzzy\_dl\_owl2.fuzzydl.degree.degree, 73  
 fuzzy\_dl\_owl2.fuzzydl.degree.degree\_expression,  
     74  
 fuzzy\_dl\_owl2.fuzzydl.degree.degree\_numeric, 75  
 fuzzy\_dl\_owl2.fuzzydl.degree.degree\_variable, 76  
 fuzzy\_dl\_owl2.fuzzydl.domain\_axiom, 76  
 fuzzy\_dl\_owl2.fuzzydl.exception, 77  
 fuzzy\_dl\_owl2.fuzzydl.exception.fuzzy\_  
     ontology\_exception, 77  
 fuzzy\_dl\_owl2.fuzzydl.exception.inconsistent\_  
     ontology\_exception, 77  
 fuzzy\_dl\_owl2.fuzzydl.feature\_function, 77  
 fuzzy\_dl\_owl2.fuzzydl.fuzzydl\_to\_owl2, 78  
 fuzzy\_dl\_owl2.fuzzydl.general\_concept\_inclusion,  
     80  
 fuzzy\_dl\_owl2.fuzzydl.individual, 81  
 fuzzy\_dl\_owl2.fuzzydl.individual.created\_  
     individual, 81  
 fuzzy\_dl\_owl2.fuzzydl.individual.individual, 82  
 fuzzy\_dl\_  
     owl2.fuzzydl.individual.representative\_  
         individual, 83  
 fuzzy\_dl\_owl2.fuzzydl.knowledge\_base, 84  
 fuzzy\_dl\_owl2.fuzzydl.label, 120  
 fuzzy\_dl\_owl2.fuzzydl.milp, 121  
 fuzzy\_dl\_owl2.fuzzydl.milp.expression, 121  
 fuzzy\_dl\_owl2.fuzzydl.milp.inequation, 122  
 fuzzy\_dl\_owl2.fuzzydl.milp.milp\_helper, 123  
 fuzzy\_dl\_owl2.fuzzydl.milp.show\_variables\_  
     helper, 127  
 fuzzy\_dl\_owl2.fuzzydl.milp.solution, 129  
 fuzzy\_dl\_owl2.fuzzydl.milp.term, 130  
 fuzzy\_dl\_owl2.fuzzydl.milp.variable, 130  
 fuzzy\_dl\_owl2.fuzzydl.modifier, 132  
 fuzzy\_dl\_owl2.fuzzydl.modifier.linear\_modifier,  
     132  
 fuzzy\_dl\_owl2.fuzzydl.modifier.modifier, 133  
 fuzzy\_dl\_owl2.fuzzydl.modifier.triangular\_  
     modifier, 134  
 fuzzy\_dl\_owl2.fuzzydl.parser, 134  
 fuzzy\_dl\_owl2.fuzzydl.parser.dl\_parser, 134  
 fuzzy\_dl\_owl2.fuzzydl.primitive\_concept\_  
     definition, 135  
 fuzzy\_dl\_owl2.fuzzydl.query, 136  
 fuzzy\_dl\_owl2.fuzzydl.query.all\_instances\_query,  
     136  
 fuzzy\_dl\_owl2.fuzzydl.query.bnp\_query, 137  
 fuzzy\_dl\_owl2.fuzzydl.query.classification\_  
     query, 138  
 fuzzy\_dl\_owl2.fuzzydl.query.defuzzify, 138  
 fuzzy\_dl\_  
     owl2.fuzzydl.query.defuzzify.defuzzify\_  
         query, 138  
 fuzzy\_dl\_owl2.fuzzydl.query.defuzzify.lom\_  
     defuzzify\_query, 139  
 fuzzy\_dl\_owl2.fuzzydl.query.defuzzify.mom\_  
     defuzzify\_query, 139  
 fuzzy\_dl\_owl2.fuzzydl.query.defuzzify.som\_  
     defuzzify\_query, 140  
 fuzzy\_dl\_owl2.fuzzydl.query.instance\_query, 140  
 fuzzy\_dl\_owl2.fuzzydl.query.kb\_satisfiable\_  
     query, 140  
 fuzzy\_dl\_owl2.fuzzydl.query.max, 141  
 fuzzy\_dl\_owl2.fuzzydl.query.max.max\_instance\_  
     query, 141  
 fuzzy\_dl\_owl2.fuzzydl.query.max.max\_query, 141  
 fuzzy\_dl\_owl2.fuzzydl.query.max.max\_related\_  
     query, 142  
 fuzzy\_dl\_owl2.fuzzydl.query.max.max\_  
     satisfiable\_query, 142  
 fuzzy\_dl\_owl2.fuzzydl.query.max.max\_subsumes\_  
     query, 143  
 fuzzy\_dl\_owl2.fuzzydl.query.min, 143  
 fuzzy\_dl\_owl2.fuzzydl.query.min.min\_instance\_  
     query, 143  
 fuzzy\_dl\_owl2.fuzzydl.query.min.min\_query, 144  
 fuzzy\_dl\_owl2.fuzzydl.query.min.min\_related\_  
     query, 144  
 fuzzy\_dl\_owl2.fuzzydl.query.min.min\_  
     satisfiable\_query, 145  
 fuzzy\_dl\_owl2.fuzzydl.query.min.min\_subsumes\_  
     query, 145  
 fuzzy\_dl\_owl2.fuzzydl.query.query, 146  
 fuzzy\_dl\_owl2.fuzzydl.query.related\_query, 146  
 fuzzy\_dl\_owl2.fuzzydl.query.satisfiable\_query,  
     147  
 fuzzy\_dl\_owl2.fuzzydl.query.subsumption\_query,  
     147  
 fuzzy\_dl\_owl2.fuzzydl.range\_axiom, 148  
 fuzzy\_dl\_owl2.fuzzydl.relation, 148  
 fuzzy\_dl\_owl2.fuzzydl.restriction, 149  
 fuzzy\_dl\_owl2.fuzzydl.restriction.has\_value\_  
     restriction, 149  
 fuzzy\_dl\_owl2.fuzzydl.restriction.restriction,  
     149  
 fuzzy\_dl\_owl2.fuzzydl.role\_parent\_with\_degree,  
     150  
 fuzzy\_dl\_owl2.fuzzydl.util, 150  
 fuzzy\_dl\_owl2.fuzzydl.util.config\_reader, 150  
 fuzzy\_dl\_owl2.fuzzydl.util.constants, 151  
 fuzzy\_dl\_owl2.fuzzydl.util.util, 168  
 fuzzy\_dl\_owl2.fuzzydl.util.utils, 169  
 fuzzy\_dl\_owl2.fuzzyowl2, 170  
 fuzzy\_dl\_owl2.fuzzyowl2.fuzzyowl2, 170  
 fuzzy\_dl\_owl2.fuzzyowl2.fuzzyowl2\_to\_fuzzydl, 176

