Extending the Description Horn Logic DHL

Linh Anh Nguyen

Institute of Informatics
University of Warsaw

CS&P’09
Kraków-Przegorzały, September 2009
Outline

1. Introduction and Motivations
2. The Description Horn Logic DHL
3. Extensions of DHL
   - EDHL
   - EDHL-Datalog
   - GDHL
Description Logics

- Description logics (DLs) are a family of knowledge representation languages which can be used to represent the terminological knowledge of an application domain in a structured and formally well-understood way.
- DLs describe domain in terms of concepts (classes), roles (relationships) and individuals.
- DLs, as decidable fragments of FOL, have well-defined formal semantics.
- DLs are used as the main formal background for a number of languages used in the semantic web technology, including ontology engineering, reasoning with ontology-based markup (meta-data), service description and discovery.
Motivations

Problem

- The data complexity of the DL \( \mathcal{ALC} \) is NP-complete.
- It is worth studying formalisms with \( \text{PTIME} \) data complexity.

Horn Fragments of Logic

- Horn rules are widely used in knowledge representation.
- Horn fragments usually have
  - a lower complexity or data complexity
  - efficient computational methods.

Approaches

- intersection of a DL with the Horn fragment of FOL
- combination of Horn fragments of a DL and FOL


**Knowledge base**

- **RBox** (axioms about roles)
  
  \[
  \text{hasChild} \sqsubseteq \text{hasDescendant} \\
  \text{hasDescendant} \circ \text{hasDescendant} \sqsubseteq \text{hasDescendant} \\
  \text{hasParent} = \text{hasChild}^\sim
  \]

- **TBox** (definitions of concepts and terminological axioms)
  
  \[
  \text{Parent} = \exists \text{hasChild}. \top \\
  \text{Father} = \text{Parent} \sqcap \text{Male} \\
  \text{Mother} = \text{Parent} \sqcap \text{Female}
  \]

- **ABox** (assertions about instances)
  
  \[
  \text{John} : \text{Father} \\
  \text{Mary} : \text{Mother} \\
  \text{hasChild}(\text{John}, \text{Jack})
  \]

- Inference system

- Interface
Introduction and Motivations

The Description Horn Logic DHL

Extensions of DHL

Description Logic $\mathcal{SHI}$ (1)

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Example</th>
<th>Semantics w.r.t. $\mathcal{I} = \langle \Delta^\mathcal{I}, \cdot^\mathcal{I} \rangle$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>John</td>
<td>$a^\mathcal{I} \in \Delta^\mathcal{I}$</td>
</tr>
<tr>
<td>$A$</td>
<td>Human</td>
<td>$A^\mathcal{I} \subseteq \Delta^\mathcal{I}$</td>
</tr>
<tr>
<td>$r$</td>
<td>hasChild</td>
<td>$r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$</td>
</tr>
<tr>
<td>$r^-$</td>
<td>hasChild$^-$</td>
<td>$(r^-)^\mathcal{I} = {(x, y) \mid (y, x) \in r^\mathcal{I}}$</td>
</tr>
</tbody>
</table>

We use letters like $R$, $S$ to denote a role of the form $r$ or $r^-$. “

- $C \sqcap D$  
  $\text{Human} \sqcap \text{Male}$  
  $C^\mathcal{I} \cap D^\mathcal{I}$

- $C \sqcup D$  
  $\text{Mother} \sqcup \text{Father}$  
  $C^\mathcal{I} \cup D^\mathcal{I}$

- $\neg C$  
  $\neg \text{Male}$  
  $\Delta^\mathcal{I} \setminus C^\mathcal{I}$

- $\exists R.C$  
  $\exists \text{hasChild}.\text{Human}$  
  $\{x \mid \exists y. (x, y) \in R^\mathcal{I} \land y \in C^\mathcal{I}\}$

- $\forall R.C$  
  $\forall \text{hasChild}.\text{Doctor}$  
  $\{x \mid \forall y. (x, y) \in R^\mathcal{I} \rightarrow y \in C^\mathcal{I}\}$
### Description Logic $SHI$ (2)

#### RBox (axioms about roles)

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Example</th>
<th>Semantics w.r.t. $\mathcal{I}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R \sqsubseteq S$</td>
<td>$\text{hasChild} \sqsubseteq \text{hasDescendant}$</td>
<td>$R^\mathcal{I} \subseteq S^\mathcal{I}$</td>
</tr>
<tr>
<td>$R \circ R \sqsubseteq R$</td>
<td>$\ldots$</td>
<td>$R^\mathcal{I}$ is transitive</td>
</tr>
</tbody>
</table>

#### TBox (terminological axioms)

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Example</th>
<th>Semantics w.r.t. $\mathcal{I}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C \sqsubseteq D$</td>
<td>$\text{Mother} \sqsubseteq \text{Parent}$</td>
<td>$C^\mathcal{I} \subseteq D^\mathcal{I}$</td>
</tr>
<tr>
<td>$C = D$</td>
<td>$\text{Parent} = \text{Mother} \sqcup \text{Father}$</td>
<td>$C^\mathcal{I} = D^\mathcal{I}$</td>
</tr>
</tbody>
</table>

