Ontology Languages

Reasoning Problems

Ira A. Fulton Schools of Engineering
Arizona State University

Objectives

Objective

Identify the kinds of reasoning in ontologies

Objective

Apply ontology tools for reasoning problems

Concept Satisfiability

- Checking whether a concept C is satisfiable with respect to a knowledge base K, i.e., whether there exists a model I of K such that $C^I \neq \emptyset$.

- Formally: $K \nvDash C \equiv \bot$

Example: Concept Satisfiability

ParentOfOnlyMaleChildren ≡ Person⊓∃hasChild.Male⊓∀hasChild.Male

Male ⊑ ¬Female

- Is each of the concepts satisfiable w.r.t. K?
  - ParentOfOnlyMaleChildren ⊓ ∃hasChild.Male
  - ParentOfOnlyMaleChildren ⊓ ∃hasChild.Female

Let's Test on Protégé

description logics	OWL in Protégé
concept	class
role	object property
constant/individual	individual
theory	ontology
$\top$	owl:Thing
$\bot$	owl:Nothing
$\exists r.C$	r some C
$\forall r.C$	r only C
$\neg C$	not C
$C_1 \sqcup C_2$	C_1 or C_2
$C_1 \sqcap C_2$	C_1 and C_2

Subsumption

- This is the problem of checking whether C is subsumed by D with respect to a knowledge base K, i.e., whether $C^I \subseteq D^I$ in every model I of K.

- Formally: K |= C ⊑ D

Example: Subsumption

$\text{Parent} \equiv \text{Person} \sqcap \exists \text{hasChild}.\text{Person} \sqcap \forall \text{hasChild}.\text{Person}$

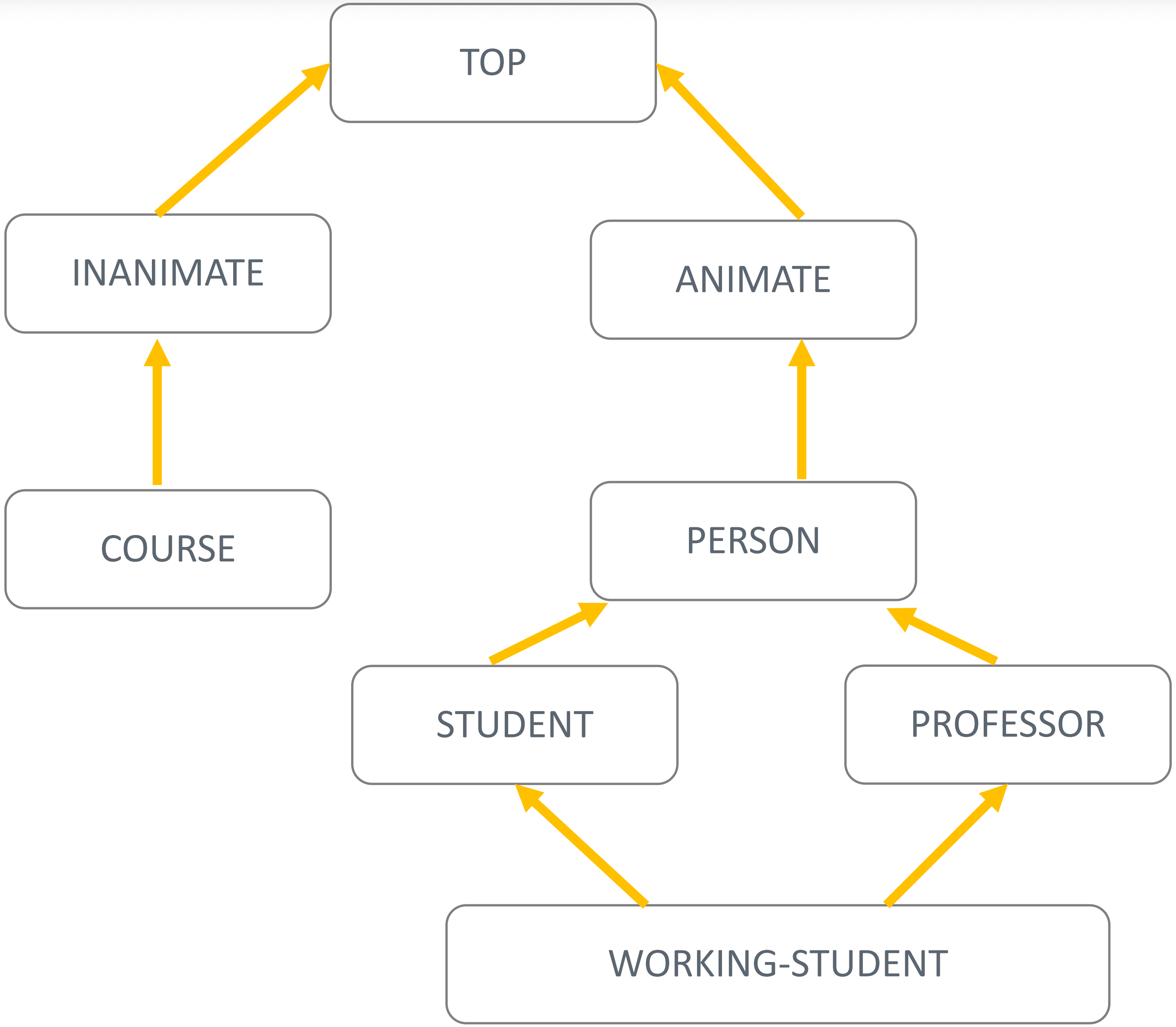
$\text{ParentOfOnlyMaleChildren} \equiv \text{Person} \sqcap \exists \text{hasChild}.\text{Male} \sqcap \forall \text{hasChild}.\text{Male}$