- `fuzzy_dl_owl2.fuzzyowl2.owl_types`, 183
- `fuzzy_dl_owl2.fuzzyowl2.owl_types.choquet_concept`, 183
- `fuzzy_dl_owl2.fuzzyowl2.owl_types.concept_definition`, 183
- `fuzzy_dl_owl2.fuzzyowl2.owl_types.fuzzy_datatype`, 183
- `fuzzy_dl_owl2.fuzzyowl2.owl_types.fuzzy_modifier`, 184
- `fuzzy_dl_owl2.fuzzyowl2.owl_types.fuzzy_nominal_concept`, 184
- `fuzzy_dl_owl2.fuzzyowl2.owl_types.fuzzy_property`, 184
- `fuzzy_dl_owl2.fuzzyowl2.owl_types.left_shoulder_function`, 185
- `fuzzy_dl_owl2.fuzzyowl2.owl_types.linear_function`, 185
- `fuzzy_dl_owl2.fuzzyowl2.owl_types.linear_modifier`, 185
- `fuzzy_dl_owl2.fuzzyowl2.owl_types.modified_concept`, 186
- `fuzzy_dl_owl2.fuzzyowl2.owl_types.modified_function`, 186
- `fuzzy_dl_owl2.fuzzyowl2.owl_types.modified_property`, 186
- `fuzzy_dl_owl2.fuzzyowl2.owl_types.owa_concept`, 187
- `fuzzy_dl_owl2.fuzzyowl2.owl_types.property_definition`, 187
- `fuzzy_dl_owl2.fuzzyowl2.owl_types.qowa_concept`, 187
- `fuzzy_dl_owl2.fuzzyowl2.owl_types.quasi_sugeno_concept`, 188
- `fuzzy_dl_owl2.fuzzyowl2.owl_types.right_shoulder_function`, 188
- `fuzzy_dl_owl2.fuzzyowl2.owl_types.sugeno_concept`, 188
- `fuzzy_dl_owl2.fuzzyowl2.owl_types.trapezoidal_function`, 189
- `fuzzy_dl_owl2.fuzzyowl2.owl_types.triangular_function`, 189
- `fuzzy_dl_owl2.fuzzyowl2.owl_types.triangular_modifier`, 189
- `fuzzy_dl_owl2.fuzzyowl2.owl_types.weighted_concept`, 190
- `fuzzy_dl_owl2.fuzzyowl2.owl_types.weighted_max_concept`, 190
- `fuzzy_dl_owl2.fuzzyowl2.owl_types.weighted_min_concept`, 191
- `fuzzy_dl_owl2.fuzzyowl2.owl_types.weighted_sum_concept`, 191
- `fuzzy_dl_owl2.fuzzyowl2.owl_types.weighted_sum_zero_concept`, 191
- `fuzzy_dl_owl2.fuzzyowl2.parser`, 192
- `fuzzy_dl_owl2.fuzzyowl2.parser.owl2_parser`, 192
- `fuzzy_dl_owl2.fuzzyowl2.parser.owl2_xml_parser`, 192
- `fuzzy_dl_owl2.fuzzyowl2.util`, 193
- `fuzzy_dl_owl2.fuzzyowl2.util.constants`, 193
- `fuzzy_dl_owl2.fuzzyowl2.util.fuzzy_xml`, 196
- `MomDefuzzifyQuery` (class in `fuzzy_dl_owl2.fuzzydl.query.defuzzify.mom_defuzzify_query`), 139
- `MUL` (*FuzzyDLKeyword* attribute), 161
- `multiply_constant()` (*Degree* method), 74
- `multiply_constant()` (*DegreeExpression* method), 75
- `multiply_constant()` (*DegreeNumeric* method), 75
- `multiply_constant()` (*DegreeVariable* method), 76
- `multiply_constant()` (*Expression* static method), 122
- `name` (*ExtThresholdConcept* attribute), 51
- `name` (*FuzzyConcreteConcept* attribute), 45
- `NAME` (*FuzzyOWL2Keyword* attribute), 195
- `name` (*HasValueConcept* attribute), 51
- `name` (*ImpliesConcept* attribute), 53
- `name` (*Individual* attribute), 83
- `name` (*Modifier* attribute), 133
- `name` (*NegatedNominal* attribute), 57
- `name` (*OperatorConcept* attribute), 60
- `name` (*QowaConcept* attribute), 62
- `name` (*SelfConcept* attribute), 63
- `name` (*SigmaConcept* attribute), 64
- `name` (*ThresholdConcept* attribute), 67
- `name` (*TrapezoidalConcreteConcept* attribute), 49
- `name` (*TruthConcept* attribute), 68
- `name` (*ValueConcept* attribute), 69
- `name` (*Variable* attribute), 131
- `name` (*WeightedConcept* attribute), 69
- `name` (*WeightedMaxConcept* attribute), 70
- `name` (*WeightedMinConcept* attribute), 71
- `name` (*WeightedSumConcept* attribute), 71
- `name` (*WeightedSumZeroConcept* attribute), 72
- `NEG_INFINITY` (*FuzzyOwl2* attribute), 176
- `NEG_THRESHOLD` (*ConceptType* attribute), 154
- `neg_threshold()` (*ThresholdConcept* static method), 67
- `negate_expression()` (*Expression* static method), 122
- `NegatedNominal` (class in `fuzzy_dl_owl2.fuzzydl.concept.negated_nominal`), 57
- `NegThreshold` (in module `fuzzy_dl_owl2.fuzzydl.concept.threshold_concept`), 67
- `new()` (*AllSomeConcept* static method), 38
- `new_atomic_concept()` (*AtomicConcept* static method), 40
- `NO_BLOCKING` (*BlockingDynamicType* attribute), 152
- `nodes_classification` (*KnowledgeBase* attribute), 115
- `NOMINAL` (*FuzzyOWL2Keyword* attribute), 195
- `nominal_absorption()` (*KnowledgeBase* method), 103
- `nominal_list` (*Individual* attribute), 83
- `nominal_variables` (*MILPHelper* attribute), 127
- `normal_form()` (*OperatorConcept* method), 60
- `normal_form()` (*Thing* method), 43
- `NOT` (*FuzzyDLKeyword* attribute), 161
- `Not` (in module `fuzzy_dl_owl2.fuzzydl.concept.operator_concept`), 61
- `not_()` (*OperatorConcept* static method), 60
- `NOT_AT_LEAST_VALUE` (*ConceptType* attribute), 154
- `NOT_AT_MOST_VALUE` (*ConceptType* attribute), 154
- `NOT_BLOCKED` (*CreatedIndividualBlockingType* attribute), 157
- `NOT_CHOQUET_INTEGRAL` (*ConceptType* attribute), 154
- `NOT_EXACT_VALUE` (*ConceptType* attribute), 154
- `NOT_EXT_NEG_THRESHOLD` (*ConceptType* attribute), 154
- `NOT_EXT_POS_THRESHOLD` (*ConceptType* attribute), 154
- `NOT_GOEDEL_IMPLIES` (*ConceptType* attribute), 154
- `NOT_HAS_VALUE` (*ConceptType* attribute), 154
- `NOT_NEG_THRESHOLD` (*ConceptType* attribute), 154
- `NOT_OWA` (*ConceptType* attribute), 154
- `NOT_POS_THRESHOLD` (*ConceptType* attribute), 154
- `NOT_QUANTIFIED_OWA` (*ConceptType* attribute), 154
- `NOT_QUASI_SUGENO_INTEGRAL` (*ConceptType* attribute), 154
- `NOT_SELF` (*ConceptType* attribute), 154
- `not_self_roles` (*Individual* attribute), 83
- `NOT_SIGMA_CONCEPT` (*ConceptType* attribute), 154
- `NOT_SUGENO_INTEGRAL` (*ConceptType* attribute), 154
- `NOT_W_MAX` (*ConceptType* attribute), 154
- `NOT_W_MIN` (*ConceptType* attribute), 154
- `NOT_W_SUM` (*ConceptType* attribute), 154
- `NOT_W_SUM_ZERO` (*ConceptType* attribute), 154
- `NOT_WEIGHTED` (*ConceptType* attribute), 154
- `NOT_ZADEH_IMPLIES` (*ConceptType* attribute), 154
- `num_assertions` (*KnowledgeBase* attribute), 115
- `num_classes` (*FuzzydlToOwl2* attribute), 79
- `num_defined_concepts` (*KnowledgeBase* attribute), 115
- `num_defined_individuals` (*KnowledgeBase* attribute), 115
- `num_new_concepts` (*Concept* attribute), 42
- `num_relations` (*KnowledgeBase* attribute), 115

## N

- `name` (*AllInstancesQuery* attribute), 137
- `name` (*ApproximationConcept* attribute), 39
- `name` (*Concept* property), 42



NUMBER (*FeatureFunctionType* attribute), 157  
 NUMBER (in module *fuzzy\_dl\_owl2.fuzzycl.util.constants*), 168  
 NUMBER\_DIGITS (*ConfigReader* attribute), 150  
 number\_of\_concepts (*KnowledgeBase* attribute), 115  
 number\_of\_roles (*KnowledgeBase* attribute), 115  
 number\_of\_variables (*MILPHelper* attribute), 127  
 numerical\_datatypes (*FuzzyOwl2ToFuzzyDL* attribute), 182

## O

obj\_expr (*DefuzzifyQuery* attribute), 139  
 obj\_expr (*InstanceQuery* attribute), 140  
 obj\_expr (*MaxQuery* attribute), 142  
 obj\_expr (*MinQuery* attribute), 144  
 obj\_expr (*RelatedQuery* attribute), 147  
 obj\_expr (*SubsumptionQuery* attribute), 147  
 object\_properties (*FuzzyOwl2ToFuzzyDL* attribute), 182  
 object\_property\_iri () (*FuzzyclToOwl2* method), 79  
 old\_01\_variables (*KnowledgeBase* attribute), 115  
 old\_binary\_variables (*KnowledgeBase* attribute), 115  
 ontologies (*FuzzyOwl2* attribute), 176  
 ontology (*FuzzyclToOwl2* attribute), 79  
 ontology (*FuzzyOwl2* attribute), 176  
 ONTOLOGY (*FuzzyOWL2Keyword* attribute), 195  
 ontology\_iri (*FuzzyclToOwl2* attribute), 80  
 ontology\_iri (*FuzzyOwl2* attribute), 176  
 ontology\_path (*FuzzyclToOwl2* attribute), 80  
 ontology\_path (*FuzzyOwl2* attribute), 176  
 OPEN\_TAG (*FuzzyOWL2Keyword* attribute), 195  
 OperatorConcept (class in  
   *fuzzy\_dl\_owl2.fuzzycl.concept.operator\_concept*), 58  
 OPERATORS (*OperatorConcept* attribute), 60  
 OPTIMIZATIONS (*ConfigReader* attribute), 150  
 optimize () (*KnowledgeBase* method), 103  
 optimize () (*MILPHelper* method), 126  
 OR (*ConceptType* attribute), 154  
 OR (*FuzzyDLKeyword* attribute), 161  
 Or (in module *fuzzy\_dl\_owl2.fuzzycl.concept.operator\_concept*), 61  
 or () (*OperatorConcept* static method), 60  
 or\_equation () (*LukasiewiczSolver* static method), 117  
 or\_equation () (*ZadehSolver* static method), 119  
 or\_negated\_equation () (*ZadehSolver* static method), 119  
 OR\_OPERATORS (*OperatorConcept* attribute), 60  
 order (*KnowledgeBase* attribute), 115  
 order () (*Util* static method), 169  
 output\_dl (*FuzzyOwl2* attribute), 176  
 OUTPUT\_EDGES (*ClassificationNode* attribute), 37  
 output\_FOWL (*FuzzyclToOwl2* attribute), 80  
 OWA (*ConceptType* attribute), 154, 193  
 OWA (*FuzzyDLKeyword* attribute), 161  
 OWA (*FuzzyOWL2Keyword* attribute), 195  
 OwaConcept (class in *fuzzy\_dl\_owl2.fuzzycl.concept.owa\_concept*), 61  
 OwaConcept (class in  
   *fuzzy\_dl\_owl2.fuzzycl.owl\_types.owa\_concept*), 187  
 OWL\_ANNOTATION\_LABEL (*ConfigReader* attribute), 150

