rlistic Documentation

Eduardo Rodriguez

Jun 08, 2025

CONTENTS:

Add your content using `reStructuredText` syntax. See the reStructuredText documentation for details.

CHAPTER

ONE

RLISTIC PACKAGE

1.1 Submodules

1.2 rlistic.common module

rlistic.common.**fuzzy_to_rl**(*fuzzy_set: dict*) → dict

Convert a fuzzy set to an RL-set using alpha-cuts.

Parameters
fuzzy_set (*dict*) – Dictionary mapping elements to membership degrees.

Returns
Dictionary mapping levels to alpha-cuts.

Return type
dict

rlistic.common.**rl_fuzzy_summary**(*rl: dict*) → dict

Compute a fuzzy summary of an RL, mapping elements to membership degrees.

Parameters
rl (*dict*) – Dictionary of an RL mapping levels to sets of elements (crisp realizations).

Returns
Fuzzy set representing the fuzzy summary of the RL as a sorted dictionary mapping elements to membership degrees, descending by degree, then element.

Return type
dict

Raises

- **TypeError** – If rl's keys are not a valid level-set per validate_level_set.
- **ValueError** – If level-set is invalid.

rlistic.common.**rl_input**(*input_file: str*, *process: Callable = None*) → dict

Read an RL from an input file, optionally process inputs with given function.

Parameters

- **input_file** (*str*) – Path to file with levels (space-separated) on first line, inputs on rest.
- **process** (*Callable, optional*) – Function that processes the input into the desired format.

Returns
Dictionary with the input RL.

Return type
dict

Raises
ValueError – If file not found, empty, has invalid levels, or wrong number of inputs.

rlistic.common.**rl_table**(*cls_name: str*, *mapping: dict*) → str

Generate a table representation of an RL.

Parameters

- **cls_name** (*str*) – Name of the class rlified by the RL.
- **mapping** (*dict*) – Dictionary mapping levels to objects.

Returns
Formatted table string with levels and objects.

Return type
str

rlistic.common.**validate_level_set**(*level_set: list[float]*) → None

Validate a level-set of an RL.

Parameters
level_set (*list[float]*) – List of levels of the RL.

Raises

- **TypeError** – If level_set is not a list or contains non-numeric values.
- **ValueError** – If level_set is empty, lacks 1, has levels outside (0,1], or is not in descending order.

rlistic.common.**validate_mapping**(*mapping: dict*) → None

Validate an RLs dictionary mapping of levels to objects.

Parameters
mapping (*dict*) – Dictionary mapping levels to objects.

Raises

- **TypeError** – If mapping is not a dict or objects are not of the same type.
- **ValueError** – If the level-set (keys) is invalid per validate_level_set.

1.3 rlistic.proxy module

class rlistic.proxy.**RL**(*mapping: dict*)

Bases: object

Proxy class for managing graduality using Representations by Levels (RLs).

__instance_class

Class of objects affected by graduality.

Type
type

__mapping

Dictionary mapping levels (floats in (0,1]) to objects.

Type
dict

`general_method`(*method_name: str*, **args*, ***kwargs*)

Apply a method level-wise across RLs, aggregating the result in an RL. Arguments can be RLs or crisp, the latter are extended to other levels. The output is an RL or crisp, if the objects on all levels are equal.

Parameters

- **method_name** (*str*) – Name of the method to apply (e.g., '__add__', 'union').
- ***args** – Positional arguments, may include RLs or crisp objects.
- ****kwargs** – Keyword arguments, may include RLs or crisp objects.

Returns
Resulting RL or crisp object (if same object on all levels).

Return type
RL or object

`get_level_set`() → list[float]

Get the level-set of the RL.

Returns
Level-set.

Return type
list[float]

`get_object`(*level*, *default=None*)

Get the object at a given level or a default value.

Parameters

- **level** (*float*) – Level to query.
- **default** – Value to return if level is not found (default: None).

Returns
Object at the level or default value.

Return type
object

`property instance_class: type`

Get the class of objects wrapped in the RL.

Returns
Class of the RL's objects.

Return type
type

`property mapping: dict`

Get the internal RL mapping.

Returns
Mapping of levels to objects.

Return type
dict

class rlistic.proxy.**RLMeta**(*name*, *bases*, *attrs*)

Bases: type

Metaclass for RL class to dynamically add dunder methods to the RL class's definition when creating it.

static **make_special_method**(*method_name: str*) → Callable

Factory for producing dunder methods for the RL class. The dunder methods delegate to the RL's general_method, same as regular methods.

Parameters
method_name (*str*) – Name of the special method (e.g., '__add__').

Returns
Method that calls general_method with the given method_name.

Return type
Callable

rlistic.proxy.**add_magic_methods**(*method_names: list[str]*) → None

Add magic methods to the RL class dynamically.

Parameters
method_names – List of magic method names (e.g., ['__eq__', '__lt__']).

Raises

- **ValueError** – If any method name is invalid (not starting/ending with '__').
- **TypeError** – If method_names is not a list or contains non-string elements.

1.4 rlistic.rlprogram module

rlistic.rlprogram.**rlify_program**(*command: list[str]*, *input_rl: dict*, *nproc: int = -1*) → dict

Run a program for each level of the RL, aggregate the output in a table representing the RL.

Parameters

- **command** (*list[str]*) – Command list (e.g., ['python', 'program.py']).
- **input_rl** (*dict*) – Input RL as a dictionary mapping levels to input strings.
- **nproc** (*int*) – Number of processes (-1: all CPUs, 1: sequential, >1: specific count).

Returns
Dictionary with the output RL.

Return type
dict

rlistic.rlprogram.**run_program**(*command: list[str]*, *inp: str*) → str

Run a program with input, return output

Parameters

- **command** (*list[str]*) – Command list (e.g., ['python', 'program.py']).
- **inp** (*str*) – Input to the program.

Returns
Output captured from the executed program.

Return type
str

1.5 rlistic.runtime module

rlistic.runtime.**rlify**(*original_class: type*) → type

Funcittion that rlifies a given class.

Parameters
original_class (*type*) – Class to be rlified.

Returns
Corresponding RL class.

Return type
type

1.6 rlistic.static module

rlistic.static.**rlify_class**(*node: ClassDef*) → tuple[str, str]

Generate code for the RL class from a class definition node using AST, copying the original class's method signatures and delegating to general_method in the body of the RL class's methods.

Parameters
node (*ast.ClassDef*) – node of the original class.

Returns
Code of the RL class. str: Name of the original class.

Return type
str

rlistic.static.**transform_file**(*input_path: str*, *output_path: str*) → None

Transform classes in input file to RL classes and write to output file.

Parameters

- **input_path** (*str*) – Path to input file with original classes.
- **output_path** (*str*) – Path to output file where RL classes will be written.

1.7 Module contents

CHAPTER

TWO

RLISTIC

PYTHON MODULE INDEX

r