#### ABox (assertions)

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Example</th>
<th>Semantics w.r.t. $\mathcal{I}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a : A$</td>
<td>$\text{John} : \text{Father}$</td>
<td>$a^\mathcal{I} \in A^\mathcal{I}$</td>
</tr>
<tr>
<td>$R(a, b)$</td>
<td>$\text{hasChild}(\text{John}, \text{Jack})$</td>
<td>$(a^\mathcal{I}, b^\mathcal{I}) \in R^\mathcal{I}$</td>
</tr>
</tbody>
</table>
The Description Horn Logic DHL

DHL was introduced by Grosof et al. as a fragment of DL \textit{SHI} with the following restriction:

- a TBox is a finite set of axioms of the form

\[ C_b \sqsubseteq C_h \text{ or } \top \sqsubseteq \forall R. C_h \]

where \( C_b \) and \( C_h \) are defined by the following BNF grammar:

\[ C_h ::= A \mid C_h \sqcap C_h \mid \forall R. C_h \]

\[ C_b ::= A \mid C_b \sqcap C_b \mid C_b \sqcup C_b \mid \exists R. \top \mid \exists R. C_b \]
The Instance Checking Problem in DHL

Is an individual \( a \) an instance of a concept \( C_b \) w.r.t. a DHL knowledge base \((R, T, A)\) ?

- i.e., is \( a^\mathcal{I} \in C^\mathcal{I}_b \) for all model \( \mathcal{I} \) of \((R, T, A)\) ?

Data Complexity

The data complexity of the instance checking problem is measured w.r.t. to the size of \( A \), assuming that \( R, T, C_b \) and \( a \) are fixed.

Theorem (Grosof et al)

The instance checking problem in DHL has \( \text{PTime} \) data complexity.
$\mathcal{ALCI}_{cf}$ extends $\mathcal{SHI}$ by allowing role axioms of the form

$$R_1 \circ \ldots \circ R_k \sqsubseteq S$$

where $k \geq 0$ and the l.h.s. stands for the identity relation if $k = 0$.

The semantics of such an axiom w.r.t. an interpretation $\mathcal{I}$ is that

$$R_1^\mathcal{I} \circ \ldots \circ R_k^\mathcal{I} \sqsubseteq S^\mathcal{I}$$
EDHL extends DHL by allowing role axioms of the form

$$R_1 \circ \ldots \circ R_k \subseteq S$$
Consider the binary tree

```
  a
 /\ \\
/  \  \
 b   c
|   |
|___|
 d   e
```

specified by the following ABox

\[ \mathcal{A} = \{ \text{root}(a), L(a, b), R(a, c), L(b, d), R(b, e), \text{middle\_level}(b) \} \]

where

- \( L(x, y) \) stands for “\( y \) is the left successor of \( x \)”
- \( R(x, y) \) stands for “\( y \) is the right successor of \( x \)”.

Let $\mathcal{R}$ be the following RBox:

$$L \sqsubseteq S \quad id \sqsubseteq T$$
$$R \sqsubseteq S \quad S^{-} \circ T \circ S \sqsubseteq T$$

This RBox defines $S$ and $T$ to be the roles such that

- $S(x, y)$ means $y$ is a successor of $x$
- $T(x, y)$ means $x$ and $y$ are at the same level of the tree.

**Note:** $\mathcal{R}$ is not a “regular” RBox.
Let $T$ be the TBox consisting of the following axioms:

\[
\exists T. \exists S.T \sqsubseteq inner\_level \\
middle\_level \sqsubseteq \forall T.\middle\_level \\
\text{root} \sqsubseteq \text{on\_left\_path} \\
\exists L^- .\text{on\_left\_path} \sqsubseteq \text{on\_left\_path}
\]

which means

\[
T(x, y) \land S(y, z) \rightarrow \text{inner\_level}(x) \\
\text{middle\_level}(x) \land T(x, y) \rightarrow \text{middle\_level}(y) \\
\text{root}(x) \rightarrow \text{on\_left\_path}(x) \\
L(y, x) \land \text{on\_left\_path}(y) \rightarrow \text{on\_left\_path}(x)
\]
EDHL: Example (4)

\[
\begin{array}{c}
L \sqsubseteq S \\
R \sqsubseteq S \\
\exists T. \exists S . T \\
\text{middle\_level} \\
\text{root} \\
\exists L^{-}. \text{on\_left\_path} \\
\text{root}(a), L(a, b), R(a, c), L(b, d), R(b, e), \text{middle\_level}(b)
\end{array}
\]

It can be shown that

- \( c \) is an instance of concept \text{middle\_level}
- \( a, b \) and \( d \) are instances of concept \text{on\_left\_path}
A Datalog knowledge base consists of
- **extensional part**: a set of facts (ground atoms) in FOL
- **intensional part**: a Datalog program, i.e. a logic program in FOL without negation and function symbols, satisfying the “range-restrictedness” condition.

