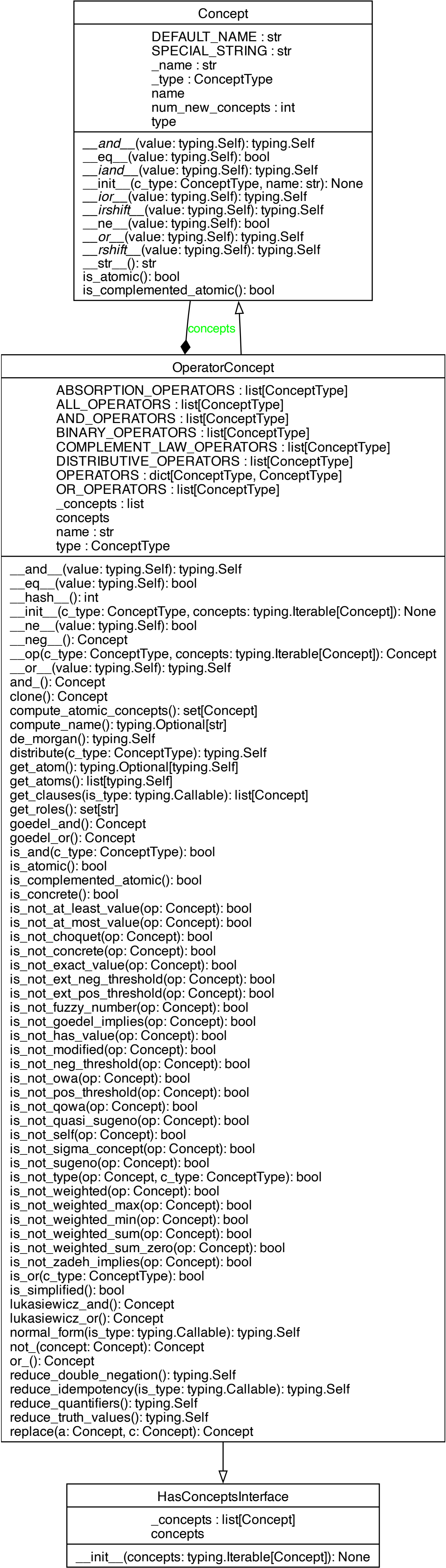
Concept
DEFAULT_NAME : str
SPECIAL_STRING : str
_name : str
_type : ConceptType
name
num_new_concepts : int
type
__and__(value: typing.Self): typing.Self
__eq__(value: typing.Self): bool
__iand__(value: typing.Self): typing.Self
__init__(c_type: ConceptType, name: str): None
__ior__(value: typing.Self): typing.Self
__irshift__(value: typing.Self): typing.Self
__ne__(value: typing.Self): bool
__or__(value: typing.Self): typing.Self
__rshift__(value: typing.Self): typing.Self
__str__(): str
is_atomic(): bool
is_complemented_atomic(): bool
concepts
OperatorConcept
ABSORPTION_OPERATORS : list[ConceptType]
ALL_OPERATORS : list[ConceptType]
AND_OPERATORS : list[ConceptType]
BINARY_OPERATORS : list[ConceptType]
COMPLEMENT_LAW_OPERATORS : list[ConceptType]
DISTRIBUTIVE_OPERATORS : list[ConceptType]
OPERATORS : dict[ConceptType, ConceptType]
OR_OPERATORS : list[ConceptType]
_concepts : list
concepts
name : str
type : ConceptType
__and__(value: typing.Self): typing.Self
__eq__(value: typing.Self): bool
__hash__(): int
__init__(c_type: ConceptType, concepts: typing.Iterable[Concept]): None
__ne__(value: typing.Self): bool
__neg__(): Concept
__op(c_type: ConceptType, concepts: typing.Iterable[Concept]): Concept
__or__(value: typing.Self): typing.Self
and_(): Concept
clone(): Concept
compute_atomic_concepts(): set[Concept]
compute_name(): typing.Optional[str]
de_morgan(): typing.Self
distribute(c_type: ConceptType): typing.Self
get_atom(): typing.Optional[typing.Self]
get_atoms(): list[typing.Self]
get_clauses(is_type: typing.Callable): list[Concept]
get_roles(): set[str]
goedel_and(): Concept
goedel_or(): Concept
is_and(c_type: ConceptType): bool
is_atomic(): bool
is_complemented_atomic(): bool
is_concrete(): bool
is_not_at_least_value(op: Concept): bool
is_not_at_most_value(op: Concept): bool
is_not_choquet(op: Concept): bool
is_not_concrete(op: Concept): bool
is_not_exact_value(op: Concept): bool
is_not_ext_neg_threshold(op: Concept): bool
is_not_ext_pos_threshold(op: Concept): bool
is_not_fuzzy_number(op: Concept): bool
is_not_goedel_implies(op: Concept): bool
is_not_has_value(op: Concept): bool
is_not_modified(op: Concept): bool
is_not_neg_threshold(op: Concept): bool
is_not_owa(op: Concept): bool
is_not_pos_threshold(op: Concept): bool
is_not_qowa(op: Concept): bool
is_not_quasi_sugeno(op: Concept): bool
is_not_self(op: Concept): bool
is_not_sigma_concept(op: Concept): bool
is_not_sugeno(op: Concept): bool
is_not_type(op: Concept, c_type: ConceptType): bool
is_not_weighted(op: Concept): bool
is_not_weighted_max(op: Concept): bool
is_not_weighted_min(op: Concept): bool
is_not_weighted_sum(op: Concept): bool
is_not_weighted_sum_zero(op: Concept): bool
is_not_zadeh_implies(op: Concept): bool
is_or(c_type: ConceptType): bool
is_simplified(): bool
lukasiewicz_and(): Concept
lukasiewicz_or(): Concept
normal_form(is_type: typing.Callable): typing.Self
not_(concept: Concept): Concept
or_(): Concept
reduce_double_negation(): typing.Self
reduce_idempotency(is_type: typing.Callable): typing.Self
reduce_quantifiers(): typing.Self
reduce_truth_values(): typing.Self
replace(a: Concept, c: Concept): Concept
HasConceptsInterface
_concepts : list[Concept]
concepts
__init__(concepts: typing.Iterable[Concept]): None