UNIVERSITY OF TRENTO - Italy

Dipartimento di Ingegneria e Scienza dell'Informazione

Know dive

LODE - The logic of Knowledge Bases (KBs)

Sept 13,2023

LODE - The logic of Knowledge bases

- Introduction
- LoDE definition
- Language
- Domain
- Interpretation function
- Entailment
- Expansion
- LoDe Reasoning
- LOD to LOE reasoning
- Entailment properties

Knowledge Bases (KB's)

The original use of the term knowledge base was to describe one of the two sub-systems of an expert system (a knowledge-based system).

A knowledge-based system [*] consists of

- a knowledge-base representing *facts* about the world and
- ways of reasoning about those *facts* to deduce new facts or highlight inconsistencies.

[*] Hayes-Roth, Frederick; Donald Waterman; Douglas Lenat (1983). Building Expert Systems. Addison-Wesley.

KB - Components

- The language (concepts) used to name the entities and properties which are the case in the world (concept definitions)

- The knowledge (complex assertions) used to describe the properties of the entities as we perceive them in the world

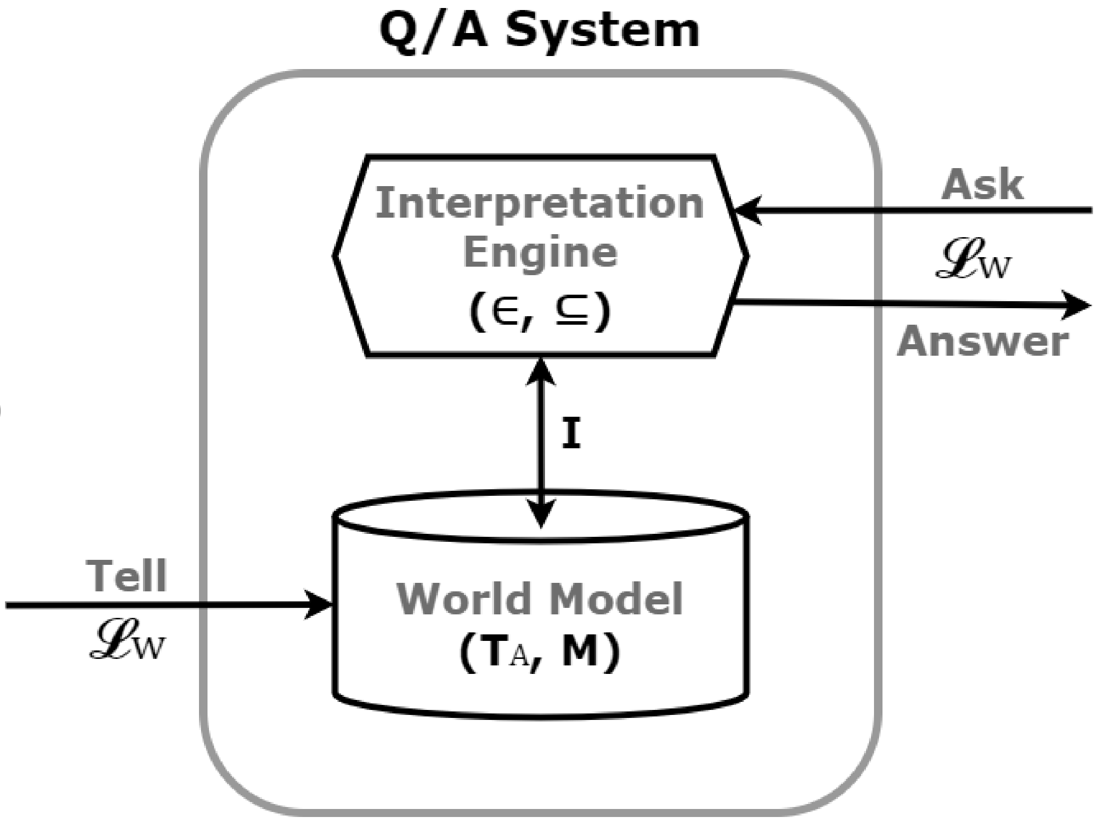
- The names of the entities which are the case in the world

