FIRE – EXTENDING RACER WITH AN ENGINE FOR SWRL-LIKE RULES

Kruthi Bhoopalam and Volker Haarslev
Concordia University, Montreal, Quebec, Canada

Introduction
Fire is a DL-based rule engine for OWL ontologies extended with SWRL-like rules.
The Fire system:
• is implemented in Java.
• intends to extend RACER’s DL reasoning for rules.
• supports a SWRL-like rule language, treating rules as true rules and not as implications.
The current implementation uses:
• oRQL (new Racer Query Language) queries and services of RACER externally using Racer Java API.
• Protégé SWRL Factory Java API.

The SWRL-like Rule Language
Syntax
The SWRL-like rule syntax follows the SWRL rule syntax.1

Further restrictions on the syntax
These restrictions are further imposed on the syntax for ease of implementation:
• Rules must be atomic (single atom in rule consequent)
• Rules must be ‘strictly Horn’ (premises of rule atoms may only be
  some of property or class)
• SWRL built-ins, sameAs, differentFrom and OWL
  DatatypeProperty predicates are not allowed in rule atoms
• Negated concept/rule atoms are not allowed in rules

Semantics
• Rules are true rules (i.e., if body holds then head must hold) and not
  implications.
• The instantiations of the rule are considered only if they are consistent
  with all the triggered rules.
• Rules are not allowed to create new individuals in the KB. The
  reasoning support follows the active domain semantics, i.e., the
  variables in rules are bound only to explicitly named individuals in
  the knowledge base (KB) and not to the arbitrary individuals
  encountered during the proof.

The SWRL-like rule syntax follows the SWRL rule syntax.1

Further restrictions on the syntax
These restrictions are further imposed on the syntax for ease of implementation:
• Rules must be atomic (single atom in rule consequent)
• Rules must be ‘strictly Horn’ (premises of rule atoms may only be
  some of property or class)
• SWRL built-ins, sameAs, differentFrom and OWL
  DatatypeProperty predicates are not allowed in rule atoms
• Negated concept/rule atoms are not allowed in rules

SEMANTICS
• Rules are true rules (i.e., if body holds then head must hold) and not
  implications.
• The instantiations of the rule are considered only if they are consistent
  with all the triggered rules.
• Rules are not allowed to create new individuals in the KB. The
  reasoning support follows the active domain semantics, i.e., the
  variables in rules are bound only to explicitly named individuals in
  the knowledge base (KB) and not to the arbitrary individuals
  encountered during the proof.

Results
The results of the KB using a DL reasoner are compared with the results obtained on
implementing the native pattern matching algorithm.

Future Directions
• Extending support for OWL DatatypeProperty, sameAs and differentFrom predicates in rule
  atoms.
• Optimize construction of RETE pattern network by exploiting common predicates among
  atoms of different rules.
• Use RACER’s publishSubscribe mechanism to efficiently learn about changes made to any
  concept.
• Extending support for rules with SWRL semantics.
• Providing Fire as a plug-in for Protégé’s SWRL Tab.
• Integration of the RETE algorithm implementation into RACER for reasoning with aRQL rules.

Open Issues
• To support reasoning for SWRL rules, it is interesting to explore an approach similar to the
  CARIN algorithm.
• Derive all possible completions of the KB using a DL reasoner.
• Fire rules for every clash-free completion derived.
• A non-contradicting result of any clash-free completion with fired rules determines an answer.
• Expressiveness supported – what fragment of SWRL can be practically supported? Still unclear.