Principles of Knowledge Representation and Reasoning
Semantic Networks and Description Logics II:
Description Logics – Terminology and Notation

Bernhard Nebel, Stefan Wölfl, and Felix Lindner
November 23, 2015
1 Introduction

- Motivation
- History
- Systems and Applications
- Description Logics in a Nutshell
Motivation

- Main problem with **semantic networks and frames**
  … the lack of **formal semantics**!
- Disadvantage of simple **inheritance networks**
  … concepts are atomic and do not have any **structure**

⇝ Brachman’s **structural inheritance networks** (1977)
Structural inheritance networks

- Concepts are defined/described using a small set of well-defined operators
- Distinction between conceptual and object-related knowledge
- Computation of subconcept relation and of instance relation
- Strict inheritance (of the entire structure of a concept): inherited properties cannot be overridden
Systems and applications

- **Systems:**
  - **KL-ONE:** First implementation of the ideas (1978)
  - then: NIKL, KL-TWO, KRYPTON, KANDOR, CLASSIC, BACK, KRIS, YAK, CRACK …
  - later: FaCT, DLP, RACER 1998
  - currently: FaCT++, RACER, Pellet, HermiT, and many more …

- **Applications:**
  - First, natural language understanding systems,
  - then configuration systems,
  - and information systems,
  - currently, it is one tool for the **Semantic Web**

- **Languages:** DAML+OIL, now **OWL** (Web Ontology Language)
Description logics

- Previously also known as KL-ONE-alike languages, frame-based languages, terminological logics, concept languages

- Description Logics (DL) allow us
  - to describe concepts using complex descriptions,
  - to introduce the terminology of an application and to structure it (TBox),
  - to introduce objects and relate them to the introduced terminology (ABox),
  - and to reason about the terminology and the objects.
Informal example

**Male** is: the opposite of **female**

A **human** is a kind of: **living entity**

A **woman** is: a **human** and a **female**

A **man** is: a **human** and a **male**

A **mother** is: a **woman** with at least one child that is a **human**

A **father** is: a **man** with at least one child that is a **human**

A **parent** is: a **mother** or a **father**

A **grandmother** is: a **woman**, with at least one child that is a **parent**

A **mother-wod** is: a **mother** with only male children

Elizabeth is a **woman**

Elizabeth has the child **Charles**

Charles is a **man**

Diana is a **mother-wod**

Diana has the child **William**

*Possible Questions:*

Is a grandmother a parent?

Is Diana a parent?

Is William a man?

Is Elizabeth a mother-wod?
2 Concepts and Roles

- Concept Forming Operators
- Role Forming Operators
Atomic concepts and roles

- **Concept names:**
  - E.g., Grandmother, Male, ...(in the following usually capitalized)
  - We will use **symbols** such as \( A, A_1, \ldots \) for concept names
  - **Semantics:** Monadic predicates \( A(\cdot) \) or set-theoretically a subset of the universe \( A^I \subseteq D \).

- **Role names:**
  - In our example, e.g., child. Often we will use names such as has-child or something similar (in the following usually lowercase).
  - Role names are **disjoint** from concept names
  - **Symbolically:** \( t, t_1, \ldots \)
  - **Semantics:** Binary relations \( t(\cdot, \cdot) \) or set-theoretically \( t^I \subseteq D \times D \).
Concept and role description

- From (atomic) concept and role names, complex concept and role descriptions can be created.
- In our example, e.g., “Human and Female.”
- Symbolically: $C$ for concept descriptions and $r$ for role descriptions.

Which particular constructs are available depends on the chosen description logic!

- **FOL semantics:** A concept description $C$ corresponds to a formula $C(x)$ with the free variable $x$. Similarly with role descriptions $r$: they correspond to formulae $r(x, y)$ with free variables $x, y$.
- **Set semantics:**

\[
C^\mathcal{I} = \{d \in D : C(d) \text{ “is true in” } \mathcal{I}\} \\
C^\mathcal{I} = \{(d, e) \in D^2 : r(d, e) \text{ “is true in” } \mathcal{I}\}
\]
Boolean operators

- **Syntax:** let $C$ and $D$ be concept descriptions, then the following are also concept descriptions:
  - $C \sqcap D$ (concept conjunction)
  - $C \sqcup D$ (concept disjunction)
  - $\neg C$ (concept negation)

