

Fire – Extending Racer by an Engine for SWRL Rules

Kruthi Bhoopalam and Volker Haarslev
Concordia University, Montreal, QC, Canada
{k_bhoopa|haarslev}@cse.concordia.ca

In this paper we propose the Fire system, a prototype rule engine for reasoning with SWRL (Semantic Web Rules Language) [5] rules and OWL (Ontology Web Language) ontologies. This system is intended as an extension to the reasoning services of the RACER system [3]. For ease of implementation, support is currently provided for SWRL rules that are restricted as follows: (i) Rules must be atomic (single atom in the rule consequent); (ii) Rules must have tree-shaped antecedents; (iii) Rules must be ‘strictly Horn’ (predicates of rule atoms may only be ‘name’ of property or class); (iv) SWRL built-ins, sameAs, differentFrom or OWL DatatypeProperty may not be used for predicates in rule atoms. The Fire system is implemented in Java in the following way; Rules defined in the ontology are extracted using Protégé’s SWRL API [6]. Facts asserted in and entailed by the ontology are obtained from RACER. The RETE algorithm [1, 2] is implemented for ‘pattern matching’ these facts with patterns in rule antecedents. Materialization of rules for which instantiations/bindings are provided by the pattern matcher, is done by synchronizing with RACER’s inferences. Simple ‘random selection’ strategy is applied in resolving the conflict when more than one ‘new’ rule instantiation is provided at the same time. The reasoning process terminates when no ‘new’ inference is found compared to the already ‘known’ facts (asserted, entailed or new inferences from rules materialized earlier). Reasoning is offered in two modes, based on the number of rule instantiations considered during materialization for a given state of the knowledge base; (i) One-at-a-time mode: ‘one’ rule instantiation for an applicable rule at a time, and (ii) Batch mode: ‘all’ rule instantiations for all applicable rules at once in a batch. Our initial results of Fire’s performance over example SWRL ontologies have been very encouraging. We are currently working on several open issues to optimize Fire’s performance and extend its services: (i) Extend support for use of OWLDatatypeProperty, sameAs and differentFrom predicates in rules; (ii) Optimize construction of the RETE pattern network by exploiting the common predicates generally existing among different rules; (iii) Use the publish-subscribe mechanism from RACER to efficiently learn about the changes (newly asserted/inferred instantiations) made to any concept in the ontology that is relevant for rule reasoning; (iv) Provide Fire as a plug-in support for Protégé’s SWRL Tab; (v) Integrate an optimized version of Fire as a rule engine for nRQL (new Racer Query Language) rules [4] into the RACER system.

References

- [1] C. L. Forgy. Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence*, 19:17–37, 1982.
- [2] J. Giarratano and G. Riley. *Expert Systems: Principles and Programming*. Boston:PWS, 1998.
- [3] Volker Haarslev and Ralf Möller. Racer system description. In *Proceedings of International Joint Conference on Automated Reasoning (IJCAR 2001)*, Siena, Italy, June 2001.
- [4] Volker Haarslev, Ralf Möller, and Michael Wessel. Querying the semantic web with Racer + nRQL. In *In Proc. of the KI-2004 International Workshop on Applications of Description Logics (ADL’04)*, Ulm, Germany, 2004.
- [5] Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, and et al. SWRL: a semantic web rule language combining OWL and RuleML, May 2004.
- [6] Holger Knublauch, A. Mark, Musen, and Alan L. Rector. Editing description logic ontologies with the Protégé OWL plugin. In *International Workshop on Description Logics*, Whistler, BC Canada, 2004.