Knowledge Representation for the Semantic Web

Winter Quarter 2012

Slides 11 – 02/23/2012

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

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
Today: Reasoning with OWL
Contents

1. Rules and RIF
2. Rules expressible in OWL
3. Extending OWL with Rules: Nominal Schemas
4. References
• Horn Logic, often as Datalog (i.e. without function symbols) and with a modified but related semantics (Herbrand semantics).

• Prominent alternative to OWL modeling:
  – Rule-based expert systems
  – Prolog / Logic Programming
  – F-Logic [Kifer, Lausen, Wu, 1995]
  – W3C Rule Interchange Format RIF (standard since 2010)

  – Often argued to be “more intuitive” for non-logicians and domain experts.
Modeling with Rules

Orphan(harrypotter)
hasParent(harrypotter,jamespotter)
Orphan(x) \land hasParent(x,y) \rightarrow Dead(y)

\begin{align*}
\text{worksAt}(x, y) \land University(y) \land \text{supervises}(x, z) \land \text{PhDStudent}(z) \\
\quad \rightarrow \text{professorOf}(x, z)
\end{align*}

\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) \\
\quad \rightarrow \text{hasConflictingAssignedPaper}(v, x)
\end{align*}
Rules

Usually, of the (syntactic) form

\[ A_1 \land \ldots \land A_n \rightarrow B \]

body \rightarrow head

where \( A_i, B \) are atomic formulas.

Note:
- no disjunctive conclusions (head)
- no existential quantifiers in conclusions (head)
• Rules are usually considered to apply only to *known* constants.

• No possibility to “create” new things “on the fly” using $\exists$.

```
Human $\sqsubseteq \exists$hasParent.Human
```

• If rules are considered FOL formulas, then combining rules with ALC leads to undecidability.

[Reduction of some type of domino problem.]
Contents

1. Rules and RIF
2. Rules expressible in OWL
3. Extending OWL with Rules: Nominal Schemas
4. References
Rules in OWL

Which rules can be encoded in OWL?

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

\[ A \sqcap \exists R. \exists S. B \sqsubseteq C \text{ becomes } A(x) \land R(x, y) \land S(y, z) \land B(z) \rightarrow C(x) \]

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

Which rules can be encoded in OWL?

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

\[ \top \sqsubseteq \leq 1R.\top \text{ becomes } R(x, y) \land R(x, z) \rightarrow y = z \]

\[ A \sqcap \exists R.\{b\} \sqsubseteq C \text{ becomes } A(x) \land R(x, b) \rightarrow C(x) \]
Rules in OWL

Which rules can be encoded in OWL?

\[
\{a\} \equiv \{b\} \text{ becomes } a = b.
\]

\[
A \sqcap B \sqsubseteq \bot \text{ becomes } A(x) \land B(x) \rightarrow f.
\]

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

\[
A \sqcup B \rightarrow C \text{ becomes } A(x) \rightarrow C'(x) \text{ and } B(x) \rightarrow C'(x)
\]
A DL axiom $\alpha$ can be translated into rules if, after translating $\alpha$ into a first-order predicate logic expression $\alpha'$, and after normalizing this expression into a set of clauses $M$, each formula in $M$ is a Horn clause (i.e., a rule).

Issue: How complicated a translation is allowed?

Naïve translation: DLP

plus some more (since OWL 2 extends OWL 1)

e.g.,

$$R \circ S \sqsubseteq T \text{ becomes } R(x, y) \land S(y, z) \rightarrow T(x, z)$$

This essentially results in OWL 2 RL.
Rolification

Elephant(x) ∧ Mouse(y) → biggerThan(x, y)

- Rolification of a concept A: \( A \equiv \exists R_A \cdot \text{Self} \)

\[ \text{Elephant} \equiv \exists R_{\text{Elephant}} \cdot \text{Self} \]

\[ \text{Mouse} \equiv \exists R_{\text{Mouse}} \cdot \text{Self} \]

\[ R_{\text{Elephant}} \circ U \circ R_{\text{Mouse}} \sqsubseteq \text{biggerThan}. \]
Rolification

