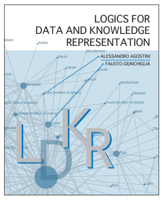
Logics for Data and Knowledge Representation

Alessandro Agostini
agostini@dit.unitn.it

Fausto Giunchiglia
fausto@dit.unitn.it

University of Trento

The Logic of Descriptions



- Introduction
- Language (Syntax)
- Semantics
  - interpretation
  - entailment
- Knowledge Bases
- Reasoning Services

Reasoning Services (Inferences with Concepts)

Typical Reasoning Services

- **Classification of concepts**: determine sub-concept / super-concept relationships (subsumption relationships) between the concepts of a given terminology.
- **Classification of objects**: determine whether a given individual is always an instance of a certain concept.
- DLs provide other reasoning services.

Reasoning Services

- The basic reasoning services for a DL KR-system aim to answer decision problems on:

  1. **Satisfiability**
  2. Subsumption
  3. Equivalence
  4. Disjointness
- Using concept complementation 2,3,4 can be reduced to 1. [We'll see it soon.]

Satisfiability

- There are three variants - variant 1 is:

  Knowledge base SAT: the problem of deciding whether a DL knowledge base KB is satisfiable (i.e. has a model).
- In short: Is there (Δ, I) s.t. $(\Delta, I) \models KB$?
- This variant is the most important, because all other variants can be reduced to this.

Concept Satisfiability

- Definition.
  (1) A DL (AL*) concept C is satisfiable if there is an interpretation (Δ,I) s.t. I(C) $\neq \emptyset$.
  - (Δ,I) satisfies C; (Δ,I) called a model of C.

  (2) C is unsatisfiable if it is not satisfiable.
- Example. For (Δ,I) interpreting the 'LDKR class' world, ∃teachOf.Ldkr is satisfiable.

Satisfiability

- There are three variants - variant 2 is:

  Concept SAT: the problem of deciding whether a concept description C does not always denote the empty set (C satisfiable).
- In short: Is there (Δ,I) s.t. I(C) $\neq \emptyset$?
- Also called **concept coherence**.
  If I(C) $\neq \emptyset$, then (Δ,I) is called a model of C.

Satisfiability

- There are three variants - variant 3 is:

  (Concept) Consistency: the problem of deciding whether a concept C is consistent in (satisfiable in) a DL knowledge base KB.
- In short: Is there an interpretation (Δ,I) s.t.
  1. (Δ,I) |= KB and
  2. I(C) $\neq \emptyset$ (i.e. there is a∈Δ s.t. I |= C(*a*))?

Instance Checking

- A typical DL reasoning service based on concept consistency is instance checking:

  Instance Checking: the problem of deciding whether an assertion C(*a*) is entailed by a DL knowledge base KB.
- In short: Is KB |= C(*a*)?

Query-answering by Instance Checking

- We focus on query-answering as a particular case of instance checking of the form
  DB |= C(*d*)
  - DB is a documentary knowledge base;
  - C(*d*) is an assertion about a named individual *d representing a document* d∈DB.
    - Note the different use of symbols *d*, d.

Subsumption

- There are two variants - variant 1 is:

  Concept SUB: the problem of deciding whether a concept D (the *subsumer*) is considered more general than a concept C (the *subsumee*). For, is C subsumed by D?
- In short: Is |= C ⊑ D? (**NB**: not "Is C ⊑ D?"!)
- Example: Is every animal (C) a mammal (D)?

Subsumption

- There are two variants - variant 2 is:

  Concept SUB in a KB: the problem of deciding whether a DL knowledge base KB entails a DL concept inclusion axiom $C \sqsubseteq D$.
- In short: Is KB $\models C \sqsubseteq D$?
- Example: Is every house (C) a home (D) according to knowledge represented in KB?

Equivalence

- There are two variants - variant 1 is:

  Equivalence: the problem of deciding whether a concept C is equivalent to a concept D (i.e. does D always denote the set that C denotes and vice versa?)
- In short: Is $\models C \equiv D$? (**NB**: not "Is $C \equiv D$?"!)
- Example: Is every house exactly a home?

Example

- Define:
  $C = \forall hasChild.Female \sqcap \forall hasChild.Student$

  $D = \forall hasChild.Female \sqcap Student$.
- Then $\models C \equiv D$. In fact, for all I,

  $I(C)=$
  $= I(\forall hasChild.Female) \cap I(\forall hasChild.Student)$

  $= I(\forall hasChild.Female \sqcap Student) = I(D)$.