$\text{Male} \sqsubseteq \text{Person}$

Q: $\text{ParentOfOnlyMaleChildren} \sqsubseteq \text{Parent}$?

Q: $\text{ParentOfOnlyMaleChildren} \sqsubseteq \text{Male}$?

Taxonomies (1 of 3)

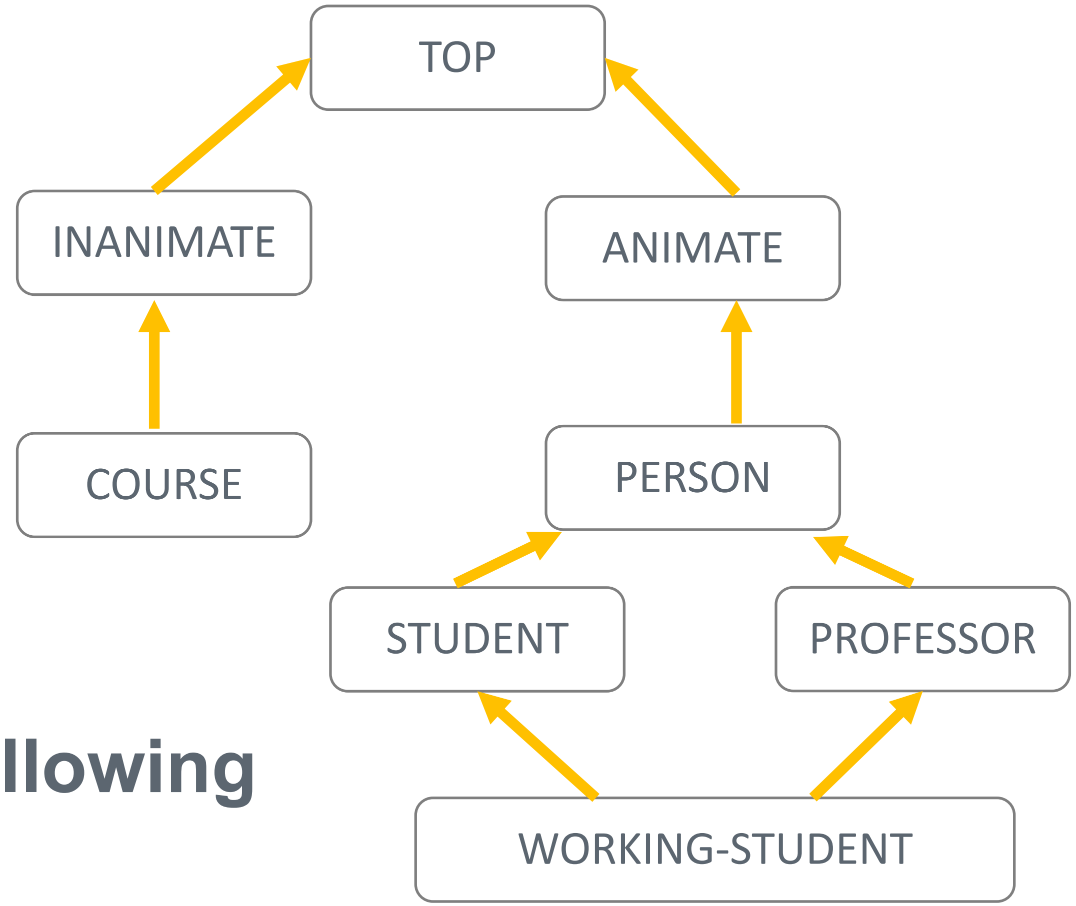


Taxonomies (2 of 3)

- The subsumption relationship between concepts defined by $\sqsubseteq$ is a partial order (i.e., it is reflexive, antisymmetric and transitive).

- Subsumption induces a taxonomy such as the one on the previous slide where only direct subsumptions have been explicitly drawn.