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

Woman(x) \land marriedTo(x, y) \land Man(y) \rightarrow hasHusband(x, y)

\[
R_{\text{Woman}} \circ marriedTo \circ R_{\text{Man}} \subseteq hasHusband
\]

careful – regularity of RBox needs to be retained:

\[
\text{hasHusband} \subseteq \text{marriedTo}
\]
worksAt(x, y) \land University(y) \land supervises(x, z) \land PhDStudent(z) 
\rightarrow \text{professorOf}(x, z)
Rules in OWL 2

- $\text{Man}(x) \land \text{hasBrother}(x,y) \land \text{hasChild}(y,z) \rightarrow \text{Uncle}(x)$
  - $\text{Man} \sqcap \exists \text{hasBrother}. \exists \text{hasChild}. \top \sqsubseteq \text{Uncle}$

- $\text{NutAllergic}(x) \land \text{NutProduct}(y) \rightarrow \text{dislikes}(x,y)$
  - $\text{NutAllergic} \equiv \exists \text{nutAllergic}. \text{Self}$
  - $\text{NutProduct} \equiv \exists \text{nutProduct}. \text{Self}$
  - $\text{nutAllergic} \circ \text{U} \circ \text{nutProduct} \sqsubseteq \text{dislikes}$

- $\text{dislikes}(x,z) \land \text{Dish}(y) \land \text{contains}(y,z) \rightarrow \text{dislikes}(x,y)$
  - $\text{Dish} \equiv \exists \text{dish}. \text{Self}$
  - $\text{dislikes} \circ \text{contains}^{-} \circ \text{dish} \sqsubseteq \text{dislikes}$
So how can we pinpoint this?

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

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

\[
- C \land \exists R\{a\} \land \exists S.(D \land \exists T\{a\}) \subseteq E
\]
So how can we pinpoint this?

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

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

\[ C \sqcap \exists R.\{a\} \sqsubseteq \exists R_1.\text{Self} \]
\[ D \sqcap \exists T.\{a\} \sqsubseteq \exists R_2.\text{Self} \]
\[ R_1 \circ S \circ R_2 \sqsubseteq V \]
Rule bodies as graphs

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

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

\[ C \cap \exists R.\{a\} \subseteq \exists R1.\text{Self} \]
\[ D \cap \exists T.\{a\} \subseteq \exists R2.\text{Self} \]
\[ R1 \circ S \circ R2 \subseteq P \]
Rule bodies as graphs

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

with \(y, z\) constants:

\[
R_{\exists \text{hasSubmittedPaper}.(\exists \text{hasAuthor}.\{y\} \cap \exists \text{atVenue}.\{z\})} \circ \text{hasReviewAssignment} \\
\circ R_{\exists \text{hasAuthor}.\{y\} \cap \exists \text{atVenue}.\{z\}} \\
\sqsubseteq \text{hasConflictingAssignedPaper}
\]
Formally

Given a rule with body \( B \), we construct a directed graph as follows:

1. Rename individuals (i.e., constants) such that each individual occurs only once – a body such as \( R(a,x) \land S(x,a) \) becomes \( R(a_1,x) \land S(x,a_2) \). Denote the resulting new body by \( B' \).

2. The vertices of the graph are then the variables and individuals occurring in \( B' \), and there is a directed edge between \( t \) and \( u \) if and only if there is an atom \( R(t,u) \) in \( B' \).

\[
C'(x) \land R(x,a) \land S(x,y) \land D(y) \land T(y,a) \rightarrow P(x,y)
\]

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

Definition 1. We call a rule with head $H$ tree-shaped (respectively, acyclic), if the following conditions hold.

