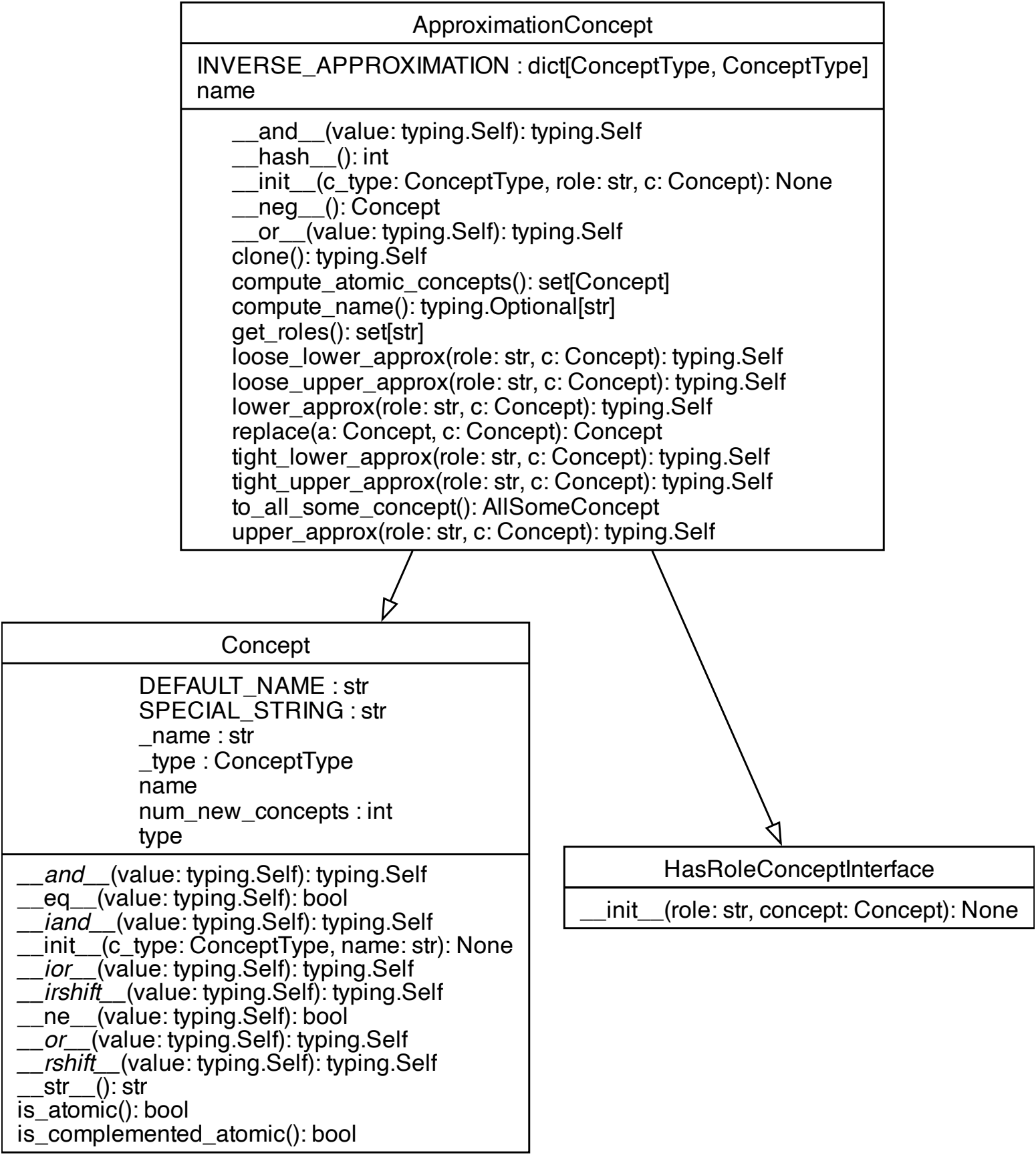
ApproximationConcept
INVERSE_APPROXIMATION : dict[ConceptType, ConceptType]
name
__and__(value: typing.Self): typing.Self
__hash__(): int
__init__(c_type: ConceptType, role: str, c: Concept): None
__neg__(): Concept
__or__(value: typing.Self): typing.Self
clone(): typing.Self
compute_atomic_concepts(): set[Concept]
compute_name(): typing.Optional[str]
get_roles(): set[str]
loose_lower_approx(role: str, c: Concept): typing.Self
loose_upper_approx(role: str, c: Concept): typing.Self
lower_approx(role: str, c: Concept): typing.Self
replace(a: Concept, c: Concept): Concept
tight_lower_approx(role: str, c: Concept): typing.Self
tight_upper_approx(role: str, c: Concept): typing.Self
to_all_some_concept(): AllSomeConcept
upper_approx(role: str, c: Concept): typing.Self
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
HasRoleConceptInterface
__init__(role: str, concept: Concept): None