## P

parent (*RoleParentWithDegree* attribute), 150  
 parse\_string () (*DLParser* static method), 135  
 parse\_string () (*FuzzyOwl2Parser* static method), 192  
 parse\_string () (*FuzzyOwl2XMLParser* static method), 193  
 parse\_string\_opt () (*DLParser* static method), 135  
 PARTITION (*MILPHelper* attribute), 127  
 partition\_loop\_A\_is\_a\_B () (*KnowledgeBase* method), 103  
 partition\_loop\_A\_is\_a\_C () (*KnowledgeBase* method), 103  
 partition\_loop\_C\_is\_a\_A () (*KnowledgeBase* method), 103  
 partition\_loop\_C\_is\_a\_D () (*KnowledgeBase* method), 103  
 partition\_loop\_to\_do\_A\_is\_a\_B () (*KnowledgeBase* method), 103  
 partition\_loop\_to\_do\_A\_is\_a\_C () (*KnowledgeBase* method), 103  
 POS\_INFINITY (*FuzzyOwl2* attribute), 176

POS\_THRESHOLD (*ConceptType* attribute), 154  
 pos\_threshold () (*ThresholdConcept* static method), 67  
 positive\_concrete\_value\_assertions (*KnowledgeBase* attribute), 115  
 PosThreshold (in module  
   *fuzzy\_dl\_owl2.fuzzycl.concept.threshold\_concept*), 67  
 preprocess () (*AllInstancesQuery* method), 137  
 preprocess () (*BtnQuery* method), 137  
 preprocess () (*ClassificationQuery* method), 138  
 preprocess () (*DefuzzifyQuery* method), 138  
 preprocess () (*KbSatisfiableQuery* method), 141  
 preprocess () (*MaxInstanceQuery* method), 141  
 preprocess () (*MaxQuery* method), 142  
 preprocess () (*MaxRelatedQuery* method), 142  
 preprocess () (*MaxSatisfiableQuery* method), 143  
 preprocess () (*MaxSubsumesQuery* method), 143  
 preprocess () (*MinInstanceQuery* method), 144  
 preprocess () (*MinQuery* method), 144  
 preprocess () (*MinRelatedQuery* method), 145  
 preprocess () (*MinSatisfiableQuery* method), 145  
 preprocess () (*MinSubsumesQuery* method), 146  
 preprocess () (*MomDefuzzifyQuery* method), 139  
 preprocess () (*Query* method), 146  
 preprocess\_tbox () (*KnowledgeBase* method), 103  
 PrimitiveConceptDefinition (class in  
   *fuzzy\_dl\_owl2.fuzzycl.primitive\_concept\_definition*), 136  
 print\_instance\_of\_labels () (*MILPHelper* method), 127  
 PRINT\_LABELS (*MILPHelper* attribute), 127  
 print\_tbox () (*KnowledgeBase* method), 103  
 PRINT\_VARIABLES (*MILPHelper* attribute), 127  
 process\_concept\_annotations () (*FuzzyOwl2* method), 172  
 process\_datatype\_annotations () (*FuzzyOwl2* method), 172  
 process\_ontology\_annotations () (*FuzzyOwl2* method), 172  
 process\_ontology\_axioms () (*FuzzyOwl2* method), 172  
 process\_property\_annotations () (*FuzzyOwl2* method), 172  
 processed\_assertions (*KnowledgeBase* attribute), 116  
 processed\_axioms (*FuzzyOwl2* attribute), 176  
 processed\_functional\_data\_properties  
   (*FuzzyOwl2ToFuzzyDL* attribute), 182  
 processed\_functional\_object\_properties  
   (*FuzzyOwl2ToFuzzyDL* attribute), 182  
 PRODUCT (*FeatureFunctionType* attribute), 157  
 PRODUCT (*FuzzyOWL2Keyword* attribute), 195  
 PropertyDefinition (class in  
   *fuzzy\_dl\_owl2.fuzzycl.owl\_types.property\_definition*), 187  
 prune () (*Individual* method), 83  
 PULP (*MILPProvider* attribute), 166  
 PULP\_CPLEX (*MILPProvider* attribute), 166  
 PULP\_GLPK (*MILPProvider* attribute), 166  
 PULP\_HIGHS (*MILPProvider* attribute), 166

## Q

Q\_OWA (*FuzzyDLKeyword* attribute), 161  
 Q\_OWA (*FuzzyOWL2Keyword* attribute), 195  
 QowaConcept (class in  
   *fuzzy\_dl\_owl2.fuzzycl.concept.qowa\_concept*), 61  
 QowaConcept (class in  
   *fuzzy\_dl\_owl2.fuzzycl.owl\_types.qowa\_concept*), 187  
 QsugenoConcept (class in *fuzzy\_dl\_owl2.fuzzycl.owl\_types.quasi\_sugeno\_concept*), 188  
 QsugenoIntegral (class in  
   *fuzzy\_dl\_owl2.fuzzycl.concept.quasi\_sugeno\_integral*), 62  
 QUANTIFIED\_OWA (*ConceptType* attribute), 154, 193  
 QUANTIFIER (*FuzzyOWL2Keyword* attribute), 195  
 quantifier (*QowaConcept* property), 62  
 QUASI\_SUGENO (*ConceptType* attribute), 193  
 QUASI\_SUGENO (*FuzzyDLKeyword* attribute), 161  
 QUASI\_SUGENO (*FuzzyOWL2Keyword* attribute), 195  
 QUASI\_SUGENO\_INTEGRAL (*ConceptType* attribute), 155  
 queries\_list (*DLParser* attribute), 135



Query (class in `fuzzy_dl_owl2.fuzzycl.query.query`), 146

## R

`r_successors` (KnowledgeBase attribute), 116  
`RANGE` (FuzzyDLKeyword attribute), 161  
`range_restrictions` (KnowledgeBase attribute), 116  
`RangeAxiom` (class in `fuzzy_dl_owl2.fuzzycl.range_axiom`), 148  
`read_object_from_file()` (KnowledgeBase method), 103  
`REAL` (ConcreteFeatureType attribute), 156  
`REAL` (FuzzyDLKeyword attribute), 161  
`recursion_unlimited()` (in module `fuzzy_dl_owl2.fuzzycl.util.utils`), 169  
`reduce_double_negation()` (OperatorConcept method), 60  
`reduce_double_negation()` (Thing method), 43  
`reduce_idempotency()` (AtomicConcept method), 40  
`reduce_idempotency()` (OperatorConcept method), 60  
`reduce_idempotency()` (Thing method), 43  
`reduce_quantifiers()` (OperatorConcept method), 60  
`reduce_quantifiers()` (Thing method), 43  
`reduce_truth_values()` (OperatorConcept method), 60  
`reduce_truth_values()` (Thing method), 43  
`REFLEXIVE` (FuzzyDLKeyword attribute), 161  
`reflexive_roles` (KnowledgeBase attribute), 116  
`RELATED` (FuzzyDLKeyword attribute), 161  
`RelatedQuery` (class in `fuzzy_dl_owl2.fuzzycl.query.related_query`), 147  
`Relation` (class in `fuzzy_dl_owl2.fuzzycl.relation`), 148  
`remove_A_is_a_B()` (KnowledgeBase method), 103  
`remove_A_is_a_C()` (KnowledgeBase method), 103  
`remove_A_is_a_X()` (KnowledgeBase method), 103  
`remove_C_is_a_A()` (KnowledgeBase method), 104  
`remove_C_is_a_D()` (KnowledgeBase method), 104  
`remove_C_is_a_X()` (KnowledgeBase method), 104  
`remove_element()` (Thing static method), 43  
`remove_input_edge()` (ClassificationNode method), 37  
`remove_output_edge()` (ClassificationNode method), 37  
`replace()` (AllSomeConcept method), 38  
`replace()` (ApproximationConcept method), 39  
`replace()` (AtomicConcept method), 40  
`replace()` (ChoquetIntegral method), 41  
`replace()` (ExtThresholdConcept method), 51  
`replace()` (FuzzyConcreteConcept method), 44  
`replace()` (HasValueConcept method), 51  
`replace()` (ImpliesConcept method), 52  
`replace()` (LinearlyModifiedConcept method), 55  
`replace()` (ModifiedConcept method), 56  
`replace()` (OperatorConcept method), 60  
`replace()` (OwaConcept method), 61  
`replace()` (QowaConcept method), 62  
`replace()` (QsugenIntegral method), 62  
`replace()` (SelfConcept method), 63  
`replace()` (SigmaConcept method), 64  
`replace()` (StringConcept method), 65  
`replace()` (SugenIntegral method), 66  
`replace()` (Thing method), 43  
`replace()` (ThresholdConcept method), 67  
`replace()` (TriangularlyModifiedConcept method), 57  
`replace()` (TruthConcept method), 68  
`replace()` (ValueConcept method), 69  
`replace()` (WeightedConcept method), 69  
`replace()` (WeightedMaxConcept method), 70  
`replace()` (WeightedMinConcept method), 71  
`replace()` (WeightedSumConcept method), 71  
`replace()` (WeightedSumZeroConcept method), 72  
`represent_tbox_with_gcis()` (KnowledgeBase method), 104  
`RepresentativeIndividual` (class in `fuzzy_dl_owl2.fuzzycl.individual.representative_individual`), 83  
`RepresentativeIndividualType` (class in `fuzzy_dl_owl2.fuzzycl.util.constants`), 166  
`restrict_range()` (KnowledgeBase method), 104  
`Restriction` (class in `fuzzy_dl_owl2.fuzzycl.restriction.restriction`), 149