- Each of the maximally connected components of the corresponding graph is in fact a tree (respectively, an acyclic graph)—or in other words, if it is a forest, i.e., a set of trees (respectively, a set of acyclic graphs).
- If $H$ consists of an atom $A(t)$ or $R(t, u)$, then $t$ is a root in the tree (respectively, in the acyclic graph).

\[ R(x, z) \land S(y, z) \rightarrow T(x, y) \] is acyclic but not tree-shaped

Theorem 1. The following hold.

- Every tree-shaped rule can be expressed in SROEL.
- Every acyclic rule can be expressed in SROEL.
1. Rules and RIF
2. Rules expressible in OWL
3. Extending OWL with Rules: Nominal Schemas
4. References
DL-safe variables

- A generalisation of DL-safety.
- DL-safe variables are special variables which bind only to named individuals (like in DL-safe rules).

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

\[ C(x) \land R(x, a) \land S(x, y) \land D(y) \land T(y, a) \rightarrow E(x) \]
can be translated into OWL 2.

duplicating nominals is ok
DL-safe variables

- A generalisation of DL-safety.
- DL-safe variables are special variables which bind only to named individuals (like in DL-safe rules).

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

\[ C(x) \land R(x, a) \land S(x, y) \land D(y) \land T(y, a) \rightarrow E(x) \]
can be translated into OWL 2.

- with, say, 100 individuals, we would obtain 100 new OWL axioms from the single rule above
DL-safety

• **DL-safe variables:** variables in rules which bind only to named individuals

• **Idea:**
  – start with rule not expressible in OWL 2
  – select some variables and declare them DL-safe such that resulting rule can be translated into several OWL 2 rules

• **DL-safe rule:** A rule with only DL-safe variables.

  It is known that “OWL 2 DL + DL-safe rules” is decidable.
  It is a *hybrid* formalism.
  E.g. OWL plus DL-safe SWRL.
Non-hybrid syntax: nominal schemas

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

assume \(y, z\) bind only to named individuals
we introduce a new construct, called

*nominal schemas*

or

*nominal variables*

\[R \exists \text{hasSubmittedPaper}.(\exists \text{hasAuthor}.\{y\} \sqcap \exists \text{atVenue}.\{z\}) \circ \text{hasReviewAssignment} \circ R \exists \text{hasAuthor}.\{y\} \sqcap \exists \text{atVenue}.\{z\} \sqsubseteq \text{hasConflictingAssignedPaper}\]
Nominal schema example 2

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

\[\exists \text{hasChild}.\{z\} \cap \exists \text{hasChild}.\exists \text{classmate}.\{z\} \subseteq C\]
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.
- We can actually also express all DL-safe (binary) Datalog rules!

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

\[
\exists U. (\{x\} \sqcap \exists R.\{y\}) \\
\sqcap \exists U. (\{y\} \sqcap A) \\
\sqcap \exists U. (\{z\} \sqcap \exists S.\{y\}) \\
\sqcap \exists U. (\{x\} \sqcap \exists T.\{z\}) \\
\sqsubseteq \exists U. (\{z\} \sqcap \exists P.\{x\})
\]
A tractable fragment

Definition 2. An occurrence of nominal schema \( \{x\} \) in a concept \( C \) is safe if \( C \) contains a sub-concept of the form \( \{v\} \sqcap \exists R.D \) for some nominal schema or nominal \( \{v\} \) such that \( \{x\} \) is the only nominal schema that occurs (possibly more than once) in \( D \). In this case, \( \{v\} \sqcap \exists R.D \) is a safe environment for this occurrence of \( \{x\} \), sometimes written as \( S(v, x) \).

