Principles of Knowledge Representation and Reasoning

Semantic Networks and Description Logics II: Description Logics – Terminology and Notation

Albert-Ludwigs-Universität Freiburg

Bernhard Nebel, Stefan Wölfl, and Julien Hué

January 29, 2014

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

$\rightsquigarrow$ 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 overriden

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.
- 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 $A^{\mathcal{I}} \subseteq \mathcal{D}$.
- 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 $t^{\mathcal{I}} \subseteq \mathcal{D} \times \mathcal{D}$.

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:

$$C^{\mathcal{I}} = \{d \in \mathcal{D} : C(d) \text{ "is true in" } \mathcal{I}\}$$

$$r^{\mathcal{I}} = \{(d,e) \in \mathcal{D}^2 : r(d,e) \text{ "is true in" } \mathcal{I}\}$$

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:
  - $\texttt{Human} \sqcap \texttt{Female}$
  - $\texttt{Father} \sqcup \texttt{Mother}$
  - $\neg\,\texttt{Female}$
- FOL semantics: $C(x) \wedge D(x)$, $C(x) \vee D(x)$, $\neg C(x)$
- Set semantics: $C^{\mathcal{I}} \cap D^{\mathcal{I}}$, $C^{\mathcal{I}} \cup D^{\mathcal{I}}$, $\mathcal{D} \setminus C^{\mathcal{I}}$

Role restrictions

- Motivation:
  - Often we want to describe something by restricting the possible "fillers" of a role, e.g. `Mother-wod`.
  - 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
  - $\texttt{Mother} \sqcap \forall \texttt{has-child.Man}$
  - $\texttt{Woman} \sqcap \exists \texttt{has-child.Parent}$
- FOL semantics:

$$(\exists r.C)(x) = \exists y (r(x,y) \wedge C(y))$$

$$(\forall r.C)(x) = \forall y (r(x,y) \rightarrow C(y))$$

- Set semantics:

$$(\exists r.C)^{\mathcal{I}} = \{d \in \mathcal{D} : \text{there ex. some } e \text{ s.t. } (d,e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}$$

$$(\forall r.C)^{\mathcal{I}} = \{d \in \mathcal{D} : \text{for each } e \text{ with } (d,e) \in r^{\mathcal{I}},\ e \in C^{\mathcal{I}}\}$$

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:
  - $\texttt{Mother} \sqcap \geq 3\, \texttt{has-child}$
  - $\texttt{Mother} \sqcap \leq 2\, \texttt{has-child}$
- FOL semantics:

$$(\geq n\, r)(x) = \exists y_1 \ldots y_n \big(r(x,y_1) \wedge \cdots \wedge r(x,y_n) \wedge y_1 \neq y_2 \wedge \cdots \wedge y_{n-1} \neq y_n\big)$$
$$(\leq n\, r)(x) = \neg(\geq n+1\; r)(x)$$

- Set semantics:

$$(\geq n\, r)^{\mathcal{I}} = \big\{d \in \mathcal{D} : \big|\{e \in \mathcal{D} : r^{\mathcal{I}}(d,e)\}\big| \geq n\big\}$$
$$(\leq n\, r)^{\mathcal{I}} = \mathcal{D} \setminus (\geq n+1\; r)^{\mathcal{I}}$$

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)
  - $\texttt{has-child}^{-1}$
- Example: $\exists \texttt{has-child}^{-1}.\texttt{Rich}$
- FOL semantics:

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

- Set semantics:

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

UNI FREIBURG
Introduction
Concepts and Roles
Concept Forming Operators
Role Forming Operators
TBox and ABox
Reasoning Services
Outlook
Literature
Appendix

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)
  - $\texttt{has-child} \circ \texttt{has-child}$
- FOL semantics:

$$(r \circ s)(x,y) = \exists z (r(x,z) \wedge s(z,y))$$

- Set semantics:

$$(r \circ s)^{\mathcal{I}} = \big\{(d,e) \in \mathcal{D}^2 : \exists f \text{ s.t. } (d,f) \in r^{\mathcal{I}} \wedge (f,e) \in s^{\mathcal{I}}\big\}$$