`RestrictionType` (class in `fuzzy_dl_owl2.fuzzycl.util.constants`), 167  
`RESULTS_PATH` (in module `fuzzy_dl_owl2.fuzzycl.util.constants`), 168  
`RIGHT_SHOULDER` (FuzzyDLKeyword attribute), 161  
`RIGHT_SHOULDER` (FuzzyOWL2Keyword attribute), 195  
`RightConcreteConcept` (class in `fuzzy_dl_owl2.fuzzycl.concept.concrete.right_concrete_concept`), 48  
`RightShoulderFunction` (class in `fuzzy_dl_owl2.fuzzycl.owl_types.right_shoulder_function`), 188  
`role` (DomainAxiom attribute), 76  
`ROLE` (FuzzyOWL2Keyword attribute), 195  
`role` (HasRoleInterface property), 54  
`role` (MaxRelatedQuery attribute), 142  
`role` (MinRelatedQuery attribute), 145  
`role` (RangeAxiom attribute), 148  
`role` (RelatedQuery attribute), 147  
`role` (SigmaConcept attribute), 64  
`role` (SigmaCount attribute), 65  
`role_absorption()` (KnowledgeBase method), 104  
`role_domain()` (KnowledgeBase method), 104  
`role_implies()` (KnowledgeBase method), 104  
`role_is_functional()` (KnowledgeBase method), 104  
`role_is_inverse_functional()` (KnowledgeBase method), 104  
`role_is_reflexive()` (KnowledgeBase method), 104  
`role_is_symmetric()` (KnowledgeBase method), 104  
`role_is_transitive()` (KnowledgeBase method), 104  
`role_name` (Relation attribute), 149  
`role_name` (Restriction attribute), 150  
`role_range()` (KnowledgeBase method), 104  
`role_relations` (Individual attribute), 83  
`role_restrictions` (Individual attribute), 83  
`role_subsumes()` (KnowledgeBase method), 104  
`role_subsumes_bool()` (KnowledgeBase method), 105  
`RoleParentWithDegree` (class in `fuzzy_dl_owl2.fuzzycl.role_parent_with_degree`), 150  
`roles_with_all_parents` (KnowledgeBase attribute), 116  
`roles_with_parents` (KnowledgeBase attribute), 116  
`roles_with_trans_children` (KnowledgeBase attribute), 116  
`round()` (Util static method), 169  
`rule_acyclic_tbox` (KnowledgeBase attribute), 116  
`RULE_ACYCLIC_TBOXES` (ConfigReader attribute), 151  
`rule_all()` (KnowledgeBase method), 105  
`rule_and()` (KnowledgeBase method), 105  
`rule_ass_nom()` (KnowledgeBase method), 105  
`RULE_ATOMIC` (KnowledgeBaseRules attribute), 163  
`rule_atomic()` (KnowledgeBase method), 105  
`RULE_BOTTOM` (KnowledgeBaseRules attribute), 163  
`rule_bottom()` (KnowledgeBase method), 105  
`rule_choquet()` (KnowledgeBase method), 105  
`RULE_CHOQUET_INTEGRAL` (KnowledgeBaseRules attribute), 163  
`RULE_COMPLEMENT` (KnowledgeBaseRules attribute), 163  
`rule_complemented()` (KnowledgeBase method), 105  
`rule_complemented_at_least_datatype_restriction()` (KnowledgeBase method), 105  
`rule_complemented_at_most_datatype_restriction()` (KnowledgeBase method), 105  
`rule_complemented_atomic()` (KnowledgeBase method), 105  
`rule_complemented_choquet()` (KnowledgeBase method), 105  
`rule_complemented_complex_assertion()` (KnowledgeBase method), 105  
`rule_complemented_concrete()` (KnowledgeBase method), 105  
`rule_complemented_exact_datatype_restriction()` (KnowledgeBase method), 106  
`rule_complemented_extended_negative_threshold()` (KnowledgeBase method), 106  
`rule_complemented_extended_positive_threshold()` (KnowledgeBase method), 106  
`rule_complemented_fuzzy_number()` (KnowledgeBase method), 106  
`rule_complemented_goedel_implication()` (KnowledgeBase method), 106  
`rule_complemented_has_value()` (KnowledgeBase method), 106

- `rule_complemented_lazy_unfolding()` (*KnowledgeBase method*), 106
- `rule_complemented_modified()` (*KnowledgeBase method*), 106
- `rule_complemented_negative_threshold()` (*KnowledgeBase method*), 106
- `rule_complemented_owa()` (*KnowledgeBase method*), 106
- `rule_complemented_positive_threshold()` (*KnowledgeBase method*), 106
- `rule_complemented_quantified_owa()` (*KnowledgeBase method*), 106
- `rule_complemented_quasi_sugeno()` (*KnowledgeBase method*), 106
- `rule_complemented_self()` (*KnowledgeBase method*), 106
- `rule_complemented_sigma_concept()` (*KnowledgeBase method*), 106
- `rule_complemented_sugeno()` (*KnowledgeBase method*), 106
- `rule_complemented_weighted_concept()` (*KnowledgeBase method*), 106
- `rule_complemented_weighted_max()` (*KnowledgeBase method*), 106
- `rule_complemented_weighted_min()` (*KnowledgeBase method*), 106
- `rule_complemented_weighted_sum()` (*KnowledgeBase method*), 106
- `rule_complemented_weighted_sum_zero()` (*KnowledgeBase method*), 107
- `rule_complemented_zadeh_implication()` (*KnowledgeBase method*), 107
- `RULE_CONCRETE` (*KnowledgeBaseRules attribute*), 163
- `rule_concrete()` (*KnowledgeBase method*), 107
- `RULE_DATATYPE` (*KnowledgeBaseRules attribute*), 163
- `rule_domain_lazy_unfolding()` (*KnowledgeBase method*), 107
- `rule_extended_negative_threshold()` (*KnowledgeBase method*), 107
- `rule_extended_positive_threshold()` (*KnowledgeBase method*), 107
- `rule_feature_function()` (*DatatypeReasoner static method*), 88
- `RULE_FUZZY_NUMBER` (*KnowledgeBaseRules attribute*), 163
- `rule_fuzzy_number()` (*KnowledgeBase method*), 107
- `RULE_GOEDEL_ALL` (*KnowledgeBaseRules attribute*), 164
- `RULE_GOEDEL_AND` (*KnowledgeBaseRules attribute*), 164
- `rule_goedel_and()` (*KnowledgeBase method*), 107
- `rule_goedel_implication()` (*KnowledgeBase method*), 107
- `RULE_GOEDEL_IMPLIES` (*KnowledgeBaseRules attribute*), 164
- `RULE_GOEDEL_OR` (*KnowledgeBaseRules attribute*), 164
- `rule_goedel_or()` (*KnowledgeBase method*), 107
- `RULE_GOEDEL_SOME` (*KnowledgeBaseRules attribute*), 164
- `RULE_HAS_VALUE` (*KnowledgeBaseRules attribute*), 164
- `rule_has_value()` (*KnowledgeBase method*), 107
- `rule_lazy_unfolding()` (*KnowledgeBase method*), 107
- `rule_loose_lower_approximation()` (*KnowledgeBase method*), 107
- `rule_loose_upper_approximation()` (*KnowledgeBase method*), 107
- `rule_lower_approximation()` (*KnowledgeBase method*), 107
- `RULE_LUKASIEWICZ_ALL` (*KnowledgeBaseRules attribute*), 164
- `RULE_LUKASIEWICZ_AND` (*KnowledgeBaseRules attribute*), 164
- `rule_lukasiewicz_and()` (*KnowledgeBase method*), 107
- `RULE_LUKASIEWICZ_OR` (*KnowledgeBaseRules attribute*), 164
- `rule_lukasiewicz_or()` (*KnowledgeBase method*), 107
- `RULE_LUKASIEWICZ_SOME` (*KnowledgeBaseRules attribute*), 164
- `RULE_MODIFIED` (*KnowledgeBaseRules attribute*), 164
- `rule_modified()` (*KnowledgeBase method*), 107
- `rule_n2()` (*KnowledgeBase method*), 107
- `rule_n3()` (*KnowledgeBase method*), 107
- `rule_negative_threshold()` (*KnowledgeBase method*), 107
- `RULE_NOT_CHOQUET_INTEGRAL` (*KnowledgeBaseRules attribute*), 164
- `RULE_NOT_CONCRETE` (*KnowledgeBaseRules attribute*), 164
- `RULE_NOT_DATATYPE` (*KnowledgeBaseRules attribute*), 164
- `RULE_NOT_FUZZY_NUMBER` (*KnowledgeBaseRules attribute*), 164
- `RULE_NOT_GOEDEL_IMPLIES` (*KnowledgeBaseRules attribute*), 164
- `RULE_NOT_HAS_VALUE` (*KnowledgeBaseRules attribute*), 164
- `RULE_NOT_MODIFIED` (*KnowledgeBaseRules attribute*), 164
- `RULE_NOT_OWA` (*KnowledgeBaseRules attribute*), 164
- `RULE_NOT_QUASI_SUGENO_INTEGRAL` (*KnowledgeBaseRules attribute*), 164
- `RULE_NOT_SELF` (*KnowledgeBaseRules attribute*), 164
- `RULE_NOT_SIGMA_COUNT` (*KnowledgeBaseRules attribute*), 164
- `rule_not_simple_restriction()` (*DatatypeReasoner static method*), 88
- `RULE_NOT_SUGENO_INTEGRAL` (*KnowledgeBaseRules attribute*), 164
- `RULE_NOT_THRESHOLD` (*KnowledgeBaseRules attribute*), 164
- `rule_not_triangular_fuzzy_number()` (*DatatypeReasoner static method*), 89
- `RULE_NOT_W_MAX` (*KnowledgeBaseRules attribute*), 164
- `RULE_NOT_W_MIN` (*KnowledgeBaseRules attribute*), 164
- `RULE_NOT_W_SUM` (*KnowledgeBaseRules attribute*), 164
- `RULE_NOT_W_SUM_ZERO` (*KnowledgeBaseRules attribute*), 164
- `RULE_NOT_WEIGHTED` (*KnowledgeBaseRules attribute*), 164
- `RULE_NOT_ZADEH_IMPLIES` (*KnowledgeBaseRules attribute*), 164
- `rule_or()` (*KnowledgeBase method*), 107
- `RULE_OWA` (*KnowledgeBaseRules attribute*), 164
- `rule_owa()` (*KnowledgeBase method*), 107
- `rule_positive_threshold()` (*KnowledgeBase method*), 107
- `rule_quantified_owa()` (*KnowledgeBase method*), 107
- `rule_quasi_sugeno()` (*KnowledgeBase method*), 107
- `RULE_QUASI_SUGENO_INTEGRAL` (*KnowledgeBaseRules attribute*), 164
- `rule_range_lazy_unfolding()` (*KnowledgeBase method*), 107
- `RULE_SELF` (*KnowledgeBaseRules attribute*), 164
- `rule_self()` (*KnowledgeBase method*), 107
- `rule_sigma_concept()` (*KnowledgeBase method*), 108
- `RULE_SIGMA_COUNT` (*KnowledgeBaseRules attribute*), 165
- `rule_simple_restriction()` (*DatatypeReasoner static method*), 89
- `rule_some()` (*KnowledgeBase method*), 108
- `rule_sugeno()` (*KnowledgeBase method*), 108
- `RULE_SUGENO_INTEGRAL` (*KnowledgeBaseRules attribute*), 165
- `RULE_THRESHOLD` (*KnowledgeBaseRules attribute*), 165
- `rule_threshold_common()` (*KnowledgeBase method*), 108
- `rule_tight_lower_approximation()` (*KnowledgeBase method*), 108
- `rule_tight_upper_approximation()` (*KnowledgeBase method*), 108
- `RULE_TOP` (*KnowledgeBaseRules attribute*), 165
- `rule_top()` (*KnowledgeBase method*), 108
- `rule_triangular_fuzzy_number()` (*DatatypeReasoner static method*), 89
- `rule_upper_approximation()` (*KnowledgeBase method*), 108
- `RULE_W_MAX` (*KnowledgeBaseRules attribute*), 165
- `RULE_W_MIN` (*KnowledgeBaseRules attribute*), 165
- `RULE_W_SUM` (*KnowledgeBaseRules attribute*), 165
- `RULE_W_SUM_ZERO` (*KnowledgeBaseRules attribute*), 165
- `RULE_WEIGHTED` (*KnowledgeBaseRules attribute*), 165
- `rule_weighted_concept()` (*KnowledgeBase method*), 108
- `rule_weighted_max()` (*KnowledgeBase method*), 108
- `rule_weighted_min()` (*KnowledgeBase method*), 108
- `rule_weighted_sum()` (*KnowledgeBase method*), 108
- `rule_weighted_sum_zero()` (*KnowledgeBase method*), 108
- `rule_zadeh_implication()` (*KnowledgeBase method*), 108
- `RULE_ZADEH_IMPLIES` (*KnowledgeBaseRules attribute*), 165
- `rules_applied` (*KnowledgeBase attribute*), 116
- `run()` (*FuzzyDLToOwl2 method*), 79

