Recent Advances Concerning OWL and Rules

Pascal Hitzler
Kno.e.sis Center
Wright State University, Dayton, OH
http://www.knoesis.org/pascal/
Textbook

Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph

Foundations of Semantic Web Technologies

Chapman & Hall/CRC, 2010

Choice Magazine Outstanding Academic Title 2010 (one out of seven in Information & Computer Science)

http://www.semantic-web-book.org
Textbook – Chinese translation

Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph

语义Web技术基础
Tsinghua University Press (清华大学出版社)，2013.

Translators:
Yong Yu, Haofeng Wang, Guilin Qi (俞勇，王昊奋，漆桂林)

http://www.semantic-web-book.org
Semantic Web journal

- EiCs: Pascal Hitzler
  Krzysztof Janowicz

- New journal with significant initial uptake.

- We very much welcome contributions at the “rim” of traditional Semantic Web research – e.g., work which is strongly inspired by a different field.

- Non-standard (open & transparent) review process.

- http://www.semantic-web-journal.net/
The Kno.e.sis Center and My Lab

- Ohio Center of Excellence in Knowledge-enabled Computing
  Director: Amit Sheth
- 15 faculty (8 in Computer Science) across 4 Departments, with ca. 50 PhD students

- Knowledge Engineering Lab (since January 2010)
  Led by myself

Currently
- 8 PhD students
- 2 Master students
- 3 undergrads

- http://www.knoesis.org/
The Semantic Web Stack
1. Description Logics and OWL
2. Rules expressible in description logics
3. Extending description logics with rules through nominal schemas
4. Algorithmizations for nominal schemas
5. Adding non-monotonicity
6. Conclusions
Web Ontology Language (OWL)

- W3C Recommendation since 2004
- OWL 2 since 2009

- based on description logics
- essentially, a decidable fragment of first-order predicate logic
<table>
<thead>
<tr>
<th>Description Logics (DLs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>classes/concepts</td>
</tr>
<tr>
<td>A, B, C</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>roles/properties</td>
</tr>
<tr>
<td>R, S</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>individuals</td>
</tr>
<tr>
<td>a, b, c</td>
</tr>
</tbody>
</table>
Some DL constructors

class conjunction
\[ C \cap D \quad \rightarrow \quad C(x) \land D(x) \]

eXistential restriction
\[ \exists R.C \quad \rightarrow \quad \exists y \ (R(x,y) \land C(y)) \]

class inclusion/subsumption
\[ C \subseteq D \quad \rightarrow \quad C(x) \rightarrow D(x) \]
\[ C \equiv D \quad \rightarrow \quad C(x) \leftrightarrow D(x) \]

role chains
\[ R_1 \circ \ldots \circ R_n \subseteq R \quad \rightarrow \quad R_1(x,x_1) \land \ldots \land R(x_n,x_{n+1}) \rightarrow R(x,x_{n+1}) \]
Some DL constructors

ThaiDish ⊑ ∃contains.Nut
Nutallergic ∩ ∃eats.Nut ⊑ Unhappy
eats ◦ contains ⊑ eats

inverse roles

$R ≅ S^\sim$

$R(x,y) ⇔ S(y,x)$

This logic is already undecidable! (see e.g. [ISWC 2007])

Name of the logic: ELRI
Decidability is a central characteristics of description logics.
Retaining Decidability

1. Disallow $\exists$:
   Essentially leads to OWL RL.
   Fragment of Datalog.
   Tractable (i.e., polynomial complexity).

2. Disallow inverse roles:
   Essentially leads to OWL EL.
   Akin “in spirit” to existential rules/Datalog+-.
   Tractable.

3. Restrict recursion in role chains (a.k.a. \textit{regularity} restriction):
   With further constructors, leads to OWL DL, a.k.a. SROIQ.
   Decidable, but not tractable.
Further essential DL constructors

The following can be used in OWL EL (logic remains tractable).

