Upper Layers in the Semantic Web

Sebastian Wandelt

Based on slides from Grigoris Antoniou, and Frank van Harmelen
Logic layer

• Main purpose: additional inferences over the actual ontology
• Few rule languages already exist
  ♦ We will talk about some today …
Logic: Rules

- Two possible approaches
  - Use rule-based languages **instead** of OWL
  - Use rule-based languages **and** OWL together
RDF/S and OWL (Lite and DL) are specializations of predicate logic
- correspond roughly to a description logic
- They define reasonable subsets of PL
- Trade-off between the expressive power and the computational complexity:
  - The more expressive the language, the less efficient the corresponding proof systems
Specializations of PL: Horn Logic

- A rule has the form: $A_1, \ldots, A_n \rightarrow B$
  - $A_i$ and $B$ are atomic formulas
- There are 2 ways of reading such a rule:
  - **Deductive rules**: If $A_1, \ldots, A_n$ are known to be true, then $B$ is also true
  - **Reactive rules**: If the conditions $A_1, \ldots, A_n$ are true, then carry out the action $B$
FOL: (All except (6)), (2)+(3)+(4): DLs
(4): Description Logic Programs (DLP), (3): Classical Negation
(4)+(5): Horn Logic Programs, (4)+(5)+(6): LP
(6): Non-monotonic features (like NAF, etc.)
Rules

B1, . . . , Bn → A

• A, B1, ... , Bn are atomic formulas
• A is the head of the rule
• B1, ... , Bn are the premises (body of the rule)
• The commas in the rule body are read conjunctively
• Variables may occur in A, B1, ... , Bn
  ♦ loyalCustomer(X), age(X) > 60 → discount(X)
  ♦ Implicitly universally quantified
Description Logic Programs

- Description Logic Programs (DLP) can be considered as the intersection of Horn logic and description logic.

- DLP allows to combine advantages of both approaches. For example:
  - A modeler may take a DL view, but
  - the implementation may be based on rule technology.
RDF and RDF Schema

- A triple of the form \((a,P,b)\) in RDF can be expressed as a fact \(P(a,b)\)
- An instance declaration of the form \(\text{type}(a,C)\) (stating \(a\) is instance of class \(C\)) can be expressed as \(C(a)\)
- The fact that \(C\) is a subclass (or subproperty) of \(D\) can be expressed as \(C(X) \rightarrow D(X)\)
Model the following with rules:

- `sameClassAs(C,D)`
- `samePropertyAs`
- Transitivity of a property `P`
- The intersection of `C_1` and `C_2` is a subclass of `D`
- `C` is subclass of the intersection of `D_1` and `D_2`
- The union of `C_1` and `C_2` is a subclass of `D`
- `C subclassOf allValuesFrom(P,D)`
- `someValuesFrom(P,D) subclassOf C`
• **sameClassAs**(C,D) (or samePropertyAs) can be expressed by the pair of rules
  🟦 C(X) → D(X)
  🟦 D(X) → C(X)

• Transitivity of a property P can be expressed as
  🟦 P(X,Y), P(Y,Z) → P(X,Z)
The intersection of $C_1$ and $C_2$ is a subclass of $D$ can be expressed as

\[ C_1(X), C_2(X) \rightarrow D(X) \]

$C$ is subclass of the intersection of $D_1$ and $D_2$ can be expressed as

\[ C(X) \rightarrow D_1(X) \]

\[ C(X) \rightarrow D_2(X) \]
• The union of $C_1$ and $C_2$ is a subclass of $D$ can be expressed by the pair of rules
  - $C_1(X) \rightarrow D(X)$
  - $C_2(X) \rightarrow D(X)$

• The opposite direction cannot be expressed in Horn logic
Restrictions in OWL

- \( C \) subClassOf allValuesFrom\((P,D)\) can be expressed as
  \[ C(X), P(X,Y) \rightarrow D(Y) \]
  Where \( P \) is a property, \( D \) is a class and allValuesFrom\((P,D)\) denote the anonymous class of all \( x \) such that \( y \) must be an instance of \( D \) whether \( P(x,y) \)

- The opposite direction cannot in general be expressed
• **someValuesFrom**(P,D) subClassOf C can be expressed as
  - P(X,Y), D(Y) → C(X)
  - Where P is a property, D is a class and someValuesFrom(P,D) denote the anonymous class of all x for which there exists at least one y instance of D, such that P(x,y)
  - The opposite direction cannot in general be expressed
Restrictions in OWL (3)

- Cardinality constraints and complement of classes cannot be expressed in Horn logic in the general case.
Semantic Web Rules Language

- Extend OWL axioms to include Horn-like clauses.
- Maximum compatibility with OWL
- Built on top of OWL (same semantics)
Conceptualization of the domain

High Expressiveness

Rules Layer

Ontology Layer

OWL-DL

SWRL
A rule in SWRL has the form

- $B_1, \ldots, B_n \rightarrow A_1, \ldots, A_m$
- Commas denote conjunction on both sides
- $A_1, \ldots, A_m, B_1, \ldots, B_n$ can be of the form $C(x)$, $P(x,y)$, $\text{sameAs}(x,y)$, or $\text{differentFrom}(x,y)$ where $C$ is an OWL description, $P$ is an OWL property, and $x, y$ are Datalog variables, OWL individuals, or OWL data values
Is a rule of the form

\[ B_1, \ldots, B_n \rightarrow A_1, \ldots, A_m \]

more expressive than rules of the form

\begin{itemize}
  \item \( B_1, \ldots, B_n \rightarrow A \)
  \item \( B_1, \ldots, B_n \rightarrow A \)
\end{itemize}
SWRL Properties

- If the head of a rule has more than one atom, the rule can be transformed to an equivalent set of rules with one atom in the head.
- Expressions, such as restrictions, can appear in the head or body of a rule.
- This feature adds significant expressive power to OWL, but at the high price of undecidability.
Where has all the decidability gone?

• Problem: existential quantifiers + recursive rules…

For instance (in DL syntax):

\[ C_2(X) \rightarrow C_1(X). \]
\[ C_1 \sqsubseteq \exists R.C_2 \]
Where has all the decidability gone?

- Problem: existential quantifiers + recursive rules…

For instance (in DL syntax):

\[ C_2(X) \rightarrow C_1(X). \]
\[ C_1 \sqsubseteq \exists R.C_2^* \]

\[ C(a) \]

by skolemization creates \( C(f(a)) \)
\[ C(f(f(a))) \]
\[ \ldots \text{ etc.} \]

* Corresponds to a rule: \((\exists Y.R(X,Y)) \rightarrow C_2(X)\) which amounts to \( \forall X.\exists Y.R(X,Y) \lor \neg C_2(X) \)
Where has all the decidability gone? Cont’d

• The simple example from the last slide might still be solved with a blocking strategy in the evaluation algorithm, but…

• … in general, interaction of existential quantifiers and recursive rules causes troubles, by introducing function symbols
DLP vs. SWRL

- DLP tries to combine the advantages of both languages (description logic and function-free rules) in their common sublanguage
- SWRL takes a more maximalist approach and unites their respective expressivities
- The challenge is to identify sublanguages of SWRL that find the right balance between expressive power and computational tractability
Horn logic is a subset of predicate logic that allows efficient reasoning, orthogonal to description logics.

Horn logic is the basis of monotonic rules.

DLP and SWRL are two important ways of combining OWL with Horn rules.

DLP is essentially the intersection of OWL and Horn logic, whereas SWRL is a much richer language.