## S

- `SAT_QUERY` (*FuzzyDLKeyword attribute*), 161
- `SatisfiableQuery` (*class* *fuzzy\_dl\_owl2.fuzzydl.query.satisfiable\_query*), 147
- `save_absorbed_tbox_to_file()` (*KnowledgeBase method*), 108
- `save_tbox_common_part_to_file()` (*KnowledgeBase method*), 108
- `save_tbox_to_file()` (*KnowledgeBase method*), 108
- `save_to_file()` (*KnowledgeBase method*), 108
- `SELF` (*ConceptType attribute*), 155

- SELF (FuzzyDLKeyword attribute), 161
- self() (SelfConcept static method), 63
- SelfConcept (class in fuzzy\_dl\_owl2.fuzzydl.concept.self\_concept), 63
- SEMI\_CONTINUOUS (VariableType attribute), 168
- SEPARATOR (in module fuzzy\_dl\_owl2.fuzzydl.util.constants), 168
- set\_binary\_variable() (Variable method), 131
- set\_binary\_variables() (MILPHelper method), 127
- SET\_BLOCKING (BlockingDynamicType attribute), 152
- set\_concrete\_individual() (CreatedIndividual method), 82
- set\_constant() (Expression method), 122
- set\_crisp\_concept() (KnowledgeBase method), 108
- set\_crisp\_role() (KnowledgeBase method), 108
- set\_datatype\_filler\_variable() (Variable method), 131
- set\_definition() (PrimitiveConceptDefinition method), 136
- set\_degree() (GeneralConceptInclusion method), 80
- set\_degree() (PrimitiveConceptDefinition method), 136
- set\_dynamic\_blocking() (KnowledgeBase method), 108
- set\_individual() (Assertion method), 36
- set\_initial\_time() (Query method), 146
- set\_label() (Individual method), 83
- set\_logic() (KnowledgeBase method), 108
- set\_lower\_limit() (Assertion method), 36
- set\_max\_value() (FuzzyDatatype method), 184
- set\_min\_value() (FuzzyDatatype method), 184
- set\_name() (Individual method), 83
- set\_name() (Modifier method), 133
- set\_name() (Variable method), 131
- set\_nominal\_variables() (MILPHelper method), 127
- set\_object\_individual() (Relation method), 148
- set\_range() (ConcreteFeature method), 73
- set\_range() (TriangularFuzzyNumber static method), 46
- set\_subject\_individual() (Relation method), 148
- set\_subsumed() (GeneralConceptInclusion method), 80
- set\_subsumer() (GeneralConceptInclusion method), 80
- set\_total\_time() (Query method), 146
- set\_truth\_constants() (KnowledgeBase method), 108
- set\_type() (ConcreteFeature method), 73
- set\_type() (Variable method), 131
- set\_unsatisfiable\_KB() (KnowledgeBase method), 108
- SHOW\_ABSTRACT\_FILLERS (FuzzyDLKeyword attribute), 161
- SHOW\_ABSTRACT\_FILLERS\_FOR (FuzzyDLKeyword attribute), 161
- show\_abstract\_role\_fillers() (ShowVariablesHelper method), 128
- SHOW\_CONCEPTS (FuzzyDLKeyword attribute), 161
- show\_concepts() (ShowVariablesHelper method), 128
- SHOW\_CONCRETE\_FILLERS (FuzzyDLKeyword attribute), 161
- show\_concrete\_fillers() (ShowVariablesHelper method), 129
- SHOW\_CONCRETE\_FILLERS\_FOR (FuzzyDLKeyword attribute), 161
- SHOW\_CONCRETE\_INSTANCE\_FOR (FuzzyDLKeyword attribute), 161
- show\_individuals() (ShowVariablesHelper method), 129
- SHOW\_INSTANCES (FuzzyDLKeyword attribute), 161
- SHOW\_LANGUAGE (FuzzyDLKeyword attribute), 161
- show\_language (KnowledgeBase attribute), 116
- show\_statistics() (KnowledgeBase method), 108
- show\_variable() (ShowVariablesHelper method), 129
- SHOW\_VARIABLES (FuzzyDLKeyword attribute), 161
- show\_vars (MILPHelper attribute), 127
- ShowVariablesHelper (class in fuzzy\_dl\_owl2.fuzzydl.milp.show\_variables\_helper), 128
- SIGMA\_CONCEPT (ConceptType attribute), 155
- SIGMA\_COUNT (FuzzyDLKeyword attribute), 161
- SigmaConcept (class in fuzzy\_dl\_owl2.fuzzydl.concept.sigma\_concept), 63
- SigmaCount (class in fuzzy\_dl\_owl2.fuzzydl.concept.sigma\_count), 64
- similarity\_relations (KnowledgeBase attribute), 116
- SINGLE\_CLOSE\_TAG (FuzzyOWL2Keyword attribute), 195
- SLASH (FuzzyOWL2Keyword attribute), 195
- Solution (class in fuzzy\_dl\_owl2.fuzzydl.milp.solution), 129
- solve() (AllInstancesQuery method), 137
- solve() (BnpQuery method), 137
- solve() (ClassificationQuery method), 138
- solve() (DefuzzifyQuery method), 138
- solve() (KbSatisfiableQuery method), 141
- solve() (MaxInstanceQuery method), 141
- solve() (MaxQuery method), 142
- solve() (MaxRelatedQuery method), 142
- solve() (MaxSatisfiableQuery method), 143
- solve() (MaxSubsumesQuery method), 143
- solve() (MinInstanceQuery method), 144
- solve() (MinQuery method), 144
- solve() (MinRelatedQuery method), 145
- solve() (MinSatisfiableQuery method), 145
- solve() (MinSubsumesQuery method), 146
- solve() (MomDefuzzifyQuery method), 139
- solve() (Query method), 146
- solve\_abox() (KnowledgeBase method), 108
- solve\_all() (ClassicalSolver static method), 84
- solve\_all() (LukasiewiczSolver static method), 117
- solve\_all() (ZadehSolver static method), 119
- solve\_and() (ClassicalSolver static method), 84
- solve\_and() (LukasiewiczSolver static method), 117
- solve\_and() (ZadehSolver static method), 120
- solve\_assertions() (KnowledgeBase method), 109
- solve\_cardinality\_list() (KnowledgeBase method), 109
- solve\_choquet\_integral\_assertion() (KnowledgeBase method), 109
- solve\_choquet\_integral\_complemented\_assertion() (KnowledgeBase method), 109
- solve\_concept\_assertion() (KnowledgeBase method), 109
- solve\_concept\_complemented\_assertion() (KnowledgeBase method), 109
- solve\_concrete\_value\_assertions() (KnowledgeBase method), 109
- solve\_crisp\_concrete\_concept\_assertion() (KnowledgeBase method), 109
- solve\_domain\_and\_range\_axioms() (KnowledgeBase method), 110
- solve\_functional\_roles() (KnowledgeBase method), 110
- solve\_fuzzy\_concrete\_concept\_complement\_assertion() (KnowledgeBase method), 110
- solve\_gci() (KnowledgeBase method), 110
- solve\_goedel\_gci() (KnowledgeBase method), 110
- solve\_gurobi() (MILPHelper method), 127
- solve\_inverse\_roles() (KnowledgeBase method), 110
- solve\_kb() (KnowledgeBase method), 110
- solve\_kleene\_dienes\_gci() (KnowledgeBase method), 110
- solve\_left\_concrete\_concept\_assertion() (KnowledgeBase method), 110
- solve\_linear\_concrete\_concept\_assertion() (KnowledgeBase method), 110
- solve\_linear\_modifier\_assertion() (KnowledgeBase method), 110
- solve\_lukasiewicz\_gci() (KnowledgeBase method), 110
- solve\_mip() (MILPHelper method), 127
- solve\_modifier\_assertion() (KnowledgeBase method), 110
- solve\_modifier\_complemented\_assertion() (KnowledgeBase method), 111
- solve\_new() (AllInstancesQuery method), 137
- solve\_not\_self\_rule() (IndividualHandler static method), 91
- solve\_one\_exist\_assertion() (KnowledgeBase method), 111
- solve\_or() (ClassicalSolver static method), 84
- solve\_or() (LukasiewiczSolver static method), 118
- solve\_or() (ZadehSolver static method), 120
- solve\_owa\_assertion() (KnowledgeBase method), 111
- solve\_owa\_complemented\_assertion() (KnowledgeBase method), 111
- solve\_pulp() (MILPHelper method), 127
- solve\_reflexive\_role() (KnowledgeBase method), 111
- solve\_reflexive\_roles() (KnowledgeBase method), 111
- solve\_relation\_restriction() (IndividualHandler static method), 91
- solve\_right\_concrete\_concept\_assertion() (KnowledgeBase method), 111
- solve\_role\_inclusion\_axioms() (KnowledgeBase method), 111