Self

\[ C \sqsubseteq \exists R. \text{Self} \quad C(x) \rightarrow R(x, x) \]

Can be used e.g. for typecasting.

nominals

\[
\begin{align*}
\{a\} & \sqsubseteq C & C(a) & \text{a is a constant} \\
C & \sqsubseteq \{a\} & C(x) & \rightarrow x = a \\
\{a\} & \equiv \{b\} & \rightarrow a = b
\end{align*}
\]

\[ A \sqcap \exists R. \{b\} \sqsubseteq C \text{ becomes } A(x) \land R(x, b) \rightarrow C(x) \]
Further essential DL constructors

The following are used in expressive (intractable) DLs

class negation
\[ \neg C \quad \neg C(x) \]

class disjunction
\[ C \sqcup D \quad C(x) \lor D(x) \]

universal restriction
\[ \forall R.C \quad \forall y (R(x,y) \rightarrow C(y)) \]

There are some more of course.
Contents

1. Description Logics and OWL
2. Rules expressible in description logics
3. Extending description logics with rules through nominal schemas
4. Algorithmizations for nominal schemas
5. Adding non-monotonicity
6. Conclusions
Which rules can be encoded in OWL?

\[ A \sqsubseteq B \text{ becomes } A(x) \rightarrow B(x) \]
\[ R \sqsubseteq S \text{ becomes } R(x, y) \rightarrow S(x, y) \]
\[ A \sqcap \exists R.\exists S. B \sqsubseteq C \text{ becomes } A(x) \land R(x, y) \land S(y, z) \land B(z) \rightarrow C(x) \]
\[ \{a\} \equiv \{b\} \text{ becomes } \rightarrow a = b. \]
\[ A \sqcap B \sqsubseteq \bot \text{ becomes } A(x) \land B(x) \rightarrow f. \]
\[ R \circ S \sqsubseteq T \text{ becomes } R(x, y) \land S(y, z) \rightarrow T(x, z) \]
Which rules can be encoded in OWL?

\[ A \sqsubseteq \neg B \sqcup C \text{ becomes } A(x) \land B(x) \rightarrow C(x) \]

\[ A \sqsubseteq \forall R.B \text{ becomes } A(x) \land R(x, y) \rightarrow B(y) \]

\[ A \sqsubseteq B \land C \text{ becomes } A(x) \rightarrow B(x) \text{ and } A(x) \rightarrow C(x) \]

\[ A \sqcup B \rightarrow C \text{ becomes } A(x) \rightarrow C(x) \text{ and } B(x) \rightarrow C(x) \]
Rolification

\[ \text{Elephant}(x) \land \text{Mouse}(y) \rightarrow \text{biggerThan}(x, y) \]

- Rolification of a concept A: \( A \equiv \exists R_A.Self \)

\[
\begin{align*}
\text{Elephant} & \equiv \exists R_{\text{Elephant}}.Self \\
\text{Mouse} & \equiv \exists R_{\text{Mouse}}.Self \\
R_{\text{Elephant}} \circ U \circ R_{\text{Mouse}} & \sqsubseteq \text{biggerThan}.
\end{align*}
\]
Rolification

\[ A(x) \land R(x, y) \rightarrow S(x, y) \text{ becomes } R_A \circ R \sqsubseteq S \]
\[ A(y) \land R(x, y) \rightarrow S(x, y) \text{ becomes } R \circ R_A \sqsubseteq S \]
\[ A(x) \land B(y) \land R(x, y) \rightarrow S(x, y) \text{ becomes } R_A \circ R \circ R_B \sqsubseteq S \]

\[ \text{Woman}(x) \land \text{marriedTo}(x, y) \land \text{Man}(y) \rightarrow \text{hasHusband}(x, y) \]
\[ R_{\text{Woman}} \circ \text{marriedTo} \circ R_{\text{Man}} \sqsubseteq \text{hasHusband} \]

Careful – regularity of RBox needs to be retained:

\[ \text{hasHusband} \sqsubseteq \text{marriedTo} \]
\[ \text{worksAt}(x, y) \land \text{University}(y) \land \text{supervises}(x, z) \land \text{PhDStudent}(z) \rightarrow \text{professorOf}(x, z) \]

\[ R_{\exists \text{worksAt.University} \circ \text{supervises} \circ \text{PhDStudent}} \subseteq \text{professorOf}. \]
Tree-shaped rules

\[ R_1(x, y) \land C_1(y) \land R_2(y, w) \land R_3(y, z) \land C_2(z) \land R_4(x, x) \rightarrow C_3(x) \]

\[ \exists R_1. (C_1 \sqinter \exists R_2. T \sqinter \exists R_3. C_2) \sqinter \exists R_4. Self \sqsubseteq C_3 \]
Acyclic Rules

\[ R_1(y, x) \land C_1(y) \land R_2(w, y) \land R_3(y, z) \land C_2(z) \land R_4(x, x) \rightarrow C_3(x) \]

\[ \exists R_1. (C_1 \sqcap \exists R_2. \top \sqcap \exists R_3. C_2) \sqcap \exists R_4. \text{Self} \sqsubseteq C_3 \]
So how can we pinpoint this?