- **Examples:**
  - Human $\sqcap$ Female
  - Father $\sqcup$ Mother
  - $\neg$ Female

- **FOL semantics:** $C(x) \land D(x)$, $C(x) \lor D(x)$, $\neg C(x)$

- **Set semantics:** $C^I \cap D^I$, $C^I \cup D^I$, $D \setminus C^I$
Role restrictions

- **Motivation:**
  - Often we want to describe something by restricting the possible “fillers” of a role, e.g. Mother—woman.
  - Sometimes we want to say that there is at least a filler of a particular type, e.g. Grandmother

- **Idea:** Use quantifiers that range over the role-fillers
  - Mother $\sqcap \forall \text{has-child. Man}$
  - Woman $\sqcap \exists \text{has-child. Parent}$

- **FOL semantics:**
  \[
  (\exists r. C)(x) = \exists y (r(x, y) \land C(y))
  \]
  \[
  (\forall r. C)(x) = \forall y (r(x, y) \rightarrow C(y))
  \]

- **Set semantics:**
  \[
  (\exists r. C)^I = \{ d \in D : \text{there ex. some } e \text{ s.t. } (d, e) \in r^I \land e \in C^I \} \]
  \[
  (\forall r. C)^I = \{ d \in D : \text{for each } e \text{ with } (d, e) \in r^I, e \in C^I \} \]
Cardinality restriction

**Motivation:**
- Often we want to describe something by restricting the number of possible “fillers” of a role, e.g., a Mother with at least 3 children or at most 2 children.

**Idea:** We restrict the cardinality of the role filler sets:
- Mother ⊓ ≥ 3 has-child
- Mother ⊓ ≤ 2 has-child

**FOL semantics:**
\[
(\geq n \ r)(x) = \exists y_1 \ldots y_n \left( r(x, y_1) \land \cdots \land r(x, y_n) \land y_1 \neq y_2 \land \cdots \land y_{n-1} \neq y_n \right)
\]
\[
(\leq n \ r)(x) = \neg (\geq n + 1 \ r)(x)
\]

**Set semantics:**
\[
(\geq n \ r)^\mathcal{I} = \{ d \in D : \left\{ e \in D : r^\mathcal{I}(d, e) \right\} \geq n \}
\]
\[
(\leq n \ r)^\mathcal{I} = D \setminus (\geq n + 1 \ r)^\mathcal{I}
\]
Inverse roles

**Motivation:**
- How can we describe the concept “children of rich parents”?

**Idea:** Define the “inverse” role for a given role (the converse relation)
- $\text{has-child}^{-1}$

**Example:** $\exists \text{has-child}^{-1}. \text{Rich}$

**FOL semantics:**

$$r^{-1}(x, y) = r(y, x)$$

**Set semantics:**

$$(r^{-1})^I = \{(d, e) \in D^2 : (e, d) \in r^I\}$$
Role composition

- **Motivation:**
  - How can we define the role `has-grandchild` given the role `has-child`?

- **Idea:** Compose roles (as one can compose binary relations)
  - `has-child ∘ has-child`

- **FOL semantics:**
  
  \[
  (r \circ s)(x, y) = \exists z (r(x, z) \land s(z, y))
  \]

- **Set semantics:**
  
  \[
  (r \circ s)^I = \{(d, e) \in D^2 : \exists f \text{ s.t. } (d, f) \in r^I \land (f, e) \in s^I\}
  \]
Role value maps

- **Motivation:**
  - How do we express the concept “women who know all the friends of their children”

- **Idea:** Relate role filler sets to each other
  - Woman ⊓ (has-child ◦ has-friend ⊑ knows)

- **FOL semantics:**
  \[(r ⊑ s)(x) = \forall y (r(x, y) \rightarrow s(x, y))\]

- **Set semantics:** Let \(r^I(d) = \{e : r^I(d, e)\}\).
  \[(r ⊑ s)^I = \{d \in D : r^I(d) \subseteq s^I(d)\}\]

- **Note:** Role value maps lead to undecidability of satisfiability testing of concept descriptions!
3 TBox and ABox

- Terminology Box
- Assertional Box
- Example
In order to introduce new terms, we use two kinds of terminological axioms:

- $A \equiv C$
- $A \sqsubseteq C$

where $A$ is a concept name and $C$ is a concept description.