- The (data and object) properties which describe how entities correlate in the world

- The possibility to query it an get answers back

KB = Reasoning as Question Answering

Tell Language (basic LODE)
+
Ask Language (basic LODE)



KB – Informal definition

KB = <***T***,***A***>

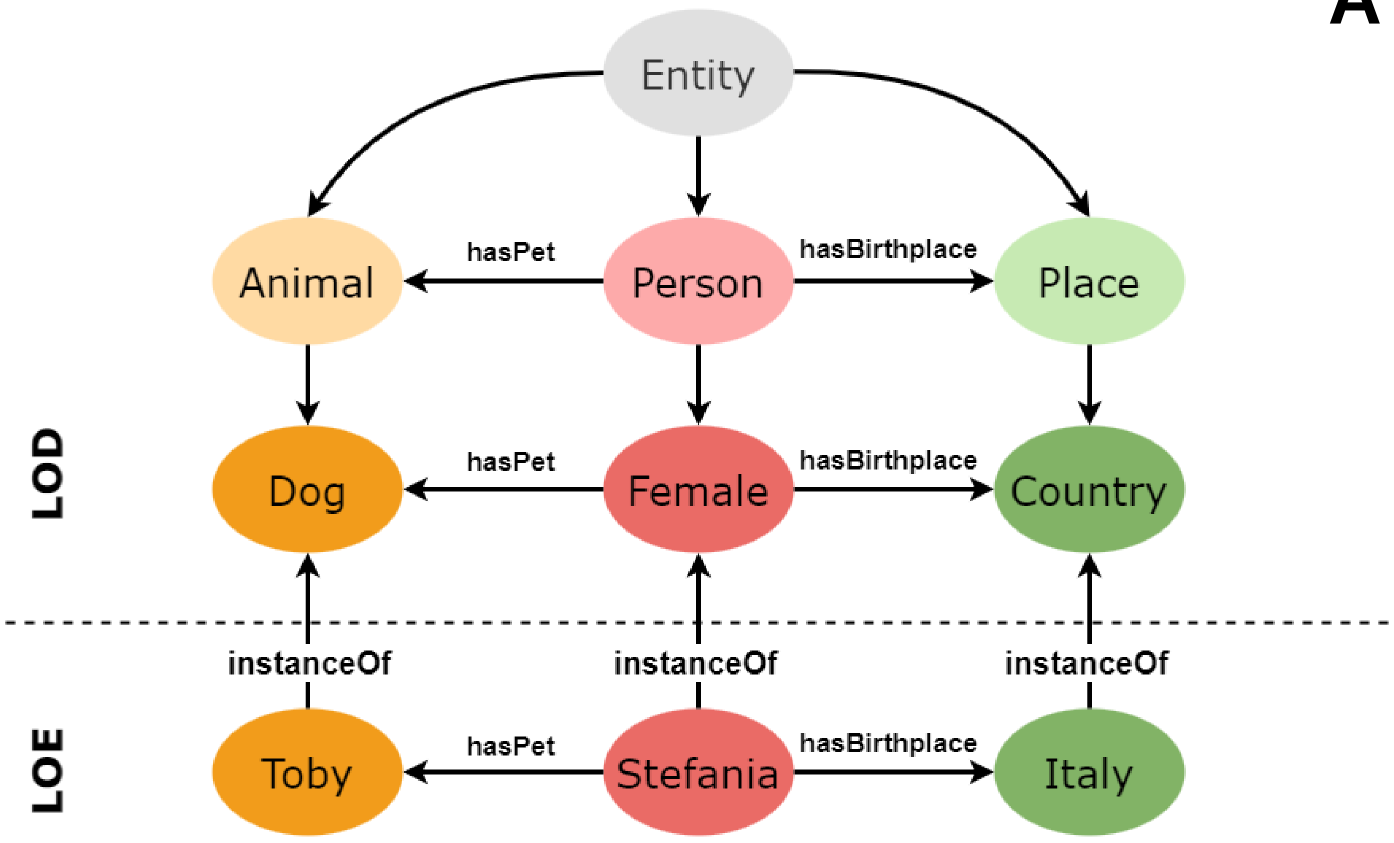
where:

- ***T*** (for ***T***erminology) is a theory which models the background general (commonsense, scientific, …) knowledge of the world. We formalize ***T*** in LOD.
- ***A*** (for ***A***ssertions) is a theory which models our knowledge of the world as we perceive it, we are told about it, we infer about i. We formalize ***A*** in LOE.

LOD and LOE provide, respectively, the means for reasoning about ***T*** and ***A*** independently.

LODE provide the means for reasoning about ***A*** based on the background knowledge encoded in ***T***.

KB Components – an example



A populated (knowledge) teleontology

- *Which LoD definitions?*
- *Which LoD descriptions?*
- *Which LoE assertions?*

Could we have an EG encoding the following assertions?

- Person (Lucia)
- HasBirthplace (person#1, Italy)
- Person(Stefania)
- HasHusband(Stefania, Mario)
- HasHusband(Lucia, Mario)

LODE - The logic of Knowledge bases

- Introduction
- LoDE definition
- Language
- Domain
- Interpretation function
- Entailment
- Expansion
- LoDe Reasoning
- LOD to LOE reasoning
- Entailment properties

LoDE – Definition

We formally define LODE as follows

$$\mathbf{LODE} = \langle \mathbf{EG}_{\mathbf{LODE}}, \models_{\mathbf{LODE}} \rangle$$

with

$$\mathbf{EG}_{\mathbf{LODE}} = \langle \mathbf{L}_{\mathbf{LODE}}, \mathbf{D}_{\mathbf{LODE}}, \mathbf{I}_{\mathbf{LODE}} \rangle$$

Below, any time no confusion arises, we drop the subscripts.

LoDE – Definition (continued)

$$EG_{LODE} = \langle L_{LODE}, D_{LODE}, I_{LODE} \rangle$$

is not given but it must be constructed by suitably integrating

$$EG_{LOE} = \langle L_{LOE}, D_{LOE}, I_{LOE} \rangle$$

and

$$ETG_{LOD} = \langle L_{LOD}, D_{LOD}, I_{LOD} \rangle$$

under the assumptions

$$D_{LOE} = D_{LOD} = D_{LODE}$$

LODE - The logic of Knowledge bases

- Introduction
- LoDE definition
- Language
- Domain
- Interpretation function
- Entailment
- Expansion
- LoDe Reasoning
- LOD to LOE reasoning
- Entailment properties

LoDE – The language of assertions A

Definition 10.6 (Language L**)**

$$\mathrm{L} = \mathrm{L}\mathcal{A} \cup \emptyset$$

where L$\mathcal{A}$ is a set of assertions. L$\mathcal{A}$ is constructed from an alphabet Aa defined as follows.

Definition 10.7 (Alphabet A**)**

$$\mathrm{A}a =< E, \{\mathrm{C}\}, \{\mathrm{P}\} >$$

where E is a set of **(names of) entities** e and of values v, {C}= ET ∪ DT ∪ L is a set of (names of) etypes ET, of dtypes and of labels L (i.e., defined etypes), {P} is a set of **properties**, also called **roles**, with {P} = {OP} ∪ {DP}, where OP is an **object property** and DP is a **data property**.

L$\mathcal{A}$ is a set of assertions, constructed from the alphabet Aa.

