Mathematical Logics
Description Logic: Tbox and Abox

Fausto Giunchiglia and Mattia Fumagallli

University of Trento

**Originally by Luciano Serafini and Chiara Ghidini*
Modified by Fausto Giunchiglia and Mattia Fumagalli

Lecture index

1. Families of Description Logics
2. TBOX: syntax and semantics
3. TBOX: terminology
4. TBOX: reasoning
5. ABOX: syntax and semantics
6. ABOX: reasoning
7. Closed World Assumption (CWA) and Open World Assumption (OWA)

Abox – Reasoning services

Given an ABox A, we reason (w.r.t. a TBox T) about the following:

- Satisfiability/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*).

NOTE: second and third services correspond to Query Answering

Reasoning via expansion of the ABox

The Reasoning services over an ABox w.r.t. an acyclic (*)TBox can be reduced to checking an expanded ABox.

We define the expansion of an ABox A with respect to T as the ABox A' that is obtained from A by replacing each concept assertion C(*a*) with the assertion C'(*a*), with C' the expansion of C with respect to T.

The expansion C'(a) of C(a) is the set of concepts C''(a) such that C''(a) are used to define C(a). The expansion A' of A with respect to T contains only primitive (not defined concepts).

- ❑**A is consistent with respect to T iff its expansion A' is consistent**

- ❑A is consistent iff A is satisfiable (**), i.e. non contradictory.

NOTE 2:

(*) acyclic when no cycles in definitions

(**) in PL, under the usual translation, with C(a) considered as a proposition different from C(b)

Abox - Reasoning via Expansion

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)

The expansion of A w.r.t. T (concepts in black, NOT in blue):

Master(Chen)	PhD(Enzo)	Assistant(Rui)
Student(Chen)	Master(Enzo)	PhD(Rui)
$\neg$Undergraduate(Chen)	Research(Enzo)	Teach(Rui)
	Student(Enzo)	Master(Rui)
	$\neg$Undergraduate(Enzo)	Research(Rui)
		Student(Rui)
		$\neg$Undergraduate(Rui)

Abox - Consistency

Satisfiability/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.

We say that A is consistent if it is consistent with respect to the empty TBox

T	A
Parent $\equiv$ Mother $\sqcup$ Father	Mother(Mary)
Father $\equiv$ Male $\sqcap$ hasChild	Father(Mary)
Mother $\equiv$ Female $\sqcap$ hasChild	
Male $\equiv$ Person $\sqcap$ $\neg$ Female	

A is not consistent w.r.t. T: In fact, from the expansion of T we get that Mother and Father are disjoint.

A is consistent (w.r.t. the empty TBox, no constraints)

Abox - Instance checking

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).

A ⊨ C(a) iff A ∪ {¬ C(a)} is inconsistent

Consider T and A from the previous example.

Is Phd(Rui) entailed?
YES! The assertion is in the expansion of A.

Abox - Instance retrieval

Instance retrieval: given a concept C, retrieve all the instances *a* which satisfy C.

Implementation: A trivial, but ineffficent implementation consists in doing instance checking for all instances.

Consider T and A from the previous example.

Find all the instances of ¬Undergraduate
Looking at the expansion of A we have {Chen, Enzo, Rui}

Abox - Concept realization

Concept realization: given *a* set of concepts and an individual a find the most specific concept(s) C (w.r.t. subsumption ordering) such that A ⊨ C(a).

Dual problem of Instance retrieval

Implementation: A trivial, but inefficent implementation consists in doing instance checking for all concepts.

Abox - Concept realization

Consider T and A from the previous 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)

Given the instance Rui, and the concept set {Student, PhD, Assistant} find <u>the most specific concept</u> C such that A$\models$C(Rui)

Rui is in the extension of all the concepts above.

The following chain of subsumptions holds: Assistant $\sqsubseteq$ PhD $\sqsubseteq$ Student

Therefore, the most specific concept is Assistant.

Mathematical Logics
Description Logic: Tbox and Abox

Fausto Giunchiglia and Mattia Fumagallli

University of Trento

**Originally by Luciano Serafini and Chiara Ghidini*
Modified by Fausto Giunchiglia and Mattia Fumagalli