Part 7
Reasoning tasks and their reducibility

Jostein Solaas Håkon Dissen

November 4, 2015

Intro

- Queries and reasoning on DLs
- Will look at a few problems
- Some can be reduced to each other

Knowledge Base Satisfiability

- KB is satisfiable iff there exists an I such that $I \models KB$
- Otherwise contradictory (unsatisfiable)
- Principle of explosion

Axiom Entailment

- We want to check if a statement α entailed by the KB
- Proof by contradiction
- We have α, β and KB
- β is the opposite of α
- Lets use Satisfiability
- Iff the $KB \cup \{\beta\}$ is unsatisfiable KB entails α

Concept Satisfiability

- If C may contain individuals, it is satisfiable
- Could signal modeling errors
- There exists a model for C that makes $C^I \neq \emptyset$
- Can be reduced to axiom entailment
- $KB \models C \sqsubseteq \bot$

Instance Retrieval

- The task of retrieving all instances of concept C
- Two problems:
  - Many models that can differ on the class of an individual
  - Models may vary, and may not even contain the same individuals
- Two solutions:
  - Retrieve only if individual belongs to C for each model of KB
  - Only retrieve named individuals
- The problem can be formulated as $KB \models C(a)$

Classification

- Seeks to create a hierarchy of subsumption relationships of concepts
- Defines $\sqsubseteq_{KB}$ by $A \sqsubseteq_{KB} B$ iff $KB \models A \sqsubseteq B$
- $\sqsubseteq_{KB}$ is a preorder, which makes it faster to calculate
- Helps in the KB modeling phase
- Preprocessing for subsequent KB work

Conjunctive Query Answering

- Sequence of logical ands
- Query either returns true/false or tuples with individuals
- $\exists y \exists z (\text{childOf}(x, y) \wedge \text{childOf}(x, z) \wedge \text{married}(y, z))$
- $\exists x \exists y \exists z (\text{childOf}(x, y) \wedge \text{childOf}(x, z) \wedge \text{married}(y, z))$
- Not polynomial

Other Reasoning Tasks

- Induction
  - Generalize facts
- Abduction
  - Given KB and α, *guess* $KB \cup KB' \models \alpha$
- Explanation
  - Given $KB \models \alpha$ find $KB' \subset KB$ such that $KB' \models \alpha$ while $KB'' \subset KB'$ and $KB'' \not\models \alpha$
- Module Extraction
  - Find smaller KBs in a large KB