LoDE – the production rules for A

<awff> ::= <etype>(<nameEntity>) |
<dtype>(<value>) |
<objProp>(<nameEntity>, <nameEntity>) |
<dataProp>(<nameEntity>, <value>)

<etype> ::= ET1 | . . . | ETn | <label>

<dtype> ::= DT1 | . . . | DTn

<objProp> ::= OP1 | . . . | OPn

<dataProp> ::= DP1 | . . . | DPn

<nameEntity> ::= $e1$ | . . . | en

<value> ::= $v1$ | . . . | vn

Observation: Compare with LoE, we are building an EG with (new!) defined etypes <label>

LoDE – The language of the terminology T

Definition 11.4 (The language L)

$$\mathrm{L} = \mathrm{L}a \cup \mathrm{L}c$$

with

$$\mathrm{L}a = \mathrm{L}A \cup \mathrm{L}Ac$$

Observation 11.2 (LOE ***versus*** LOD**)** LOE allows for atomic assertions. LOD allows for (different) **atomic assertions** (L*A)*, for **complex assertions** (L*Ac*) and also **complex formulas** (L*c*), but only at the knowledge level. LoDE allows for LOE atomic assertions using etypes defined via LOD complex formulas

LoDE – the production rules for T

<cwff>	::= <label> ≡ <genus> ⊓ <differentia>	% A concept definition / description
<genus>	::= < label > | etype	% Etype(s) root(s) of the subsumption hierarchy
<differentia>	::= < awff >	
<cwff>	::= < label > ⊑ ¬ < label >	% A disjointness constraint
<awff>	::= <assertion> | ¬ <assertion>| <awff> ⊓ <awff>	% Atomic formulas
<assertion>	::= ⊤ | ⊥ | <etype> | <dtype> | ∃<objProp>.<etype> | ∃<dataProp>.<dtype>| ∀<objProp>.<etype> | ∀<dataProp>.<dtype>	% (Atomic) assertions, NO labels here!
<label>	::= DET1 | . . . | DETn |	% DETi is defined etype/concept

Observation 1 : Compare with LoD, we are building a teleontology

Observation 2 : <label> connects LOD to LOE in LoDE

LODE - The logic of Knowledge bases

- Introduction
- LoDE definition
- Language
- Domain
- Interpretation function
- Entailment
- Expansion
- Reasoning problems
- Entailment properties

LoDE – Domain

Definition (LoE/ LoD/ LoDE Domain)

$$D = < E, \{C\}, \{R\} >$$

with:

$$E = \{e\} \cup \{v\}$$

$$\{C\} = ET \cup DT \cup DET$$

$$\{R\} = \{OR\} \cup \{DR\}$$

where:

- E is a set of **entities** and **values**,
- $ET = \{E_T\}$, $E_T = \{e\}$ and $DT = \{D_T\}$, $D_T = \{v\}$, $DET = \{DE_T\}$, are **sets of etypes, dtypes**, and **defined etypes**, respectively
- OR, DR are **(binary) object** and **data relations**.

LODE - The logic of Knowledge bases

- Introduction
- LoDE definition
- Language
- Domain
- Interpretation function
- Entailment
- Expansion
- LoDe Reasoning
- LOD to LOE reasoning
- Entailment properties

LoDE – Interpretation function

Definition (Interpretation function I_{LODE}):

$$I_{LODE} = \langle I_{LOD}, I_{LOE} \rangle$$

Observation: I_{LOD} is first applied to compute the extension the of defined etypes, which are then used to compute the semantics of assertions (via I_{LOE}). This is a two step process:

1. Compute the extension of LOD complex assertions
2. Compute the extension of LOD (atomic) assertions (which are used to compute the semantics of LOE EG assertions)

LODE - The logic of Knowledge bases

- Introduction
- LoDE definition
- Language
- Domain
- Interpretation function
- Entailment
- Expansion
- LoDe Reasoning
- LOD to LOE reasoning
- Entailment properties

Entailment relation

Definition (Entailment |=)

$$\mathrm{M} \models a \Longleftrightarrow I\,(a) \in \mathrm{M}$$

with $a \in \mathrm{L}A$

Observation: The same of LoE, but (!) on an EG hugely extended with defined etypes (denoted by labels)

LODE - The logic of Knowledge bases

- Introduction
- LoDE definition
- Language
- Domain
- Interpretation function
- Entailment
- Expansion
- LoDe Reasoning
- LOD to LOE reasoning
- Entailment properties