solve\_some() (*ClassicalSolver* static method), 84  
 solve\_some() (*LukasiewiczSolver* static method), 118  
 solve\_some() (*ZadehSolver* static method), 120  
 solve\_sugeno\_integral\_assertion() (*KnowledgeBase* method), 111  
 solve\_sugeno\_integral\_complemented\_assertion() (*KnowledgeBase* method), 111  
 solve\_trapezoidal\_concrete\_concept\_assertion() (*KnowledgeBase* method), 112  
 solve\_triangular\_concrete\_concept\_assertion() (*KnowledgeBase* method), 112  
 solve\_triangular\_modifier\_assertion() (*KnowledgeBase* method), 112  
 solve\_w\_max\_assertion() (*KnowledgeBase* method), 112  
 solve\_w\_max\_complemented\_assertion() (*KnowledgeBase* method), 112  
 solve\_w\_min\_assertion() (*KnowledgeBase* method), 112  
 solve\_w\_min\_complemented\_assertion() (*KnowledgeBase* method), 112  
 solve\_w\_sum\_assertion() (*KnowledgeBase* method), 112  
 solve\_w\_sum\_complemented\_assertion() (*KnowledgeBase* method), 112  
 solve\_w\_sum\_zero\_assertion() (*KnowledgeBase* method), 112  
 solve\_w\_sum\_zero\_complemented\_assertion() (*KnowledgeBase* method), 113  
 solve\_zadeh\_gci() (*KnowledgeBase* method), 113  
 SomDefuzzifyQuery (class in *fuzzy\_dl\_owl2.fuzzycl.query.defuzzify.som\_defuzzify\_query*), 140  
 SOME (*ConceptType* attribute), 155  
 SOME (*FuzzyDLKeyword* attribute), 161  
 Some (in module *fuzzy\_dl\_owl2.fuzzycl.concept.all\_some\_concept*), 38  
 some() (*AllSomeConcept* static method), 38  
 SPECIAL\_STRING (*Concept* attribute), 42  
 STAR\_SEPARATOR (in module *fuzzy\_dl\_owl2.fuzzycl.util.constants*), 168  
 STRING (*ConcreteFeatureType* attribute), 156  
 STRING (*FuzzyDLKeyword* attribute), 161  
 string\_datatypes (*FuzzyOwl2ToFuzzyDL* attribute), 182  
 string\_features (*MILPHelper* attribute), 127  
 string\_values (*MILPHelper* attribute), 127  
 StringConcept (class in *fuzzy\_dl\_owl2.fuzzycl.concept.string\_concept*), 65  
 SUB (*FuzzyDLKeyword* attribute), 161  
 SUBSET\_BLOCKING (*BlockingDynamicType* attribute), 152  
 subsumed (*GeneralConceptInclusion* attribute), 80  
 subsumer (*GeneralConceptInclusion* attribute), 81  
 subsumption\_flags (*KnowledgeBase* attribute), 116  
 SubsumptionQuery (class in *fuzzy\_dl\_owl2.fuzzycl.query.subsumption\_query*), 147  
 subtract\_expressions() (*Expression* static method), 122  
 subtract\_from\_expression() (*Degree* method), 74  
 subtract\_from\_expression() (*DegreeExpression* method), 75  
 subtract\_from\_expression() (*DegreeNumeric* method), 75  
 subtract\_from\_expression() (*DegreeVariable* method), 76  
 SUBTRACTION (*FeatureFunctionType* attribute), 157  
 SUGENO (*ConceptType* attribute), 193  
 SUGENO (*FuzzyDLKeyword* attribute), 162  
 SUGENO (*FuzzyOWL2Keyword* attribute), 196  
 SUGENO\_INTEGRAL (*ConceptType* attribute), 155  
 SugenoConcept (class in *fuzzy\_dl\_owl2.fuzzycl.owl\_types.sugeno\_concept*), 188  
 SugenoIntegral (class in *fuzzy\_dl\_owl2.fuzzycl.concept.sugeno\_integral*), 66  
 SUM (*FeatureFunctionType* attribute), 158  
 SUM (*FuzzyDLKeyword* attribute), 162  
 SYMMETRIC (*FuzzyDLKeyword* attribute), 162  
 symmetric\_roles (*KnowledgeBase* attribute), 116  
 synonym\_absorption\_A\_is\_a\_B() (*KnowledgeBase* method), 113  
 synonym\_absorption\_to\_do\_A\_is\_a\_B() (*KnowledgeBase* method), 113

## T

t\_definitions (*KnowledgeBase* attribute), 116  
 t\_disjoints (*KnowledgeBase* attribute), 116  
 t\_G (*KnowledgeBase* attribute), 116  
 t\_inclusions (*KnowledgeBase* attribute), 116  
 t\_synonyms (*KnowledgeBase* attribute), 116  
 temp\_relations\_list (*KnowledgeBase* attribute), 116  
 temp\_string\_concept\_list (*KnowledgeBase* attribute), 116  
 temp\_string\_list (*KnowledgeBase* attribute), 116  
 Term (class in *fuzzy\_dl\_owl2.fuzzycl.milp.term*), 130  
 Thing (class in *fuzzy\_dl\_owl2.fuzzycl.concept.concept*), 42  
 ThresholdConcept (class in *fuzzy\_dl\_owl2.fuzzycl.concept.threshold\_concept*), 66  
 TIGHT\_LOWER\_APPROX (*ConceptType* attribute), 155  
 tight\_lower\_approx() (*ApproximationConcept* static method), 39  
 TIGHT\_LOWER\_APPROXIMATION (*FuzzyDLKeyword* attribute), 162  
 TIGHT\_UPPER\_APPROX (*ConceptType* attribute), 155  
 tight\_upper\_approx() (*ApproximationConcept* static method), 39  
 TIGHT\_UPPER\_APPROXIMATION (*FuzzyDLKeyword* attribute), 162  
 TightLowerApprox (in module *fuzzy\_dl\_owl2.fuzzycl.concept.approximation\_concept*), 39  
 TightUpperApprox (in module *fuzzy\_dl\_owl2.fuzzycl.concept.approximation\_concept*), 39  
 times() (*TriangularFuzzyNumber* static method), 46  
 tmp\_features (*KnowledgeBase* attribute), 116  
 to\_all\_some\_concept() (*ApproximationConcept* method), 39  
 to\_expression() (*FeatureFunction* method), 78  
 to\_owl\_annotation() (*FuzzyclToOwl2* method), 79  
 to\_str() (*FuzzyXML* static method), 196  
 TODAY (in module *fuzzy\_dl\_owl2.fuzzycl.parser.dl\_parser*), 135  
 TODAY (in module *fuzzy\_dl\_owl2.fuzzycl.util.util*), 169  
 TOP (*ConceptType* attribute), 155  
 TOP (*FuzzyDLKeyword* attribute), 162  
 TOP (in module *fuzzy\_dl\_owl2.fuzzycl.concept.truth\_concept*), 68  
 total\_time (*Query* attribute), 146  
 TRANSITIVE (*FuzzyDLKeyword* attribute), 162  
 transitive\_roles (*KnowledgeBase* attribute), 116  
 translate\_owl2ontology() (*FuzzyOwl2* method), 172  
 TRAPEZOIDAL (*FuzzyDLKeyword* attribute), 162  
 TRAPEZOIDAL (*FuzzyOWL2Keyword* attribute), 196  
 TrapezoidalConcreteConcept (class in *fuzzy\_dl\_owl2.fuzzycl.concept.concrete.trapezoidal\_concrete\_concept*), 49  
 TrapezoidalFunction (class in *fuzzy\_dl\_owl2.fuzzycl.owl\_types.trapezoidal\_function*), 189  
 TRIANGULAR (*FuzzyDLKeyword* attribute), 162  
 TRIANGULAR (*FuzzyOWL2Keyword* attribute), 196  
 TRIANGULAR\_MODIFIER (*FuzzyDLKeyword* attribute), 162  
 TriangularConcreteConcept (class in *fuzzy\_dl\_owl2.fuzzycl.concept.concrete.triangular\_concrete\_concept*), 49  
 TriangularFunction (class in *fuzzy\_dl\_owl2.fuzzycl.owl\_types.triangular\_function*), 189  
 TriangularFuzzyNumber (class in *fuzzy\_dl\_owl2.fuzzycl.concept.concrete.fuzzy\_number.triangular\_fuzzy\_number*), 45  
 TriangularlyModifiedConcept (class in *fuzzy\_dl\_owl2.fuzzycl.concept.modified.triangularly\_modified\_concept*), 56  
 TriangularModifier (class in *fuzzy\_dl\_owl2.fuzzycl.modifier.triangular\_modifier*), 134  
 TriangularModifier (class in *fuzzy\_dl\_owl2.fuzzycl.owl\_types.triangular\_modifier*), 190  
 truth\_constants (*KnowledgeBase* attribute), 116  
 TruthConcept (class in *fuzzy\_dl\_owl2.fuzzycl.concept.truth\_concept*), 67  
 type (*Concept* property), 42