- Tree-shaped bodies
- First argument of the conclusion is the root

\[ C(x) \land R(x,a) \land S(x,y) \land D(y) \land T(y,a) \rightarrow E(x) \]

\[ C \cap \exists R.\{a\} \cap \exists S.(D \cap \exists T.\{a\}) \subseteq E \]
Rule bodies as graphs

\[ C(x) \land R(x, a) \land S(x, y) \land D(y) \land T(y, a) \rightarrow P(x, y) \]

\[ a_1 \leftarrow x \rightarrow y \rightarrow a_2 \]

\[
\begin{align*}
C \cap \exists R.\{a\} & \subseteq \exists R1.\text{Self} \\
D \cap \exists T.\{a\} & \subseteq \exists R2.\text{Self} \\
R1 \circ S \circ R2 & \subseteq P
\end{align*}
\]
So how can we pinpoint this?

- Tree-shaped bodies
- First argument of the conclusion is the root

\[ C(x) \land R(x,a) \land S(x,y) \land D(y) \land T(y,a) \rightarrow V(x,y) \]

\[ C \sqcap \exists R.\{a\} \subseteq \exists R1.\text{Self} \]
\[ D \sqcap \exists T.\{a\} \subseteq \exists R2.\text{Self} \]
\[ R1 \circ S \circ R2 \subseteq V \]
1. Description Logics and OWL
2. Rules expressible in description logics
3. Extending description logics with rules through nominal schemas
4. Algorithmizations for nominal schemas
5. Adding non-monotonicity
6. Conclusions
Rule bodies as graphs

\[
\begin{align*}
\text{hasReviewAssignment}(v, x) \land \text{hasAuthor}(x, y) \land \text{atVenue}(x, z) \\
\land \text{hasSubmittedPaper}(v, u) \land \text{hasAuthor}(u, y) \land \text{atVenue}(u, z) \\
\rightarrow \text{hasConflictingAssignedPaper}(v, x)
\end{align*}
\]

with \(y, z\) constants:

\[
R \exists \text{hasSubmittedPaper}. (\exists \text{hasAuthor}. \{y\} \land \exists \text{atVenue}. \{z\}) \circ \text{hasReviewAssignment} \\
\circ R \exists \text{hasAuthor}. \{y\} \land \exists \text{atVenue}. \{z\} \\
\sqsubseteq \text{hasConflictingAssignedPaper}
\]
Non-hybrid syntax: nominal schemas

assume $y,z$ bind only to named individuals
we introduce a new construct, called

nominal schemas

or nominal variables
Nominal schema example 2

\[\text{hasChild}(x, y) \land \text{hasChild}(x, z) \land \text{classmate}(y, z) \rightarrow C(x)\]

\[\exists \text{hasChild}.\{z\} \sqcap \exists \text{hasChild}.\exists \text{classmate}.\{z\} \sqsubseteq C\]
Adding nominal schemas to OWL 2 DL

- Decidability is retained.
- Complexity is *the same*.

- A naïve implementation is straightforward:

  Replace every axiom with nominal schemas by a set of OWL 2 axioms, obtained from *grounding* the nominal schemas.

However, this may result in a lot of new OWL 2 axioms. The naïve approach will probably only work for ontologies with *few* nominal schemas.
What do we gain?

- A powerful macro.
- A conceptual bridge to rule formalism:

We can actually also express all DL-safe Datalog rules!

\[ R(x, y) \land A(y) \land S(z, y) \land T(x, z) \rightarrow P(z, x) \]

\[ \exists U. \left( \{x\} \sqcap \exists R. \{y\} \right) \]
\[ \quad \sqcap \exists U. \left( \{y\} \sqcap A \right) \]
\[ \quad \sqcap \exists U. \left( \{z\} \sqcap \exists S. \{y\} \right) \]
\[ \quad \sqcap \exists U. \left( \{x\} \sqcap \exists T. \{z\} \right) \]
\[ \subseteq \exists U. \left( \{z\} \sqcap \exists P. \{x\} \right) \]
Contents