Concept expansion

Definition (Concept expansion) Let T be an unfolded LoDE definitional TBox and A a LoDE ABox. Let C be a concept defined in T as $C \equiv C'$. Let a be an individual (entity) occurring in A. Assume $C(a)$ occurs in A. Then **the expansion exp(C(a)) of $C(a)$** in A in T is defined as follows:

$$\exp(C(a)) = \{C(a), \exp(C_1(a)), \ldots, \exp(C_n(a))\}$$

where C is defined as

$$C \equiv C_1 \sqcap \ldots \sqcap C_n$$

where $C_1, \ldots, C_n$ are the assertions occurring as conjuncts in C'.

Concept expansion (continued)

Definition (Conjunct expansion) Let $C_i(a)$, be an assertion in A which is a conjunct in the expansion of C as from the definition $C \equiv C'$. Then **the expansion exp($C_i(a)$) of $C_i(a)$** is defined as follows

- If C_l is an etype C, then

$$\exp(C_i(a)) = C(a)$$

- If C_l is of the form $\exists$P.C, then

$$\exp(C_i(a)) = P(a_1,a_2)$$

  with a_2 new in A.

- If C_l is of the form $\forall$P.C, with P(a_1,a_2) then

$$\exp(C_i(a)) = C(a_2)$$

Concept expansion (observations)

Observation 1: The expansion of a defined etype applied to an individual allows to unfold it into multiple independent assertions, one per conjunct.

Observation 2: The expansion of conjunct consisting of an etype applied to an individual allows to infer a new etype for that individual (for instance, from woman(e) to Mother(e))

Observation 3: The expansion of conjunct consisting of an existential quantification allows to create a new link to an individual which is anonymous (in the sense that it can be any of the known individuals) (for instance, from ∃HasChild.Person(e1) to HasChild(e1, anonymous#1) with anonymous#1 not occuring in the EG) .

Observation 4: The expansion of conjunct consisting of a universal quantification allows to know the etype of the individual in the codomain of the link) (for instance, from ∀HasChild.Person(e1) to Person(e1) with e1 occuring in the EG) .

Expansion (main result)

Definition (ABox expansion wrt a TBox). Apply concept expansion to all occurrences of all concepts occurring in the ABox and defined in the terminology T' obtained by unfolding the TBox T. The resulting ABox A' is called **expansion** of A with respect to T .

Theorem. Let T be a terminology and A and an ABox. Let A' be the result of expanding A based on the terminology T' obtained by unfolding T. Then M is a model of T ∪ A if and only if M is a model of A'.

$$M \models T \cup A \Longleftrightarrow M \models A'$$

Expansion of an unfolded TBox (example 3 (cont))

T=

- Bachelor $\equiv$ Student $\sqcap$ Undergrad
- Master $\equiv$ Student $\sqcap \neg$ Undergrad
- PhD $\equiv$ Student $\sqcap \neg$ Undergrad $\sqcap$ Research
- Assistant $\equiv$ Student $\sqcap \neg$ Undergrad $\sqcap$ Research $\sqcap$ Teach

A = {PhD(Rui)}

C = PhD

Exp(PhD(Rui)) = {PhD(Rui), Student(Rui), ¬Undergrad(Rui), Research(Rui), Master(Rui)}

Expansion - example

T = { Mother ≡ Female ⊓ ∃HasChild.Person, Female ≡ Person ⊓ ¬Male}

A = {Mother (Anna)}

The expansion A' of A with respect to T is:

A'={Mother(Anna), Female(Anna), HasChild(Anna, Anonymous#1), Person (Anna), Person (Anonymous#1), ¬Male (Anna)}

Expansion - example

T = { FemaleMother ≡ Female ⊓ ∀HasChild.female, Female ≡ Person ⊓ ¬Male}

A = {Mother (Anna), HasChild(Anna,Mary)}

The expansion A' of A with respect to T is:

A'={FemaleMother(Anna), Female(anna), Female(Mary), Person (Anna), ¬Male (Anna), Person(Mary), ¬Male (Mary)}

LODE - The logic of Knowledge bases

- Introduction
- LoDE definition
- Language
- Domain
- Interpretation function
- Entailment
- Expansion
- LoDe Reasoning
- LOD to LOE reasoning
- Entailment properties

LoDE reasoning

Theorem. Let $a' \in \exp(a)$. Then

$$A' \models a' \iff a' \in A' \iff T \cup A \models a$$

Observation 1. Theorem above corollary of main result of expansion.