Definition 3. Let \( n \geq 0 \) be an integer. A \( SROELV(\sqcap, \times) \) knowledge base \( KB \) is a \( SROELV_n(\sqcap, \times) \) knowledge base if in each of its axioms \( C \sqsubseteq D \), there are at most \( n \) nominal schemas appearing more than once in non-safe form, and all remaining nominal schemas appear only in \( C \).

\[ SROELV_n(\sqcap, \times) \] is tractable (Polytime)
covers OWL 2 EL
covers OWL 2 RL (DL-safe)
covers most of OWL 2 QL
Polytime smart transformation

\[ \exists \text{hasReviewAssignment}.((\{x\} \cap \exists \text{hasAuthor}.\{y\}) \cap (\{x\} \cap \exists \text{atVenue}.\{z\})) \]
\[ \cap \exists \text{hasSubmittedPaper}.(\exists \text{hasAuthor}.\{y\} \cap \exists \text{atVenue}.\{z\}) \]
\[ \subseteq \exists \text{hasConflictingAssignedPaper}.\{x\} \]

becomes \((a_i, a_j \text{ range over all named individuals})\)

\[ (\exists U.O_y) \cap (\exists U.O_z) \cap \exists \text{hasReviewAssignment}.(\{a_i\} \cap \{a_i\}) \]
\[ \cap \exists \text{hasSubmittedPaper}.(\exists \text{hasAuthor}.O_y \cap \exists \text{atVenue}.O_z) \]
\[ \subseteq \exists \text{hasConflictingAssignedPaper}.\{a_i\} \]

\[ \exists U.(\{a_i\} \cap \exists \text{hasAuthor}.\{a_j\}) \subseteq \exists U.(\{a_j\} \cap O_y) \]
\[ \exists U.(\{a_i\} \cap \exists \text{atVenue}.\{a_j\}) \subseteq \exists U.(\{a_j\} \cap O_z) \]
OWL syntax for nominal schemas

Functional Syntax:

Add the last line, \((\text{ObjectVariable})\), to the \text{ClassExpression} production rule:

\[
\text{ClassExpression} := \\
\text{Class} | \\
\text{ObjectIntersectionOf} | \text{ObjectUnionOf} \text{ObjectComplementOf} | \text{ObjectOneOf} | \\
\text{ObjectSomeValuesFrom} | \text{ObjectAllValuesFrom} | \text{ObjectHasValue} | \text{ObjectHasSelf} | \\
\text{ObjectMinCardinality} | \text{ObjectMaxCardinality} | \text{ObjectExactCardinality} | \\
\text{DataSomeValuesFrom} | \text{DataAllValuesFrom} | \text{DataHasValue} | \\
\text{DataMinCardinality} | \text{DataMaxCardinality} | \text{DataExactCardinality} | \\
\text{ObjectVariable}
\]

Add the next production rule to the grammar:

\[
\text{ObjectVariable} := \text{’ObjectVariable (’ quotedString ’ ^ ^ xsd:string’)}
\]
### Translation to Turtle:

<table>
<thead>
<tr>
<th>Functional-Style Syntax</th>
<th>S Triples Generated in an Invocation of $T(S)$</th>
<th>Main Node of $T(S)$</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ObjectVariable(”v1” ^^ xsd:string)</code></td>
<td><code>_:x rdf:type owl:ObjectVariable</code></td>
<td><code>_:x</code></td>
</tr>
<tr>
<td></td>
<td><code>_:x owl:variableId ”v1”^^xsd:string</code></td>
<td></td>
</tr>
</tbody>
</table>
### 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

<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>

from the TONES repository:
Naïve implementation – experiments

Optimization through smart grounding (all ns occurring safely)

<table>
<thead>
<tr>
<th></th>
<th>0 ns</th>
<th>1 ns</th>
<th>2 ns</th>
<th>3 ns</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rex (100)</strong></td>
<td>0.025</td>
<td>0.013</td>
<td>1.689</td>
<td>OOM</td>
</tr>
<tr>
<td>Rex Optimized</td>
<td>0.058</td>
<td>0.023</td>
<td>0.046</td>
<td>0.053</td>
</tr>
<tr>
<td><strong>Spatial (100)</strong></td>
<td>0.035</td>
<td>0.021</td>
<td>1.536</td>
<td>OOM</td>
</tr>
<tr>
<td>Spatial Optimized</td>
<td>0.018</td>
<td>0.013</td>
<td>0.033</td>
<td>0.044</td>
</tr>
<tr>
<td><strong>Xenopus (100)</strong></td>
<td>0.063</td>
<td>0.07</td>
<td>1.598</td>
<td>OOM</td>
</tr>
<tr>
<td>Xenopus Optimized</td>
<td>0.099</td>
<td>0.037</td>
<td>0.083</td>
<td>0.097</td>
</tr>
</tbody>
</table>