1. Description Logics and OWL
2. Rules expressible in description logics
3. Extending description logics with rules through nominal schemas
4. Algorithmizations for nominal schemas
5. Adding non-monotonicity
6. Conclusions
Naïve implementation – experiments

<table>
<thead>
<tr>
<th></th>
<th>No axioms added</th>
<th>1 different ns</th>
<th>2 different ns</th>
<th>3 different ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fam (5)</td>
<td>0.01”</td>
<td>0.01”</td>
<td>0.01”</td>
<td>0.04”</td>
</tr>
<tr>
<td>Swe (22)</td>
<td>3.58”</td>
<td>3.73”</td>
<td>3.85”</td>
<td>10.86”</td>
</tr>
<tr>
<td>Bui (42)</td>
<td>2.7”</td>
<td>2.5”</td>
<td>2.75”</td>
<td>1’ 14’</td>
</tr>
<tr>
<td>Wor (80)</td>
<td>0.11”</td>
<td>0.12”</td>
<td>1.1”</td>
<td>OOM *</td>
</tr>
<tr>
<td>Tra (183)</td>
<td>0.05”</td>
<td>0.05”</td>
<td>5.66”</td>
<td>OOM</td>
</tr>
<tr>
<td>FTr (368)</td>
<td>0.03”</td>
<td>0.05”</td>
<td>35.53”</td>
<td>OOM</td>
</tr>
<tr>
<td>Eco (482)</td>
<td>0.04”</td>
<td>0.07”</td>
<td>56.59”</td>
<td>OOM</td>
</tr>
</tbody>
</table>

OOM = Out of Memory

from the TONES repository:

<table>
<thead>
<tr>
<th>Ontology</th>
<th>Classes</th>
<th>Data P.</th>
<th>Object P.</th>
<th>Individuals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fam</td>
<td>4</td>
<td>1</td>
<td>11</td>
<td>5</td>
</tr>
<tr>
<td>Swe</td>
<td>189</td>
<td>6</td>
<td>25</td>
<td>22</td>
</tr>
<tr>
<td>Bui</td>
<td>686</td>
<td>0</td>
<td>24</td>
<td>42</td>
</tr>
<tr>
<td>Wor</td>
<td>1842</td>
<td>0</td>
<td>31</td>
<td>80</td>
</tr>
<tr>
<td>Tra</td>
<td>445</td>
<td>4</td>
<td>89</td>
<td>183</td>
</tr>
<tr>
<td>FTr</td>
<td>22</td>
<td>6</td>
<td>52</td>
<td>368</td>
</tr>
<tr>
<td>Eco</td>
<td>339</td>
<td>8</td>
<td>45</td>
<td>482</td>
</tr>
</tbody>
</table>
Delayed grounding

- Adding nominal schemas to existing tableaux algorithms:

\[
\text{grounding: if } C \in L(s), \{z\} \text{ is a nominal schema in } C, \\
C[z/a_i] \notin L(s) \text{ for some } i, 1 \leq i \leq \ell \\
\text{then } L(s) := L(s) \cup \{C[z/a_i]\}
\]

plus some restrictions on existing tableaux rules, essentially to ensure that (1) no variable binding is broken and (2) nominal schemas are not propagated through the tableau.
Further Tableaux Optimizations

• variant of absorption [Steigmiller, Glimm, Liebig, IJCAI-13]
• essentially, a sort of smart rewriting as pre-processing

Example 1

Our running example

\[
\exists r.(\{x\} \sqcap \exists a.\{y\} \sqcap \\
\exists v.\{z\}) \sqcap \exists s.(\exists a.\{y\} \sqcap \exists v.\{z\}) \subseteq \exists c.\{x\}
\]

can be almost completely absorbed into the following axioms:

\[
\begin{align*}
O & \sqsubseteq \downarrow x.T_x \\
O & \sqsubseteq \downarrow y.T_y \\
O & \sqsubseteq \downarrow z.T_z \\
T_y & \sqsubseteq \forall a_.T_1
\end{align*}
\]

\[
\begin{align*}
T_z & \sqsubseteq \forall v_.T_2 & (T_1 \sqcap T_2) & \sqsubseteq T_3 \\
T_3 & \sqsubseteq \forall s_.T_4 & (T_3 \sqcap T_x) & \sqsubseteq T_5 \\
T_5 & \sqsubseteq \forall r_.T_6 & (T_4 \sqcap T_6) & \sqsubseteq T_7
\end{align*}
\]

where \(T_x, T_y, T_z, T_1, \ldots, T_7\) are fresh atomic concepts. Only \(\exists c.\{x\}\) cannot be absorbed and has to be grounded on demand.
Further Tableaux Optimizations

