FuzzydlToOwl2

concepts : dict[str, OWLClassExpression]
datatypes : dict[str, OWLDatatype]
fuzzyLabel : OWLAnnotationProperty
input_FDL : str
kb : NoneType
modifiers : dict[str, OWLDatatype]
num_classes : int
ontology : OWLOntology
ontology_iri : IRI
ontology_path : str
output_FOWL : str

__get_class_1(name: str): OWLClassExpression
__get_class_2(c: Concept): OWLClassExpression
__get_class_q_owa(c: Concept): OWLClassExpression
__get_class_weighted(c: Concept): OWLClassExpression
__get_class_weighted_min_max_sum(c: Concept): OWLClassExpression
__init__(input_file: str, output_file: str, base_iri: str): None
_get_concrete_concept_specifics(c: FuzzyConcreteConcept): tuple[str, dict[str, str]]
_process_assertion(ass: Assertion): None
_process_concrete_concept(c: FuzzyConcreteConcept): None
_process_individual(ind: Individual): None
_process_modifier(mod: Modifier): None
add_entity_annotation(annotation: str, entity: OWLEntity): None
add_ontology_annotation(annotation: str): None
annotate_gci(gci: GeneralConceptInclusion): None
annotate_pcd(c1: OWLClassExpression, pcd: PrimitiveConceptDefinition): None
annotation_property_iri(o: object): IRI
class_iri(o: object): IRI
data_property_iri(o: object): IRI
datatype_iri(o: object): IRI
exist_data_property(role: str): bool
exist_object_property(role: str): bool
get_annotations_for_axiom(value: typing.Union[float, DegreeNumeric]): set[OWLAnnotation]
get_base(c: Concept): OWLClassExpression
get_class(name: str): OWLClassExpression
get_data_property(role: str): typing.Union[OWLDataProperty, OWLObjectProperty]
get_individual(name: str): OWLNamedIndividual
get_new_atomic_class(name: str): OWLClassExpression
get_object_property(role: str): typing.Union[OWLDataProperty, OWLObjectProperty]
individual_iri(o: object): IRI
iri(o: object, iri_type: type): IRI
object_property_iri(o: object): IRI
run(): None
to_owl_annotation(annotation: str): OWLAnnotation