A query to a Datalog knowledge base is a conjunction of (relational) atoms. **Answers** for a query are defined as usual.

The instance checking problem in EDHL is reducible to the query answering problem in Datalog.

The instance checking problem in EDHL has **PTime** data complexity.
In the CARIN approach, a knowledge base consists of:

- **the bottom layer**: a terminology for defining concepts and roles
- **the top layer**: a logic program of FOL for defining other predicates.
Datalog-Like Knowledge Bases Using EDHL

EDHL-Datalog

- A knowledge base \((R, T, P, A)\) in EDHL-Datalog consists of:
  - an RBox \(R\)
  - an EDHL TBox \(T\)
  - a Datalog program \(P\), which may use concept names as unary predicates and role names as binary predicates in bodies of program clauses
  - a set \(A\) of ground facts (i.e. ground atomic formulas).

- A query to a knowledge base in EDHL-Datalog is a conjunction of (relational) atoms, as in the case of Datalog.

- Answers to a query are defined as usual.
EDHL-Datalog: Example

\[ R = \emptyset \]

\[ T = \{ \exists hasChild. \top \sqsubseteq parent, \]
\[ \quad \text{parent} \sqcap \text{male} \sqsubseteq \text{father}, \]
\[ \quad \text{parent} \sqcap \text{female} \sqsubseteq \text{mother} \} \]

\[ P = \{ \text{father}(x) \land hasChild(x, y) \land \text{age}(y, k) \land k \leq 3 \rightarrow \text{discount}(x, 10), \]
\[ \quad \text{mother}(x) \land hasChild(x, y) \land \text{age}(y, k) \land k \leq 3 \rightarrow \text{discount}(x, 15) \} \]

\[ A = \{ \text{female}(\text{Jane}), \text{male}(\text{Mike}), \text{male}(\text{Peter}), \]
\[ \quad \text{hasChild}(\text{Jane}, \text{Peter}), \text{hasChild}(\text{Mike}, \text{Peter}), \text{age}(\text{Peter}, 2) \} \]

where \( \leq \) is a special predicate with the usual semantics.
Theorem

- A knowledge base in EDHL-Datalog can effectively be translated into an equivalent knowledge base in Datalog.
- The data complexity of EDHL-Datalog is in PTIME.
GDHL: A Further Extension with Function Symbols

GDHL extends EDHL-Datalog by:

- **allowing the constructor** $\exists R.C$ **to appear in the r.h.s. of terminological inclusion axioms** $C_b \subseteq C_h$
  
  - for example:
    
    $\text{parent} \subseteq \exists \text{hasChild}. \top$
    
    $\exists \text{hasChild}. \top \subseteq \text{parent}$

- **using definite logic programs (of FOL) instead of Datalog programs**
  
  - for example:
    
    $\text{father}(x) \land \text{hasChild}(x, y) \land \text{age}(y, k) \land \text{leq}(k, s^3(0)) \rightarrow \text{discount}(x, 10)$
GDHL: Definition

A GDHL TBox is a finite set of axioms of the form $C_b \subseteq C_h$, where $C_b$ and $C_h$ are defined by the following BNF grammar:

\[
C_h ::= A \mid C_h \cap C_h \mid \exists R.T \mid \exists R.C \mid \forall R.C_h
\]
\[
C_b ::= \top \mid A \mid C_b \cap C_b \mid C_b \cup C_b \mid \exists R.C_b
\]

A knowledge base in GDHL is a tuple $(\mathcal{R}, \mathcal{T}, \mathcal{P})$, where

- $\mathcal{R}$ is an RBox
- $\mathcal{T}$ is a GDHL TBox
- $\mathcal{P}$ is a definite logic program, which may use concept names as unary predicates and role names as binary predicates (only) in the bodies of its program clauses.
GDHL: Properties

Proposition

A knowledge base in GDHL can effectively be translated into an equivalent definite logic program.
Conclusions

We have formulated the useful extensions EDHL, EDHL-Datalog and GDHL of DHL.

- Each of these extensions is more expressive than the previous.
- None of them was studied before.

The instance checking problem in EDHL has $\text{PTIME}$ data complexity.

The query language EDHL-Datalog has $\text{PTIME}$ data complexity.

The translation from EDHL-Datalog into Datalog allows using efficient computational methods of Datalog.

EDHL-Datalog is more convenient than Datalog for Semantic Web.

Answering a query to a knowledge base in GDHL is reducible to answering a query to a definite logic program, for which advanced methods can be used.
Future Work

We intend to extend EDHL and EDHL-Datalog by

- allowing number restrictions and negation to appear in the left hand side of terminological inclusion axioms
- allowing negation to appear in bodies of Datalog program clauses
- using stratified model semantics or well-founded semantics to deal with negation.