[Steigmiller, Glimm, Liebig, IJCAI-13]

Table 2: DL-safe Rules for UOBM-Benchmarks

<table>
<thead>
<tr>
<th>Name</th>
<th>DL-safe Rule</th>
<th>Matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>isFirendOf(?x, ?y), like(?x, ?z), like(?y, ?z) → hasLink1(?x, ?y)</td>
<td>4,037</td>
</tr>
<tr>
<td>R2</td>
<td>isFirendOf(?x, ?y), takesCourse(?x, ?z), takesCourse(?y, ?z) → hasLink2(?x, ?y)</td>
<td>82</td>
</tr>
<tr>
<td>R3</td>
<td>takesCourse(?x, ?z), takesCourse(?y, ?z), hasSameHomeTownWith(?x, ?y) → hasLink3(?x, ?y)</td>
<td>940</td>
</tr>
<tr>
<td>R4</td>
<td>hasDoctoralDegreeFrom(?x, ?z), hasMasterDegreeFrom(?x, ?w), hasDoctoralDegreeFrom(?y, ?z), hasMasterDegreeFrom(?y, ?w), worksFor(?x, ?v), worksFor(?y, ?v), → hasLink4(?x, ?y)</td>
<td>369</td>
</tr>
<tr>
<td>R5</td>
<td>isAdvisedBy(?x, ?z), isAdvisedBy(?y, ?z), like(?x, ?w), like(?y, ?w), like(?z, ?w) → hasLink5(?x, ?y)</td>
<td>286</td>
</tr>
</tbody>
</table>

Table 3: Comparison of the increases in reasoning time of the consistency tests for UOBM1\D extended by rules in seconds

<table>
<thead>
<tr>
<th>Rule</th>
<th>upfront grounding</th>
<th>direct propagation</th>
<th>representative propagation</th>
<th>HermiT</th>
<th>Pellet</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>without BC</td>
<td>with BC</td>
<td>without BC</td>
<td>with BC</td>
</tr>
<tr>
<td>R1</td>
<td>(10.99) mem</td>
<td>9.12</td>
<td>7.10</td>
<td>5.06</td>
<td>3.38</td>
</tr>
<tr>
<td>R2</td>
<td>(10.92)</td>
<td>4.05</td>
<td>3.33</td>
<td>2.33</td>
<td>2.11</td>
</tr>
<tr>
<td>R3</td>
<td>(13.33)</td>
<td>3.55</td>
<td>1.98</td>
<td>0.62</td>
<td>0.76</td>
</tr>
<tr>
<td>R4</td>
<td>(16.44)</td>
<td>0.30</td>
<td>1.08</td>
<td>0.09</td>
<td>0.07</td>
</tr>
<tr>
<td>R5</td>
<td>(time)</td>
<td>–</td>
<td>1.87</td>
<td>0.50</td>
<td>0.43</td>
</tr>
</tbody>
</table>
Algorithm for ELROVn

Based on [Krötzsch, JELIA10]

<table>
<thead>
<tr>
<th>Ontology</th>
<th>Individuals</th>
<th>no ns</th>
<th>1 ns</th>
<th>2 ns</th>
<th>3 ns</th>
<th>4 ns</th>
<th>5 ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rex (full ground.)</td>
<td>100</td>
<td>263</td>
<td>263 (321)</td>
<td>267 (972)</td>
<td>273</td>
<td>275</td>
<td>259</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>480</td>
<td>518 (1753)</td>
<td>537 (OOM)</td>
<td>538</td>
<td>545</td>
<td>552</td>
</tr>
<tr>
<td></td>
<td>10000</td>
<td>2904</td>
<td>2901 (133179)</td>
<td>3120 (OOM)</td>
<td>3165</td>
<td>3192</td>
<td>3296</td>
</tr>
<tr>
<td>Spatial (full ground.)</td>
<td>100</td>
<td>22</td>
<td>191 (222)</td>
<td>201 (1163)</td>
<td>198</td>
<td>202</td>
<td>207</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>134</td>
<td>417 (1392)</td>
<td>415 (OOM)</td>
<td>421</td>
<td>431</td>
<td>432</td>
</tr>
<tr>
<td></td>
<td>10000</td>
<td>1322</td>
<td>1792 (96437)</td>
<td>1817 (OOM)</td>
<td>1915</td>
<td>1888</td>
<td>1997</td>
</tr>
<tr>
<td>Xenopus (full ground.)</td>
<td>100</td>
<td>62</td>
<td>332 (383)</td>
<td>284 (1629)</td>
<td>311</td>
<td>288</td>
<td>280</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>193</td>
<td>538 (4751)</td>
<td>440 (OOM)</td>
<td>430</td>
<td>456</td>
<td>475</td>
</tr>
<tr>
<td></td>
<td>10000</td>
<td>1771</td>
<td>2119 (319013)</td>
<td>1843 (OOM)</td>
<td>1886</td>
<td>2038</td>
<td>2102</td>
</tr>
</tbody>
</table>
Approximating OWL through ELROVn