A terminology or TBox is a finite set of such axioms with the following additional restrictions:

- no multiple definitions of the same symbol such as $A \equiv C$, $A \sqsubseteq D$
- no cyclic definitions (even not indirectly), such as $A \equiv \forall r . B$, $B \equiv \exists s . A$
TBoxes: semantics

- TBoxes restrict the set of possible interpretations.
- **FOL semantics:**
  - \( A \equiv C \) corresponds to \( \forall x (A(x) \leftrightarrow C(x)) \)
  - \( A \sqsubseteq C \) corresponds to \( \forall x (A(x) \rightarrow C(x)) \)
- **Set semantics:**
  - \( A \equiv C \) corresponds to \( A^\mathcal{I} = C^\mathcal{I} \)
  - \( A \sqsubseteq C \) corresponds to \( A^\mathcal{I} \subseteq C^\mathcal{I} \)

- Non-empty interpretations which satisfy all terminological axioms are called **models** of the TBox.
Assertional box

In order to state something about objects in the world, we use two forms of assertions:

- $a : C$
- $(a, b) : r$

where $a$ and $b$ are individual names (e.g., ELIZABETH, PHILIP), $C$ is a concept description, and $r$ is a role description.

An ABox is a finite set of assertions.
ABoxes: semantics

- **Individual names** are interpreted as elements of the universe under the unique-name-assumption, i.e., different names refer to different objects.

- **Assertions** express that an object is an instance of a concept or that two objects are related by a role.

- **FOL semantics:**
  - $a : C$ corresponds to $C(a)$
  - $(a, b) : r$ corresponds to $r(a, b)$

- **Set semantics:**
  - $a^I \in D$
  - $a : C$ corresponds to $a^I \in C^I$
  - $(a, b) : r$ corresponds to $(a^I, b^I) \in r^I$

- **Models** of an ABox and of ABox + TBox can be defined analogously to models of a TBox.
Example TBox

Male = ¬Female
Human ⊑ Living_entity
Woman = Human ⊓ Female
Man = Human ⊓ Male
Mother = Woman ⊓ ∃has-child.Human
Father = Man ⊓ ∃has-child.Human
Parent = Father ⊔ Mother
Grandmother = Woman ⊓ ∃has-child.Parent
Mother-without-daughter = Mother ⊓ ∀has-child.Male
Mother-with-many-children = Mother ⊓ (∑ ≥ 3 has-child)
Example ABox

CHARLES: Man
EDWARD: Man
ANDREW: Man
DIANA: Mother-without-daughter
(ELIZABETH, CHARLES): has-child
(ELIZABETH, EDWARD): has-child
(ELIZABETH, ANDREW): has-child
(DIANA, WILLIAM): has-child
(CHARLES, WILLIAM): has-child
DIANA: Woman
ELIZABETH: Woman
4 Reasoning Services
Some reasoning services

- Does a description $C$ make sense at all, i.e., is it satisfiable? A concept description $C$ is satisfiable, if there exists an interpretation $\mathcal{I}$ such that $C^{\mathcal{I}} \neq \emptyset$.

- Is one concept a specialization of another one, is it subsumed? $C$ is subsumed by $D$ (in symbols $C \subseteq D$) if we have for all interpretations $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.

- Is $a$ an instance of a concept $C$? $a$ is an instance of $C$ if for all interpretations, we have $a^{\mathcal{I}} \in C^{\mathcal{I}}$.

- **Note**: These questions can be posed with or without a TBox that restricts the possible interpretations.
5 Outlook
Outlook

- Can we reduce the reasoning services to perhaps just one problem?
- What could be reasoning algorithms?
- What can we say about complexity and decidability?
- What has all that to do with modal logics?
- How can one build efficient systems?
Literature I

The Description Logic Handbook: Theory, Implementation, Applications,

Ronald J. Brachman and James G. Schmolze.
An overview of the KL-ONE knowledge representation system.

