The upper layers of the Semantic Web

Riccardo Rosati

Corso di Laurea Magistrale in Ingegneria Informatica
Sapienza Università di Roma
2016/2017
The upper layers of the Semantic Web Tower include:

- **Trust**
- **Proof**
- **Logic**
- **Ontology vocabulary**
- **RDF + rdfschema**
- **XML + NS + xmschema**
- **Unicode**
- **URI**

The lower layers focus on foundational technologies like Unicode and URI.
The Logic/Proof/Trust layers

SW Stack upper layers:
- the Logic layer enables the writing of rules
- the Proof layer executes the rules
- the Trust layer decides whether to trust the given proof or not

technology for these layer at a very early stage:
- few standards exist
- open architectural issues
The notion of rule

- rule = “if-then” statement
- a rule can be static or dynamic
  - static rule (implication): if condition C1 is true then conclude that also condition C2 is true
  - dynamic rule: if event E occurs and condition C holds then execute action A
- semantics of rules:
  - procedural (operational)
  - declarative
Rule bases as knowledge bases

- static rules may be considered as statements expressing knowledge
- rule base = knowledge base
- interpretation of a rule similar (but not equal) to the boolean implication operator
- “constructive” (one-way) implication (contrapositive does not hold)
- more generally, semantics of rules based on (various notions of) closed-world assumption
Expressive limitations of DLs and OWL

- the typical expressiveness of Description Logics does not allow for addressing the following aspects:
  - defining **predicates of arbitrary arity** (not just unary and binary) using **variable quantification** beyond the tree-like structure of DL concepts (many DLs are subsets of the two-variable fragment of FOL)
  - formulating **expressive queries** over DL knowledge bases (beyond concept subsumption and instance checking)
  - formalizing various **forms of closed-world reasoning** over DL KBs (DLs and OWL have an open-world semantics)
  - more generally, expressing forms of **nonmonotonic knowledge**, like default rules
Default mechanisms

- example: suppose the OWL ontology models a domain of people
- we would like to model a “default rule” that states that, in the absence of a specific assertion (stating that the person is blind) a person is not blind
- the above kind of information cannot be expressed in an OWL TBox
  - the only way to express it in OWL is to add one ABox assertion (which states that the person is not blind) for every person in the domain
- semantics based on closed-world assumption are needed to express default mechanisms
Rule formalisms

- static rules:
  - logic programming languages:
    - Prolog
    - answer set programs
    - nonmonotonic Datalog
- dynamic rules:
  - ECA rules
  - production rules
  - ...
Logic programming: Prolog

Prolog rule: statement of the form

- \( a : - b_1, b_2, \ldots, b_n \)

- intuitive reading:
  “if \( b_1 \) and \( b_2 \) and ... and \( b_n \), then \( a \)”

- \( a = \) rule head

- \( b_1, b_2, \ldots, b_n = \) rule body

- \( a \) and all \( b_i \)’s are first-order atoms

- some \( b_i \) may be negated
Logic programming

examples:

• uncle(x,y) :- father(x,z), brother(z,y).
• grandparent(x,z) :- parent(x,y), parent(y,z).

recursive rules:

• ancestor(x,y) :- parent(x,y).
• ancestor(x,y) :- parent(x,z), ancestor(z,y).

use of negation:

• innocent(x) :- person(x), not guilty(x),
Rules as an alternative ontology language

geneneral idea: use rules as an ontology language

- first proposal: use rule-based languages instead of OWL
  - change of the Semantic Web Stack
- second proposal: use rule-based languages AND OWL as ontology languages
  - different change of the Stack (two-stack)
  - rules are not on top of OWL anymore, they are besides OWL
One-stack vs. two-stack architecture

The upper layers of the SW
RIF

- RIF (Rule Interchange Format) = W3C Working Group
- aim: providing standards for rules interchange
- recommendation (June 2010) of six new standard formats
RIF

Six new standards (June 2010):

• **RIF Core Dialect**, which provides a standard, base level of functionality for interchange

• **RIF Basic Logic Dialect** and **RIF Production Rule Dialect** provide extended functionality matching two common classes of rule engines

• **RIF Framework for Logic Dialects** describes how to extend RIF for use with a large class of systems

• **RIF Datatypes and Built-Ins 1.0** borrows heavily from XQuery and XPath for a set of basic operations

• **RIF RDF and OWL Compatibility** specifies how RIF works with RDF data and OWL ontologies

See [http://www.w3.org/TR/rif-overview/](http://www.w3.org/TR/rif-overview/)
The Semantic Web cake revisited
Proof layer

- main purpose: to provide explanations about the answers given by automated agents that consume the provided information
  - = provenance problem
- very few results so far:
  - general OWL-DL ontologies
  - fragments of OWL-DL
  - some rule languages
Trust layer

- Trust layer: research at an early stage
- strictly depends on the choices concerning the lower layers
- some preliminary results:
  - provenance/pinpointing in Description Logic ontologies: finding the explanation for an answer
  - techniques for authorization
  - quality of the answers / ranking (top-k answers)