- We rewrite mincardinality restrictions into maxcardinality restrictions or approximate using an existential.
- We rewrite universal quantification into existential quantification.
- We approximate maxcardinality restrictions using functionality.
- We approximate inverse roles and functionality using nominal schemas.
- We approximate negation using class disjointness.
- We approximate disjunction using conjunction.

- **inverses:**
  \[
  \{x\} \cap \exists R.\{y\} \subseteq \{y\} \cap \exists S.\{x\}
  \]

- **functionality**
  \[
  C \sqsubseteq 1R.D
  \]

  \[
  C \cap \exists R.\{{z1}\} \cap D \cap \exists R.\{{z2}\} \cap D \subseteq \exists U.\{{z1}\} \cap \{{z2}\}
  \]
## Approximation results (using IRIS)

<table>
<thead>
<tr>
<th>Ontology</th>
<th>HermiT</th>
<th>Fact++</th>
<th>Pellet</th>
<th>Ours</th>
<th>Ours Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAMS</td>
<td>3</td>
<td>2</td>
<td>10</td>
<td>107</td>
<td>100%</td>
</tr>
<tr>
<td>DOLCE</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>53</td>
<td>100%</td>
</tr>
<tr>
<td>GALEN</td>
<td>4</td>
<td>2</td>
<td>17</td>
<td>7840</td>
<td>90.8%</td>
</tr>
<tr>
<td>GO</td>
<td>36</td>
<td>75</td>
<td>59</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>GardinerCorpus</td>
<td>14</td>
<td>6</td>
<td>17</td>
<td>89</td>
<td>92.3%</td>
</tr>
<tr>
<td>OBO</td>
<td>34</td>
<td>61</td>
<td>139</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Contents

1. Initial examples
2. Rules expressible in description logics
3. Extending description logics with rules through nominal schemas
4. Algorithmizations for nominal schemas
5. Adding non-monotonicity
6. Conclusions
Adding non-monotonicity

- [Knorr, Hitzler, Maier ECAI2012]

- Extension of an autoepistemic description logic approach by nominal schemas.

- Results in a language which incorporates most of the major approaches to non-monotonic extensions of DLs.

- E.g. covers
  - hybrid MKNF [Motik & Rosati], which in turn covers
  - non-disjunctive ASP
  - DL Programs / dlvhex (Eiter et al.)

- Also covers OWL / SROIQ(D) of course.
1. Initial examples
2. Rules expressible in description logics
3. Extending description logics with rules through nominal schemas
4. Algorithmizations for nominal schemas
5. Adding non-monotonicity
6. Conclusions
Conclusions

• Paradigms are converging.

• More work needed e.g. re.
  – algorithmizations
  – relating OWL EL and existential rules research
  – making non-monotonic reasoning fit for semantic web applications
Collaborators on the covered topics

David Carral, Kno.e.sis Center, Wright State University
Matthias Knorr, UN Lisboa, Portugal
Adila Krisnadhi, Kno.e.sis Center, Wright State University
Markus Krötzsch, Oxford University, UK
Frederick Maier, Kno.e.sis Center, Wright State University
Sebastian Rudolph, Karlsruhe Institute of Technology, Germany
Kunal Sengupta, Kno.e.sis Center, Wright State University
Cong Wang, Kno.e.sis Center, Wright State University
References

A tutorial:


Background reading:


References


References


References


References


References

- Benjamin N. Grosof, Ian Horrocks, Raphael Volz, Stefan Decker: Description logic programs: combining logic programs with description logic. WWW 2003: 48-57
References