Equivalence

- There are two variants - variant 2 is:

  Equivalence in a KB: the problem of deciding whether a DL knowledge base KB entails a DL equivalence axiom $C \equiv D$.
- In short: Is KB $\models C \equiv D$?

Disjointness

- There are three variants - variant 1 is:

  Concept DISJ: the problem of deciding whether two concepts C, D are disjoint.
- In short: For all (Δ, I), is $I(C) \cap I(D) = \emptyset$?
- Definition. DL concepts C,D are disjoint if for all interpretations (Δ, I), $I(C) \cap I(D) = \emptyset$.
- Example: ICT and DIT (i.e. $\models ICT \sqcap DIT \sqsubseteq \bot$).

Disjointness

- There are three variants - variant 2 is:

  Concept DISJ in a KB: the problem of deciding whether two concepts C, D are disjoint in a DL knowledge base KB.
- In short: For all interpretations (Δ, I), is $I(C) \cap I(D) = \emptyset$ if $(\Delta, I) \models$ KB?
- In symbols (using $\sqsubseteq$): Is KB $\models C \sqcap D \sqsubseteq \bot$?

Example

- Let KB = { Body⊑∃isPartOf.Human,
  isDirectPartOf ⊑ isPartOf,
  Men ⊑ Human, Men(John),
  isPartOf(head,John),Head(head) }.
- Then KB |= ¬Head⊓∃isPartOf.Body ⊑⊥.
- Exercise.
  Is KB |= Head⊓∀isDirectPartOf.Human ⊑⊥?

Properties

Reduction to Satisfiability

- Among the decision problems defined on

1. **Satisfiability (SAT)**
2. Subsumption
3. Equivalence
4. Disjointness

SAT is the most important. To decide SAT (efficiently) is crucial to answer all others.

Reduction to Satisfiability

- Theorem. Let C, D be concepts of an AL* language with ¬ (general negation) (e.g. *ALC*). Then:

1. |= C⊑D iff C⊓¬D is unsatisfiable.
2. |= C≡D iff C⊓¬D, ¬C⊓D are unsatisfiable.
3. |= C⊓D ⊑⊥ iff C⊓D is unsatisfiable.

Proof: omitted.

Remarks

- So it is sufficient to develop algorithms that decide concept SAT if one is interested in decision procedures for any of the other three reasoning services.
- But when studying the complexity of the above inferences in a particular DL (AL*) language, it is not sufficient to restrict oneself to satisfiability. Why?

Remarks

- Subsumption and equivalence problems for a language without full complement (¬) give rise to satisfiability decision problems for concepts not contained in the language.
- From the viewpoint of worst-case complexity, subsumption and equivalence are the most specific and the most general kind of reasoning services, respectively.

Reduction to Subsumption

- All reasoning services can be rephrased via concept subsumption, while unsatisfiability is a special case of each reasoning service:
- Theorem. Let C, D be AL*-concepts.

1. C is unsatisfiable iff $\models C \sqsubseteq \bot$.
2. $\models C \equiv D$ (equiv.) iff $\models C \sqsubseteq D$ and $\models C \sqsubseteq D$.
3. C and D are disjoint iff $\models C \sqcap D \sqsubseteq \bot$.

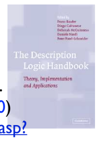
Decidability of SAT

- Theorem. Concept satisfiability of AL* concepts is decidable.
- Since for all AL* concepts holds that $\models C \sqsubseteq D$ iff $C \sqcap D$ is unsatisfiable, we have:
- Corollary. Concept subsumption of AL* concepts is decidable.

Summary on DLs

- DLs are a family of logic-based knowledge representation (KR) formalisms which
  - describe domain in terms of
    - concepts
    - roles
    - individuals ("grounding")
- DLs have many applications; but best known as basis of ontology languages, e.g. OWL.

Some Resources



- Books:
  - F. Baader et. al., *Handbook of Description Logics*. Cambridge University Press, 2002. (Chs 1,2,4,10) http://www.cambridge.org/catalogue/catalogue.asp?isbn=0521781760
- Papers & Links (in any):
  - http://dit.unitn.it/~ldkr#Biblio/