UNI FREIBURG
Introduction
Concepts and Roles
Concept Forming Operators
Role Forming Operators
TBox and ABox
Reasoning Services
Outlook
Literature
Appendix

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
  - $\texttt{Woman} \sqcap (\texttt{has-child} \circ \texttt{has-friend} \sqsubseteq \texttt{knows})$
- FOL semantics:

$$(r \sqsubseteq s)(x) = \forall y \big(r(x,y) \rightarrow s(x,y)\big)$$

- Set semantics: Let $r^{\mathcal{I}}(d) = \{e : r^{\mathcal{I}}(d,e)\}$.

$$(r \sqsubseteq s)^{\mathcal{I}} = \big\{d \in \mathcal{D} : r^{\mathcal{I}}(d) \subseteq s^{\mathcal{I}}(d)\big\}$$

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

UNI FREIBURG
Introduction
Concepts and Roles
Concept Forming Operators
Role Forming Operators
TBox and ABox
Reasoning Services
Outlook
Literature
Appendix

3 TBox and ABox

- Terminology Box
- Assertional Box
- Example

Terminology box

- In order to introduce new terms, we use two kinds of terminological axioms:
  - $A \doteq C$
  - $A \sqsubseteq C$

  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 $A \doteq C$, $A \sqsubseteq D$
  - no cyclic definitions (even not indirectly), such as $A \doteq \forall r\,.\,B$, $B \doteq \exists s\,.\,A$

TBoxes: semantics

- TBoxes restrict the set of possible interpretations.
- FOL semantics:
  - $A \doteq C$ corresponds to $\forall x \big(A(x) \leftrightarrow C(x)\big)$
  - $A \sqsubseteq C$ corresponds to $\forall x \big(A(x) \rightarrow C(x)\big)$
- Set semantics:
  - $A \doteq C$ 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:
  - $a^{\mathcal{I}} \in D$
  - $a : C$ corresponds to $a^{\mathcal{I}} \in C^{\mathcal{I}}$
  - $(a,b) : r$ corresponds to $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$
- Models of an ABox and of ABox + TBox can be defined analogously to models of a TBox.

Example TBox

$$
\begin{aligned}
\texttt{Male} &\doteq \neg\texttt{Female}\\
\texttt{Human} &\sqsubseteq \texttt{Living\_entity}\\
\texttt{Woman} &\doteq \texttt{Human} \sqcap \texttt{Female}\\
\texttt{Man} &\doteq \texttt{Human} \sqcap \texttt{Male}\\
\texttt{Mother} &\doteq \texttt{Woman} \sqcap \exists\texttt{has-child.Human}\\
\texttt{Father} &\doteq \texttt{Man} \sqcap \exists\texttt{has-child.Human}\\
\texttt{Parent} &\doteq \texttt{Father} \sqcup \texttt{Mother}\\
\texttt{Grandmother} &\doteq \texttt{Woman} \sqcap \exists\texttt{has-child.Parent}\\
\texttt{Mother-without-daughter} &\doteq \texttt{Mother} \sqcap \forall\texttt{has-child.Male}\\
\texttt{Mother-with-many-children} &\doteq \texttt{Mother} \sqcap (\geq 3\,\texttt{has-child})
\end{aligned}
$$

Example ABox


```
CHARLES:   Man                          DIANA: Woman
EDWARD:    Man                          ELIZABETH: Woman
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
```


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 $C \sqsubseteq D$) 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

Baader, F., D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider.
The Description Logic Handbook: Theory, Implementation, Applications,
Cambridge University Press, Cambridge, UK, 2003.

Ronald J. Brachman and James G. Schmolze.
An overview of the KL-ONE knowledge representation system.
Cognitive Science, 9(2):171–216, April 1985.

Franz Baader, Hans-Jürgen Bürckert, Jochen Heinsohn, Bernhard Hollunder, Jürgen Müller, Bernhard Nebel, Werner Nutt, and Hans-Jürgen Profitlich.
Terminological Knowledge Representation: A proposal for a terminological logic.
Published in Proc. **International Workshop on Terminological Logics**, 1991, DFKI Document D-91-13.