Observation 2. To summarize:

- T is a teleontology
- A is an EG
- By unfolding T into T' and then using T' we expand A into A'
- A' is an extended EG where the original nodes and links of the A have been extended with the cojnuncts resulting from the expansion of A with respect to T'

Concept expansion (observations)

Observation 1: A LoDE EG may contain both primitive and defined etypes.

Observation 2: The expansion of an assertion involving a primitive etype does not change anything. The expansion of defined terms increases the number of nodes and also of links of the EG.

Observation 3: The teleontology provides the space of terms (definitions) and facts about concepts (descriptions). The EG defines the level of abstraction at which the EG is expressed.

Observation 4: For each assertion in the EG, the level of detail at which an EG is expressed can be increased by expansion or decreased by its inverse operation, thus allowing for the fine tuning of the EG to the user questions.

LODE (LOE) Reasoning problems

Instance retrieval Given an etype (or object/ data property), retrieve all the entities (or pairs entity, entity/data) which satisfy the etype (object/data property)

$$M \models E$$
$$M \models P$$

with M= I(T): A satisfiability problem!

LODE (LOE) Reasoning problems

Instance checking, Checking whether an assertion is entailed by a Model, i.e. checking whether

$$M \models E(e)$$
$$M \models P(e1, e2)$$

with M= I(T): A model checking problem!

LODE - The logic of Knowledge bases

- Introduction
- LoDE definition
- Language
- Domain
- Interpretation function
- Entailment
- Expansion
- LoDe Reasoning
- LOD to LOE reasoning
- Entailment properties

LODE - Reasoning problems

- Consistency of A with respect to T: *are the facts in A coherent with the descriptions of T?*
- Instance checking: *is an assertion implied by the KB?*
- Instance retrieval: *return the (possibly empty) set of instances that satisfy an input concept*
- Concept realization: *return the most specific concept satisfying an input entity*

Reasoning problems (compare with LOE)

Consistency An ABox A is consistent with respect to T if there is an interpretation I which is a model of both A and T .

Instance checking Checking whether an assertion *C*(*a*) is entailed by an ABox, i.e. checking whether *a* belongs to *C*.

A |= *C*(*a*) if every interpretation that satisfies A also satisfies *C*(*a*).

Instance retrieval Given a concept *C*, retrieve all the instances a which satisfy *C*

Concept realization Given a set of concepts and an individual *a* find the most specific concept(s) *C* (w.r.t. subsumption) such that A |= *C*(*a*)

Consistency checking

Definition 6 (Consistency checking) An ABox A is consistent with respect to a TBox T , if there is an interpretation that is a model of both A and T . We say that A is consistent if it is consistent w.r.t the empty T

Consistency checking (example)

T

- $Parent \equiv Mother \sqcup Father$
- $Father \equiv Male \sqcap hasChild$
- $Mother \equiv Female \sqcap hasChild$
- $Male \equiv Person \sqcap \neg Female$

A

- *Mother(Mary)*
- *Father(Mary)*

We can see from the expansion of T that a father cannot be also a mother, and so A is inconsistent with respect to T

Instance checking

Definition 7 (Instance checking) A procedure to verify if the individual *a* belongs to a certain concept *C* or a certain role *r*. In formulas

$$A \models C(a)$$

if every interpretation that satisfies A also satisfies $C(a)$.

We can relate this reasoning task to the consistency checking:

$$A \models C(a) \equiv A \cup \{\neg C(a)\} \text{ is inconsistent}$$

Thus, if we demonstrate that $A \cup \{\neg C(a)\}$ is inconsistent, the individual *a* does not belong to the concept *C*.