Taxonomies (3 of 3)



Question: What is the place of the following concept in the example taxonomy?

$$N \equiv ANIMATE \sqcap (STUDENT \sqcup PROFESSOR)$$

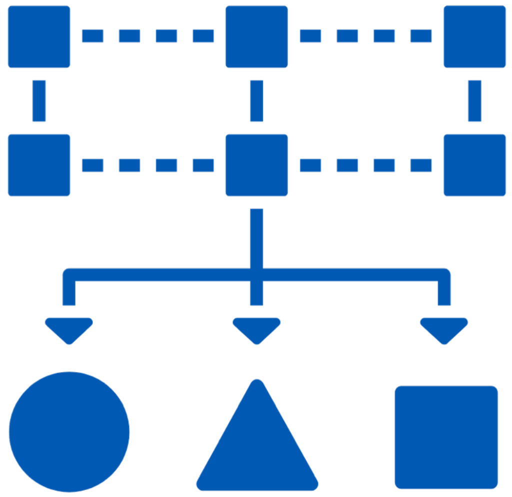
Classification

- **The problem of classification:** Given a concept C and a TBox T, for all concepts D, determine whether D subsumes C, or D is subsumed by C

- Intuitively, this amounts to finding the “right place” for C in the taxonomy implicitly present in T

- **Classification** is the task of inserting new concepts in a taxonomy. It is sorting in partial orders

- What is the solution to the classification problem posed in the previous slide?



Knowledge Base Satisfiability

- Checking whether K is satisfiable, i.e., whether it has a model.

- Is the following KB satisfiable?

$$\text{ParentOfOnlyMaleChildren} \equiv \text{Person} \sqcap \exists \text{hasChild.Male} \sqcap \forall \text{hasChild.Male}$$

$$\text{Male} \sqsubseteq \text{Person},\ \text{Female} \sqsubseteq \text{Person},\ \text{Male} \sqsubseteq \neg\text{Female}$$

Male(JOHN), Male(NICK), Female(ANNA),

hasChild(JOHN, NICK), hasChild(JOHN, ANNA),

ParentOfOnlyMaleChildren(JOHN)

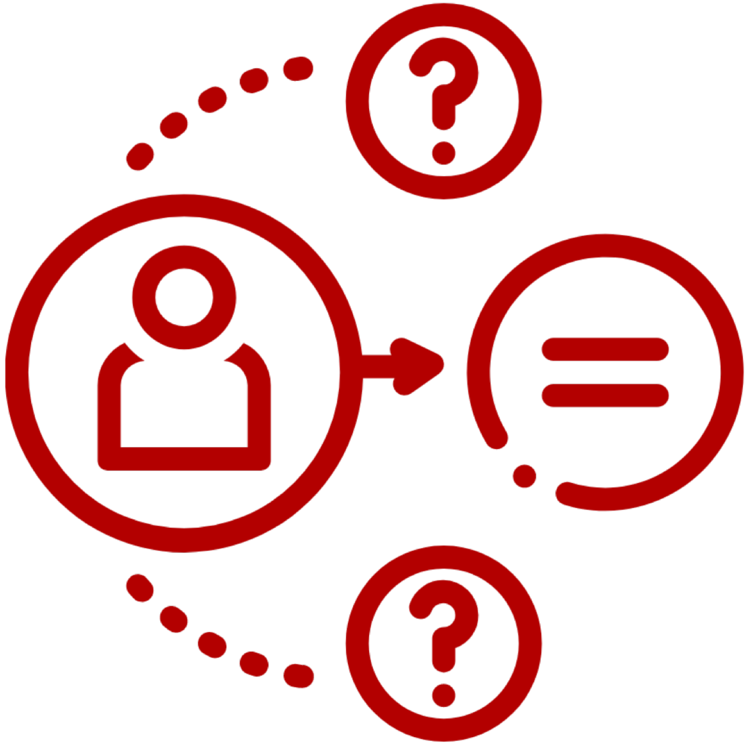
Instance Checking

- The problem of checking whether the assertion C(a) is satisfied in every model of K

- Formally: $K \models C(a)$

Answering Concept Queries

- Find all a such that {a | K |= C(a)}.



Reduction to Satisfiability

- Some of the previous reasoning problems can be solved by reducing them to the problem of knowledge base satisfiability:

$\Gamma \models F$ iff
$\Gamma \cup \{\neg F\}$ is unsat.

- Concept Satisfiability: K $\not\models$ C ≡ ⊥ iff there exists an x such that K ∪ {C(x)} is satisfiable

- Subsumption: K |= C ⊑ D iff there exists an x such that K∪{(C⊓¬D)(x)} is not satisfiable

- Instance Checking: K |= C(a) iff K ∪ {¬C(a)} is not satisfiable

Reasoning Algorithms

- Terminating, complete and efficient algorithms for deciding satisfiability, and all the other reasoning problems mentioned earlier, are available for ALC
- These algorithms are based on tableaux-calculi techniques
- Completeness is important for the usability of description logics in real applications
- Such algorithms have been shown to be efficient for real knowledge bases, even if the problem in the corresponding logic is in PSPACE or EXPTIME

Wrap-Up