Literature II

Bernhard Nebel.
Reasoning and Revision in Hybrid Representation Systems.
Lecture Notes in Artificial Intelligence 422. Springer-Verlag, Berlin, Heidelberg, New York, 1990.

Summary: Concept descriptions

Abstract	Concrete	Interpretation
A	A	$A^{\mathcal{I}}$
$C \sqcap D$	(and C D)	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
$C \sqcup D$	(or C D)	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
$\neg C$	(not C)	$\mathcal{D} - C^{\mathcal{I}}$
$\forall r.C$	(all r C)	$\{d \in \mathcal{D} : r^{\mathcal{I}}(d) \subseteq C^{\mathcal{I}}\}$
$\exists r$	(some r)	$\{d \in \mathcal{D} : r^{\mathcal{I}}(d) \neq \emptyset\}$
$\geq n\, r$	(atleast n r)	$\{d \in \mathcal{D} : \lvert r^{\mathcal{I}}(d)\rvert \geq n\}$
$\leq n\, r$	(atmost n r)	$\{d \in \mathcal{D} : \lvert r^{\mathcal{I}}(d)\rvert \leq n\}$
$\exists r.C$	(some r C)	$\{d \in \mathcal{D} : r^{\mathcal{I}}(d) \cap C^{\mathcal{I}} \neq \emptyset\}$
$\geq n\, r.C$	(atleast n r C)	$\{d \in \mathcal{D} : \lvert r^{\mathcal{I}}(d) \cap C^{\mathcal{I}}\rvert \geq n\}$
$\leq n\, r.C$	(atmost n r C)	$\{d \in \mathcal{D} : \lvert r^{\mathcal{I}}(d) \cap C^{\mathcal{I}}\rvert \leq n\}$
$r \doteq s$	(eq r s)	$\{d \in \mathcal{D} : r^{\mathcal{I}}(d) = s^{\mathcal{I}}(d)\}$
$r \neq s$	(neq r s)	$\{d \in \mathcal{D} : r^{\mathcal{I}}(d) \neq s^{\mathcal{I}}(d)\}$
$r \sqsubseteq s$	(subset r s)	$\{d \in \mathcal{D} : r^{\mathcal{I}}(d) \subseteq s^{\mathcal{I}}(d)\}$
$g \doteq h$	(eq g h)	$\{d \in \mathcal{D} : g^{\mathcal{I}}(d) = h^{\mathcal{I}}(d) \neq \emptyset\}$
$g \neq h$	(neq g h)	$\{d \in \mathcal{D} : \emptyset \neq g^{\mathcal{I}}(d) \neq h^{\mathcal{I}}(d) \neq \emptyset\}$
$\{i_1, i_2, \ldots, i_n\}$	(oneof $i_1 \ldots i_n$)	$\{i_1^{\mathcal{I}}, i_2^{\mathcal{I}}, \ldots, i_n^{\mathcal{I}}\}$

Summary: Role descriptions

Abstract	Concrete	Interpretation
t	t	$t^{\mathcal{I}}$
f	f	$f^{\mathcal{I}}$, **(functional role)**
$r \sqcap s$	(and r s)	$r^{\mathcal{I}} \cap s^{\mathcal{I}}$
$r \sqcup s$	(or r s)	$r^{\mathcal{I}} \cup s^{\mathcal{I}}$
$\neg r$	(not r)	$\mathcal{D} \times \mathcal{D} - r^{\mathcal{I}}$
r^{-1}	(inverse r)	$\{(d,d') : (d',d) \in r^{\mathcal{I}}\}$
$r\lvert_C$	(restr r C)	$\{(d,d') \in r^{\mathcal{I}} : d' \in C^{\mathcal{I}}\}$
r^+	(trans r)	$(r^{\mathcal{I}})^+$
$r \circ s$	(compose r s)	$r^{\mathcal{I}} \circ s^{\mathcal{I}}$
1	self	$\{(d,d) : d \in \mathcal{D}\}$