Terminological Knowledge Representation: A proposal for a terminological logic.
Bernhard Nebel. 
Reasoning and Revision in Hybrid Representation Systems. 
Summary: Concept descriptions

<table>
<thead>
<tr>
<th>Abstract</th>
<th>Concrete</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A)</td>
<td>(A)</td>
<td>(A^I)</td>
</tr>
<tr>
<td>(C \cap D)</td>
<td>(and (C) (D))</td>
<td>(C^I \cap D^I)</td>
</tr>
<tr>
<td>(C \cup D)</td>
<td>(or (C) (D))</td>
<td>(C^I \cup D^I)</td>
</tr>
<tr>
<td>(\neg C)</td>
<td>(not (C))</td>
<td>(D - C^I)</td>
</tr>
<tr>
<td>(\forall r. C)</td>
<td>(all (r) (C))</td>
<td>({d \in D : r^I(d) \subseteq C^I})</td>
</tr>
<tr>
<td>(\exists r)</td>
<td>(some (r))</td>
<td>({d \in D : r^I(d) \neq \emptyset})</td>
</tr>
<tr>
<td>(\geq n r)</td>
<td>(atleast (n) (r))</td>
<td>({d \in D :</td>
</tr>
<tr>
<td>(\leq n r)</td>
<td>(atmost (n) (r))</td>
<td>({d \in D :</td>
</tr>
<tr>
<td>(\exists r. C)</td>
<td>(some (r) (C))</td>
<td>({d \in D : r^I(d) \cap C^I \neq \emptyset})</td>
</tr>
<tr>
<td>(\geq n r. C)</td>
<td>(atleast (n) (r) (C))</td>
<td>({d \in D :</td>
</tr>
<tr>
<td>(\leq n r. C)</td>
<td>(atmost (n) (r) (C))</td>
<td>({d \in D :</td>
</tr>
<tr>
<td>(r = s)</td>
<td>(eq (r) (s))</td>
<td>({d \in D : r^I(d) = s^I(d)})</td>
</tr>
<tr>
<td>(r \neq s)</td>
<td>(neq (r) (s))</td>
<td>({d \in D : r^I(d) \neq s^I(d)})</td>
</tr>
<tr>
<td>(r \subseteq s)</td>
<td>(subset (r) (s))</td>
<td>({d \in D : r^I(d) \subseteq s^I(d)})</td>
</tr>
<tr>
<td>(g = h)</td>
<td>(eq (g) (h))</td>
<td>({d \in D : g^I(d) = h^I(d) \neq \emptyset})</td>
</tr>
<tr>
<td>(g \neq h)</td>
<td>(neq (g) (h))</td>
<td>({d \in D : \emptyset \neq g^I(d) \neq h^I(d) \neq \emptyset})</td>
</tr>
<tr>
<td>({i_1, i_2, \ldots, i_n})</td>
<td>(oneof (i_1 \ldots i_n))</td>
<td>({i_1^I, i_2^I, \ldots, i_n^I})</td>
</tr>
</tbody>
</table>
### Summary: Role descriptions

<table>
<thead>
<tr>
<th>Abstract</th>
<th>Concrete</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t )</td>
<td>( t )</td>
<td>( t^I )</td>
</tr>
<tr>
<td>( f )</td>
<td>( f )</td>
<td>( f^I ), (functional role)</td>
</tr>
<tr>
<td>( r \sqcap s ) (and ( r \ s ))</td>
<td>( r^I \cap s^I )</td>
<td></td>
</tr>
<tr>
<td>( r \sqcup s ) (or ( r \ s ))</td>
<td>( r^I \cup s^I )</td>
<td></td>
</tr>
<tr>
<td>( \neg r ) (not ( r ))</td>
<td>( D \times D - r^I )</td>
<td></td>
</tr>
<tr>
<td>( r^{-1} ) (inverse ( r ))</td>
<td>( {(d, d') : (d', d) \in r^I} )</td>
<td></td>
</tr>
<tr>
<td>( r \mid C ) (restr ( r ) ( C ))</td>
<td>( {(d, d') \in r^I : d' \in C^I} )</td>
<td></td>
</tr>
<tr>
<td>( r^+ ) (trans ( r ))</td>
<td>( (r^I)^+ )</td>
<td></td>
</tr>
<tr>
<td>( r \circ s ) (compose ( r \ s ))</td>
<td>( r^I \circ s^I )</td>
<td></td>
</tr>
<tr>
<td>( 1 ) (self)</td>
<td>( {(d, d) : d \in D} )</td>
<td></td>
</tr>
</tbody>
</table>