How to reason with OWL in a logic programming system

Markus Krötzsch  Denny Vrandecic
AIFB Karlsruhe  AIFB Karlsruhe

Pascal Hitzler  Michael Sintek
AIFB Karlsruhe  DFKI Kaiserslautern

RuleML2006, Athens, GA
November 2006
Semantic Web Languages: Separate worlds

- OWL DL
  - open world
  - monotonic
  - description logics
  - first-order logic
  - decidable

- Logic Programming
  - closed world
  - non-monotonic
  - rules
  - procedural flavour
  - undecidable

- both approaches are needed for applications
- study of interoperability is imperative
- here: sound and complete reasoning for OWL with Prolog
Approach

• We utilize results by Motik et al. on the KAON2 transformation algorithms and system.

• KAON2 OWL reasoner: http://kaon2.semanticweb.org

KAON2 Reasoner core architecture

Query

SHIQ(D) TBox (no nominals)

Transformation to Disjunctive Datalog [ExpTime]

SHIQ(D) ABox

Disjunctive Datalog Reasoning Engine [coNP]

Answer

suffices for some queries e.g. instance retrieval for named classes
Theorem (Hustadt, Motik, Sattler 2004)

Transformation of OWL knowledge base KB into Disjunctive Datalog DD(KB)

Then, the following hold:

1. KB is unsatisfiable if and only if DD(KB) is unsatisfiable.
2. KB ⊨ α if and only if DD(KB) ⊨ α, where α is of the form A(a) or R(a, b), and A is an atomic concept.
3. KB ⊨ C(a) for a nonatomic concept C if and only if, for Q a new atomic concept, DD(KB ∪ {C ⊑ Q}) ⊨ Q(a).
## Simple example transformation (ALC)

### KB

<table>
<thead>
<tr>
<th>Rule 1</th>
<th>Rule 2</th>
<th>Rule 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person ⊑ ∃ parent.Person</td>
<td>∃ parent.(∃ parent.Person) ⊑ Grandchild</td>
<td>Person(a)</td>
</tr>
</tbody>
</table>

### DD(KB)

<table>
<thead>
<tr>
<th>Rule 1</th>
<th>Rule 2</th>
<th>Rule 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q₁(x), Person(y) ← parent(x,y)</td>
<td>← parent(x,y), Q₁(y), Grandchild(x)</td>
<td>← Q₁(x), Person(x)</td>
</tr>
<tr>
<td>Grandchild(x) ← Person(x)</td>
<td>Person(a)</td>
<td></td>
</tr>
</tbody>
</table>
KAON2 Reasoner core architecture: Horn-SHIQ

Query

OWL DL TBox
(no nominals)

Transformation to
Disjunctive Datalog
[ExpTime]

Horn-SHIQ

OWL DL ABox

Disjunctive Datalog Reasoning Engine [coNP] [P]

Answer

suffices for some queries e.g. instance retrieval for named classes
Horn-SHIQ

- Fragment of OWL DL
  - Polynomial data complexity (ABox)
  - ExpTime combined complexity (ABox+TBox)
[OWLED06]

\[\begin{align*}
C_0^+ & \rightarrow T \mid \bot \mid \neg C_0^- \mid C_0^+ \cap C_0^+ \mid C_0^+ \cup C_0^+ \mid \forall R.C_0^+ \\
C_0^- & \rightarrow T \mid \bot \mid \neg C_0^+ \mid C_0^- \cap C_0^- \mid C_0^- \cup C_0^- \mid \exists R.C_0^- \mid A \\
C_1^+ & \rightarrow T \mid \bot \mid \neg C_1^- \mid C_1^+ \cap C_1^+ \mid C_0^+ \cup C_1^+ \mid \exists R.C_1^+ \mid \forall R.C_1^+ \mid \geq n R.C_1^+ \mid \leq 1 R.C_0^- \mid A \\
C_1^- & \rightarrow T \mid \bot \mid \neg C_1^+ \mid C_0^- \cup C_1^- \mid C_1^- \cup C_1^- \mid \exists R.C_1^- \mid \forall R.C_1^- \mid \geq 2 R.C_0^- \mid \leq n R.C_1^+ \mid A
\end{align*}\]
KAON2 Reasoner core architecture:

Query → Transformation to Disjunctive Datalog [ExpTime] → Disjunctive Datalog Reasoning Engine [coNP] [P] → Answer

Horn-SHIQ

OWL DL ABox

suffices for some queries e.g. instance retrieval for named classes

Using Prolog
Difficultly: Integrity constraints

- Some OWL statements become integrity constraints which are not usually supported under Prolog.