TYPE (FuzzyOWL2Keyword attribute), 196  
 type (GeneralConceptInclusion attribute), 81  
 type (Inequation attribute), 123  
 type (OperatorConcept attribute), 60  
 type (QowaConcept attribute), 62  
 type (QsugenoIntegral attribute), 62  
 type (RepresentativeIndividual attribute), 84  
 type (SubsumptionQuery attribute), 147  
 type (Variable attribute), 132

## U

unlock() (CreatedIndividualHandler static method), 87  
 unlock\_children() (KnowledgeBase method), 113  
 unlock\_directly\_blocked() (CreatedIndividualHandler static method), 87  
 unlock\_indirectly\_blocked() (CreatedIndividualHandler static method), 87  
 unlock\_individual() (KnowledgeBase method), 113  
 unlock\_pairwise() (CreatedIndividualHandler static method), 87  
 unlock\_simple() (IndividualHandler static method), 91  
 UNCHECKED (CreatedIndividualBlockingType attribute), 157  
 update\_role\_successors() (CreatedIndividualHandler static method), 87  
 UPPER\_APPROX (ConceptType attribute), 155  
 upper\_approx() (ApproximationConcept static method), 39  
 UPPER\_APPROXIMATION (FuzzyDLKeyword attribute), 162  
 upper\_bound (Variable attribute), 132  
 UpperApprox (in module  
   fuzzy\_dl\_owl2.fuzzydl.concept.approximation\_concept), 39  
 UpVar (in module fuzzy\_dl\_owl2.fuzzydl.milp.variable), 132  
 Util (class in fuzzy\_dl\_owl2.fuzzydl.util.util), 169

## V

value (DegreeNumeric attribute), 75  
 value (HasValueInterface property), 54  
 ValueConcept (class in  
   fuzzy\_dl\_owl2.fuzzydl.concept.value\_concept), 68  
 Variable (class in fuzzy\_dl\_owl2.fuzzydl.milp.variable), 131  
 variable (DegreeVariable attribute), 76  
 variable (SigmaCount attribute), 65  
 VARIABLE\_NAME (Variable attribute), 131  
 VARIABLE\_NUMBER (Variable attribute), 131  
 variables (MILPHelper attribute), 127  
 variables (ShowVariablesHelper attribute), 129  
 VariableType (class in fuzzy\_dl\_owl2.fuzzydl.util.constants), 168

## W

W\_MAX (ConceptType attribute), 155  
 W\_MAX (FuzzyDLKeyword attribute), 162  
 W\_MIN (ConceptType attribute), 155  
 W\_MIN (FuzzyDLKeyword attribute), 162  
 W\_SUM (ConceptType attribute), 155  
 W\_SUM (FuzzyDLKeyword attribute), 162  
 W\_SUM\_ZERO (ConceptType attribute), 155  
 W\_SUM\_ZERO (FuzzyDLKeyword attribute), 162  
 warning() (Util static method), 169  
 WEIGHT (FuzzyOWL2Keyword attribute), 196  
 weight (Label attribute), 121  
 weight (ThresholdConcept property), 67  
 weight (WeightedConcept property), 69  
 weight\_variable (ExtThresholdConcept property), 51  
 WEIGHTED (ConceptType attribute), 155  
 WEIGHTED (FuzzyOWL2Keyword attribute), 196  
 WEIGHTED\_CONCEPT (ConceptType attribute), 193  
 WEIGHTED\_MAX (ConceptType attribute), 193  
 WEIGHTED\_MAXIMUM (FuzzyOWL2Keyword attribute), 196  
 WEIGHTED\_MIN (ConceptType attribute), 193  
 WEIGHTED\_MINIMUM (FuzzyOWL2Keyword attribute), 196  
 WEIGHTED\_SUM (ConceptType attribute), 193

WEIGHTED\_SUM (FuzzyOWL2Keyword attribute), 196  
 WEIGHTED\_SUM\_ZERO (ConceptType attribute), 193  
 WEIGHTED\_SUMZERO (FuzzyOWL2Keyword attribute), 196  
 WeightedConcept (class in  
   fuzzy\_dl\_owl2.fuzzydl.concept.weighted\_concept), 69  
 WeightedConcept (class in  
   fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.weighted\_concept), 190  
 WeightedMaxConcept (class in  
   fuzzy\_dl\_owl2.fuzzydl.concept.weighted\_max\_concept), 70  
 WeightedMaxConcept (class in fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.weighted\_max\_concept), 190  
 WeightedMinConcept (class in  
   fuzzy\_dl\_owl2.fuzzydl.concept.weighted\_min\_concept), 70  
 WeightedMinConcept (class in fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.weighted\_min\_concept), 191  
 WeightedSumConcept (class in  
   fuzzy\_dl\_owl2.fuzzydl.concept.weighted\_sum\_concept), 71  
 WeightedSumConcept (class in fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.weighted\_sum\_concept), 191  
 WeightedSumZeroConcept (class in fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.weighted\_sum\_zero\_concept), 72  
 WeightedSumZeroConcept (class in fuzzy\_dl\_owl2.fuzzyowl2.owl\_types.weighted\_sum\_zero\_concept), 191  
 WEIGHTS (FuzzyOWL2Keyword attribute), 196  
 weights (HasWeightedConceptsInterface property), 55  
 weights\_equal() (Label static method), 120  
 write\_asymmetric\_object\_property\_axiom() (FuzzyOwl2 method), 172  
 write\_asymmetric\_object\_property\_axiom() (FuzzyOwl2ToFuzzyDL method), 178  
 write\_choquet\_concept\_definition() (FuzzyOwl2 method), 172  
 write\_choquet\_concept\_definition() (FuzzyOwl2ToFuzzyDL method), 178  
 write\_concept\_assertion\_axiom() (FuzzyOwl2 method), 172  
 write\_concept\_assertion\_axiom() (FuzzyOwl2ToFuzzyDL method), 178  
 write\_concept\_declaration() (FuzzyOwl2 method), 172  
 write\_concept\_declaration() (FuzzyOwl2ToFuzzyDL method), 178  
 write\_data\_property\_assertion\_axiom() (FuzzyOwl2 method), 172  
 write\_data\_property\_assertion\_axiom() (FuzzyOwl2ToFuzzyDL method), 178  
 write\_data\_property\_declaration() (FuzzyOwl2 method), 172  
 write\_data\_property\_declaration() (FuzzyOwl2ToFuzzyDL method), 178  
 write\_data\_property\_domain\_axiom() (FuzzyOwl2 method), 172  
 write\_data\_property\_domain\_axiom() (FuzzyOwl2ToFuzzyDL method), 179  
 write\_data\_property\_range\_axiom() (FuzzyOwl2 method), 172  
 write\_data\_property\_range\_axiom() (FuzzyOwl2ToFuzzyDL method), 179  
 write\_different\_individuals\_axiom() (FuzzyOwl2 method), 172  
 write\_different\_individuals\_axiom() (FuzzyOwl2ToFuzzyDL method), 179  
 write\_disjoint\_classes\_axiom() (FuzzyOwl2 method), 172  
 write\_disjoint\_classes\_axiom() (FuzzyOwl2ToFuzzyDL method), 179  
 write\_disjoint\_data\_properties\_axiom() (FuzzyOwl2 method), 172  
 write\_disjoint\_data\_properties\_axiom() (FuzzyOwl2ToFuzzyDL method), 179  
 write\_disjoint\_object\_properties\_axiom() (FuzzyOwl2 method), 173