<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>Rex</td>
<td>552</td>
<td>0</td>
<td>6</td>
<td>100</td>
</tr>
<tr>
<td>Spatial</td>
<td>106</td>
<td>0</td>
<td>13</td>
<td>100</td>
</tr>
<tr>
<td>Xenopus</td>
<td>710</td>
<td>0</td>
<td>5</td>
<td>100</td>
</tr>
</tbody>
</table>
### Naïve implementation – experiments

Note: with 2 different ns we cover all of OWL 2 RL (but functionality)

<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>0.00”</td>
<td>0.01”</td>
<td>0.00”</td>
</tr>
<tr>
<td>Swe (22)</td>
<td>3.58”</td>
<td>0.08”</td>
<td>3.73”</td>
<td>0.07”</td>
</tr>
<tr>
<td>Bui (42)</td>
<td>2.7”</td>
<td>0.16”</td>
<td>2.5”</td>
<td>0.15”</td>
</tr>
<tr>
<td>Wor (80)</td>
<td>0.11”</td>
<td>0.04”</td>
<td>0.12”</td>
<td>0.05”</td>
</tr>
<tr>
<td>Tra (183)</td>
<td>0.05”</td>
<td>0.03”</td>
<td>0.05”</td>
<td>0.02”</td>
</tr>
<tr>
<td>FTr (368)</td>
<td>0.03”</td>
<td>4.28”</td>
<td>0.05</td>
<td>5.32”</td>
</tr>
<tr>
<td>Eco (482)</td>
<td>0.04”</td>
<td>0.24”</td>
<td>0.07”</td>
<td>0.02”</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>No ns</th>
<th>1 ns</th>
<th>2 ns</th>
<th>3 ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rex (100)</td>
<td>0.025</td>
<td>0.009</td>
<td>0.031</td>
<td>0.013</td>
</tr>
<tr>
<td>Rex Optimized (100)</td>
<td>0.058</td>
<td>0.023</td>
<td>0.046</td>
<td>0.011</td>
</tr>
<tr>
<td>Spatial (100)</td>
<td>0.035</td>
<td>0.029</td>
<td>0.021</td>
<td>0.014</td>
</tr>
<tr>
<td>Spatial Optimized (100)</td>
<td>0.018</td>
<td>0.013</td>
<td>0.033</td>
<td>0.007</td>
</tr>
<tr>
<td>Xenopus (100)</td>
<td>0.063</td>
<td>0.018</td>
<td>0.07</td>
<td>0.19</td>
</tr>
<tr>
<td>Xenopus Optimized (100)</td>
<td>0.099</td>
<td>0.037</td>
<td>0.083</td>
<td>0.018</td>
</tr>
</tbody>
</table>
Remember the partonomies problem?

In the partonomies lecture, we had several issues with modeling the part-of ontology following Winston.

E.g., relations cannot be transitive, asymmetric, and irreflexive at the same time.

We can now approximate this as follows:

Characterize the relation (e.g., R) as transitive and asymmetric.
Furthermore, specify \( \{x\} \cap \exists R.\{x\} \subseteq \bot. \)

More generally, if you run into a rule which you cannot model in OWL, simply approximate using nominal schemas.
Contents

1. Reasoning Needs
2. Rules expressible in OWL
3. Extending OWL with Rules: Nominal Schemas
4. References
This part of the lecture is very close to:


Rules in OWL:


Nominal Schemas:


References

Nominal Schemas:


• Cong Wang, Pascal Hitzler, A Tractable Resolution Procedure for $\text{SROEL} \forall_n(\text{us},x)$. Technical Report, 2012.