- $$C \sqcap D \equiv \bot$$ translates to $$\leftarrow C(x) \land D(x)$$

- workaround: $$\text{inc} \leftarrow C(x) \land D(x)$$
Difficulty: Equality

- Some OWL statements require equality for expressing them in first-order logic.

- For our purposes, the following Horn rules suffice:

\[
X \approx X, \quad X \approx Y \leftarrow Y \approx X, \quad X \approx Z \leftarrow X \approx Y \land Y \approx Z \\
C(Y) \leftarrow C(X) \land X \approx Y \quad \text{for every concept name } C \\
R(Y_1, Y_2) \leftarrow R(X_1, X_2) \land X_1 \approx Y_1 \land X_2 \approx Y_2 \quad \text{for every role name } R
\]
Example
TBox/RBox

(1) \text{Parent} \equiv \exists \text{hasChild.}\top

(2) \text{Person} \sqsubseteq \exists \text{childOf}.\text{Person}

(3) \text{ManyChildren} \sqsubseteq \geq 2 \text{hasChild.}\top

(4) \text{NoSiblings} \sqsubseteq \text{Person} \cap \forall \text{childOf.}(\leq 1 \text{hasChild.}\top)

(5) \text{childOf} \equiv \text{hasChild}^{-1}

\begin{align*}
\text{person}(X) & :\neg \text{nosiblings}(X). \\
\text{parent}(X) & :\neg \text{haschild}(X, Y). \\
\text{haschild}(Y, X) & :\neg \text{childof}(X, Y). \\
\text{haschild}(X, X_f2) & :\neg \text{parent}(X), \text{S}_f2(X, X_f2). \\
\text{childof}(X, X_f3) & :\neg \text{person}(X), \text{S}_f3(X, X_f3). \\
Y_1 \approx Y_2 & :\neg \text{nosiblings}(X), \text{childof}(X, Y), \text{haschild}(Y, X_1), \text{haschild}(Y, Y_2). \\
\text{inc} & :\neg \text{manychildren}(X), \text{nosiblings}(X_0), \text{childof}(X_0, X). \\
\text{inc} & :X_f1 \approx X_f0, \text{manychildren}(X), \text{S}_f1(X, X_f1), \text{S}_f0(X, X_f0).
\end{align*}

\begin{align*}
\text{S}_f(X, f(X)) & :\neg \text{O}(X). \\
\text{HU}(X) & :\neg \text{O}(X). \\
\text{HU}(f(X)) & :\neg \text{O}(X). \\
\text{(for } f \in \{f_0, f_1, f_2, f_3\})
\end{align*}

\begin{align*}
X \approx X & :\neg \text{HU}(X). \\
X \approx Y & :Y \approx X, \text{HU}(X), \text{HU}(Y). \\
X \approx Z & :X \approx Y, Y \approx Z, \text{HU}(X), \text{HU}(Y), \text{HU}(Z). \\
\text{C}(Y) & :\neg \text{C}(X), X \approx Y, \text{HU}(X), \text{HU}(Y). \\
\text{(for } C \in \{\text{person}, \text{parent}, \text{manychildren}, \text{nosiblings}\})
\end{align*}

\begin{align*}
\text{R}(Y_1, Y_2) & :\neg \text{R}(X_1, X_2), X_1 \approx Y_1, X_2 \approx Y_2, \text{HU}(X_1), \text{HU}(X_2), \text{HU}(Y_1), \text{HU}(X_2). \\
\text{(for } R \in \{\text{childof}, \text{haschild}\})
\end{align*}

\begin{align*}
\text{O}(\text{Elaine}). & \quad \text{O}(\text{Sir Lancelot}). \quad \text{O}(\text{Lancelot du Lac}).
\end{align*}
Implementation

Transformation available through KAON2
http://kaon2.semanticweb.org

or via owltools command line interface dlpconvert
http://owltools.ontoware.org
see software demo [OWLED06] this evening
– optional serialisations:
  • Prolog
  • F-Logic
  • RuleML0.9
  • SWRL
Acknowledgement

• The presented results are corollaries from the work by Boris Motik on KAON2.

• Very helpful discussions with Boris are gratefully acknowledged.
Thank you!

- related [OWLED06]-presentations:
  - Today, 1345 hrs:
    M. Krötzsch, S. Rudolph and P. Hitzler. On the Complexity of Horn Description Logics
  - Today, 1700 hrs:
    D. Vrandecic: OWL Tools demo