`write_disjoint_object_properties_axiom()`  
(*FuzzyOwl2ToFuzzyDL method*), 179

`write_disjoint_union_axiom()` (*FuzzyOwl2 method*), 173

`write_disjoint_union_axiom()` (*FuzzyOwl2ToFuzzyDL method*), 179

`write_equivalent_classes_axiom()` (*FuzzyOwl2 method*), 173

`write_equivalent_classes_axiom()` (*FuzzyOwl2ToFuzzyDL method*), 179

`write_equivalent_data_properties_axiom()` (*FuzzyOwl2 method*), 173

`write_equivalent_data_properties_axiom()` (*FuzzyOwl2ToFuzzyDL method*), 179

`write_equivalent_object_properties_axiom()` (*FuzzyOwl2 method*), 173

`write_equivalent_object_properties_axiom()` (*FuzzyOwl2ToFuzzyDL method*), 179

`write_feature_equation()` (*DatatypeReasoner static method*), 89

`write_functional_data_property_axiom()` (*FuzzyOwl2 method*), 173

`write_functional_data_property_axiom()` (*FuzzyOwl2ToFuzzyDL method*), 179

`write_functional_object_property_axiom()` (*FuzzyOwl2 method*), 173

`write_functional_object_property_axiom()` (*FuzzyOwl2ToFuzzyDL method*), 179

`write_fuzzy_logic()` (*FuzzyOwl2 method*), 173

`write_fuzzy_logic()` (*FuzzyOwl2ToFuzzyDL method*), 179

`write_fuzzy_nominal_concept_definition()` (*FuzzyOwl2 method*), 173

`write_fuzzy_nominal_concept_definition()` (*FuzzyOwl2ToFuzzyDL method*), 179

`write_fuzzy_number_equation()` (*DatatypeReasoner static method*), 89

`write_inverse_functional_object_property_axiom()` (*FuzzyOwl2 method*), 173

`write_inverse_functional_object_property_axiom()` (*FuzzyOwl2ToFuzzyDL method*), 179

`write_inverse_object_property_axiom()` (*FuzzyOwl2 method*), 173

`write_inverse_object_property_axiom()` (*FuzzyOwl2ToFuzzyDL method*), 179

`write_irreflexive_object_property_axiom()` (*FuzzyOwl2 method*), 173

`write_irreflexive_object_property_axiom()` (*FuzzyOwl2ToFuzzyDL method*), 180

`write_left_shoulder_function_definition()` (*FuzzyOwl2 method*), 173

`write_left_shoulder_function_definition()` (*FuzzyOwl2ToFuzzyDL method*), 180

`write_linear_function_definition()` (*FuzzyOwl2 method*), 173

`write_linear_function_definition()` (*FuzzyOwl2ToFuzzyDL method*), 180

`write_linear_modifier_definition()` (*FuzzyOwl2 method*), 173

`write_linear_modifier_definition()` (*FuzzyOwl2ToFuzzyDL method*), 180

`write_modified_concept_definition()` (*FuzzyOwl2 method*), 173

`write_modified_concept_definition()` (*FuzzyOwl2ToFuzzyDL method*), 180

`write_modified_function_definition()` (*FuzzyOwl2 method*), 173

`write_modified_function_definition()` (*FuzzyOwl2ToFuzzyDL method*), 180

`write_modified_property_definition()` (*FuzzyOwl2 method*), 174

`write_modified_property_definition()` (*FuzzyOwl2ToFuzzyDL method*), 180

`write_negative_data_property_assertion_axiom()` (*FuzzyOwl2 method*), 174

`write_negative_data_property_assertion_axiom()` (*FuzzyOwl2ToFuzzyDL method*), 180

`write_negative_object_property_assertion_axiom()` (*FuzzyOwl2 method*), 174

`write_negative_object_property_assertion_axiom()` (*FuzzyOwl2ToFuzzyDL method*), 180

`write_not_feature_equation()` (*DatatypeReasoner static method*), 90

`write_not_fuzzy_number_equation()` (*DatatypeReasoner static method*), 90

`write_object_property_assertion_axiom()` (*FuzzyOwl2 method*), 174

`write_object_property_assertion_axiom()` (*FuzzyOwl2ToFuzzyDL method*), 180

`write_object_property_declaration()` (*FuzzyOwl2 method*), 174

`write_object_property_declaration()` (*FuzzyOwl2ToFuzzyDL method*), 180

`write_object_property_domain_axiom()` (*FuzzyOwl2 method*), 174

`write_object_property_domain_axiom()` (*FuzzyOwl2ToFuzzyDL method*), 180

`write_object_property_range_axiom()` (*FuzzyOwl2 method*), 174

`write_object_property_range_axiom()` (*FuzzyOwl2ToFuzzyDL method*), 180

`write_object_to_file()` (*KnowledgeBase method*), 113

`write_owa_concept_definition()` (*FuzzyOwl2 method*), 174

`write_owa_concept_definition()` (*FuzzyOwl2ToFuzzyDL method*), 180

`write_qowa_concept_definition()` (*FuzzyOwl2 method*), 174

`write_qowa_concept_definition()` (*FuzzyOwl2ToFuzzyDL method*), 181

`write_quasi_sugeno_concept_definition()` (*FuzzyOwl2 method*), 174

`write_quasi_sugeno_concept_definition()` (*FuzzyOwl2ToFuzzyDL method*), 181

`write_reflexive_object_property_axiom()` (*FuzzyOwl2 method*), 174

`write_reflexive_object_property_axiom()` (*FuzzyOwl2ToFuzzyDL method*), 181

`write_right_shoulder_function_definition()` (*FuzzyOwl2 method*), 174

`write_right_shoulder_function_definition()` (*FuzzyOwl2ToFuzzyDL method*), 181

`write_same_individual_axiom()` (*FuzzyOwl2 method*), 174

`write_same_individual_axiom()` (*FuzzyOwl2ToFuzzyDL method*), 181

`write_sub_data_property_of_axiom()` (*FuzzyOwl2 method*), 175

`write_sub_data_property_of_axiom()` (*FuzzyOwl2ToFuzzyDL method*), 181

`write_sub_object_property_of_axiom()` (*FuzzyOwl2 method*), 175

`write_sub_object_property_of_axiom()` (*FuzzyOwl2ToFuzzyDL method*), 181

`write_sub_property_chain_of_axiom()` (*FuzzyOwl2 method*), 175

`write_sub_property_chain_of_axiom()` (*FuzzyOwl2ToFuzzyDL method*), 181

`write_subclass_of_axiom()` (*FuzzyOwl2 method*), 175

`write_subclass_of_axiom()` (*FuzzyOwl2ToFuzzyDL method*), 181

`write_sugeno_concept_definition()` (*FuzzyOwl2 method*), 175

`write_sugeno_concept_definition()` (*FuzzyOwl2ToFuzzyDL method*), 181

`write_symmetric_object_property_axiom()` (*FuzzyOwl2 method*), 175

`write_symmetric_object_property_axiom()` (*FuzzyOwl2ToFuzzyDL method*), 181

`write_transitive_object_property_axiom()` (*FuzzyOwl2 method*), 175

[write\\_transitive\\_object\\_property\\_axiom\(\)](#)  
*(FuzzyOwl2ToFuzzyDL method), 181*  
[write\\_trapezoidal\\_function\\_definition\(\)](#) *(FuzzyOwl2 method), 175*  
[write\\_trapezoidal\\_function\\_definition\(\)](#)  
*(FuzzyOwl2ToFuzzyDL method), 181*  
[write\\_triangular\\_function\\_definition\(\)](#) *(FuzzyOwl2 method), 175*  
[write\\_triangular\\_function\\_definition\(\)](#)  
*(FuzzyOwl2ToFuzzyDL method), 181*  
[write\\_triangular\\_modifier\\_definition\(\)](#) *(FuzzyOwl2 method), 175*  
[write\\_triangular\\_modifier\\_definition\(\)](#)  
*(FuzzyOwl2ToFuzzyDL method), 182*  
[write\\_weighted\\_concept\\_definition\(\)](#) *(FuzzyOwl2 method), 175*  
[write\\_weighted\\_concept\\_definition\(\)](#)  
*(FuzzyOwl2ToFuzzyDL method), 182*  
[write\\_weighted\\_max\\_concept\\_definition\(\)](#) *(FuzzyOwl2 method), 175*  
[write\\_weighted\\_max\\_concept\\_definition\(\)](#)  
*(FuzzyOwl2ToFuzzyDL method), 182*  
[write\\_weighted\\_min\\_concept\\_definition\(\)](#) *(FuzzyOwl2 method), 175*  
[write\\_weighted\\_min\\_concept\\_definition\(\)](#)  
*(FuzzyOwl2ToFuzzyDL method), 182*  
[write\\_weighted\\_sum\\_concept\\_definition\(\)](#) *(FuzzyOwl2 method), 176*  
[write\\_weighted\\_sum\\_concept\\_definition\(\)](#)  
*(FuzzyOwl2ToFuzzyDL method), 182*  
[write\\_weighted\\_sum\\_zero\\_concept\\_definition\(\)](#)  
*(FuzzyOwl2 method), 176*  
[write\\_weighted\\_sum\\_zero\\_concept\\_definition\(\)](#)  
*(FuzzyOwl2ToFuzzyDL method), 182*

## X

[x\\_prime\\_individuals](#) *(KnowledgeBase attribute), 116*

## Y

[y\\_prime\\_individuals](#) *(KnowledgeBase attribute), 116*

## Z

[ZADEH](#) *(FuzzyDLKeyword attribute), 162*  
[ZADEH](#) *(FuzzyLogic attribute), 162*  
[ZADEH](#) *(FuzzyOWL2Keyword attribute), 196*  
[ZADEH](#) *(LogicOperatorType attribute), 166*  
[ZADEH\\_IMPLIES](#) *(ConceptType attribute), 155*  
[ZADEH\\_IMPLIES](#) *(FuzzyDLKeyword attribute), 162*  
[zadeh\\_implies\(\)](#) *(ImpliesConcept static method), 53*  
[zadeh\\_implies\(\)](#) *(KnowledgeBase method), 113*  
[zadeh\\_implies\\_equation\(\)](#) *(ZadehSolver static method), 120*  
[zadeh\\_implies\\_leq\\_equation\(\)](#) *(ZadehSolver static method), 120*  
[ZadehImplies](#) *(in module fuzzy\_dl\_owl2.fuzzydl.concept.implies\_concept), 53*  
[ZadehSolver](#) *(class in fuzzy\_dl\_owl2.fuzzydl.knowledge\_base), 118*