Instance checking (example)

T

- $Undergraduate \sqsubseteq \neg Teach$
- $Bachelor \equiv Student \sqcap Undergraduate$
- $Master \equiv Student \sqcap \neg Undergraduate$
- $PhD \equiv Master \sqcap Research$
- $Assistant \equiv PhD \sqcap Teach$

A

- $Master(Chen)$
- $PhD(Enzo)$
- $Assistant(Rui)$

Instance checking (example)

Example 6

Is PhD(Rui) satisfied? **(Yes)**

The problem can be formalized as:

$$A \models PhD(Rui)$$

To solve the task, we have to expand the TBox into the Abox.

Proof:

- Assistant(Rui)
- PhD(Rui)
- Teach(Rui)

Formally the procedure proves that A ∪ {¬*PhD*(*Rui*) is unsatisfiable}

Instance retrieval /concept realization (example)

T

- $Undergraduate \sqsubseteq \neg Teach$
- $Bachelor \equiv Student \sqcap Undergraduate$
- $Master \equiv Student \sqcap \neg Undergraduate$
- $PhD \equiv Master \sqcap Research$
- $Assistant \equiv PhD \sqcap Teach$

A

- $Master(Chen)$
- $PhD(Enzo)$
- $Assistant(Rui)$

Instance retrieval (example)

Find all instances of ¬ Undergraduate **{Chen, Enzo, Rui}**

Proof:

Master(Chen)
Student(Chen)
¬ Undergrad(Chen)

PhD(Enzo)
Master(Enzo)
Research(Enzo)
Student(Enzo)
¬ Undergrad(Enzo)

Assistant(Rui)
PhD(Rui)
Teach(Rui)
Master(Rui)
Research(Rui)
Student(Rui)
¬Undergrad(Rui)

Concept realization (example)

Given the instance *Rui* and the concepts {*Student, PhD, Assistant*} find the most specific concept that satisfies *Rui*, i.e. such that A |= *C*(*Rui*)

Proof:

The subsumption chain for the three proposed concepts is:

$$\text{Assistant} \sqsubseteq \text{PhD} \sqsubseteq \text{Student}$$

Starting from the left, we check if *Rui* belongs to such concept, and we immediately find our answer in the concept Assistant

Reasoning reduction

Proposition (Reduction of reasoning)

■ Instance retrieval and concept realization can be implemented via instance checking

■ Instance checking can be reduced to consistency

Remark Reasoning tasks for Abox can be reduced to consistency checking.

Remark All Tbox reasoning tasks can be reduced to concept consistency. Thus, every reasoning task seen so far to ABox consistency checking.

Remark A TBox T can be said to be consistent if there is a model of both T and the empty ABox.

Reasoning problems (Reduction)

- **TO BE DONE**

LODE - The logic of Knowledge bases

- Introduction
- LoDE definition
- Language
- Domain
- Interpretation function
- Entailment
- Expansion
- Reasoning problems
- Entailment properties

Observations (Logical entailment LODE – properties)

Intuition (Reflexivity): $w \ |= w$

YES!

Intuition (Cut): If $\Gamma \ |= w1$ and $\Sigma \cup \{w1\} \ |= w2$ then $\Gamma \cup \Sigma \ |= w2$

TRIVIAL: only with W2 in Σ

Intuition (Compactness) If $\Gamma \ |= w$ then there is a finite subset $\Gamma 0 \subseteq \Gamma$ such that $\Gamma 0 \ |= w$

TRIVIAL: we only have formulas of finite length

Observations (Logical entailment LODE – properties)

Intuition (Monotonicity): If $\Gamma \models w$ then $\Gamma \cup \Sigma \models w$

YES!

Intuition (NonMonotonicity) $\Gamma \models w$ and $\Gamma \cup \Sigma$ not$\models w$

NO!

NOTE: same as LOE

LODE - The logic of Knowledge Bases (KBs)

Sept 13,2023