RIFEnge: A Rule Engine for RIF Language

Ismail Akbari, Yevgen Biletskiy, Weichang Du
Faculty of Computer Science, University of New Brunswick

UNB EST. 1785 UNIVERSITY OF NEW BRUNSWICK
Computer Science Fredericton

Abstract

Nowadays with a growing interest in Semantic Web technologies, rule languages and rule engines are receiving an increasing attention. In literature, there are several rule languages aiming at rule interchange, transform and building a general rule markup language for Semantic Web including RIF, RuleML and SWRL. This paper describes a rule inference engine called RIFEngne developed for RIF reasoning on the Semantic Web. The RIFEngne is a top-down and bottom-up reasoning engine. It can answer queries as well as infer new knowledge from available rules.

Keywords

Semantic Web, Rule Engine, Rule Markup Languages, RIF, RuleML

Web Rule Languages

The syntax (i.e. facts and rules) of every rule language consists of alphabet, terms and formulae which themselves consist of constants, variables, built-ins, functions, connective symbols, quantifiers, and other auxiliary symbols (see figure 1) [1].

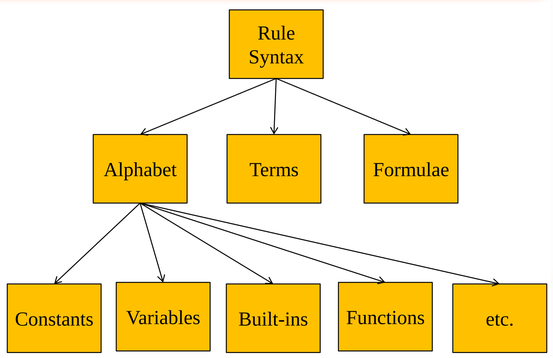

Figure 1 . Structure of a rule language

Generally, a rule (in a rule markup language) consists of consequence and antecedent containing clauses combined through logical operators. If all conditions (premises) in antecedent turned out true, then the consequence part of the rule will result in true (and actions may fire).
Table 1 shows some of the features of Web rule languages that can be used to differentiate rule markup languages from each other.

Table 1. Common features of Web rule languages

Feature	Example (in a hypothetical language)
Variables	?x
Facts	Father(“John”)
Functions	SameAs(“John Smith”, “J. Smith”)
Built-in functions	Subtract(?x, 2)
Logical quantifiers	ForAll, Exists
Logical operators	AND, OR, NOT
Data types	String, Integer
Slotted/Positional	FullName(First -> “John” Last->”Smith”) FullName (“John”, “Smith”)
Rule set and rule name	Group (rule set 1)
Webized	Name<http://www.example.com/people#names>(“John Smith”)
Production operations	Call_Customer (?customer)
Monotonicity	Monotonicity indicates that learning a new piece of knowledge cannot reduce the set of what is already available.
Rule interchange format	XML, JSON

RIFEnge Rule Engine

Knowing details of a rule syntax and semantics are an essential part a start phase to develop a reasoning engine for that language. RIFEnge system developed for reasoning over RIF-BLD [2] rule bases includes a rule parser, visualizer, and an inference engine. RIFEnge has a RIF translator (RIF presentation syntax to its XML syntax) as well. Parsing rules is a prerequisite for rule reasoning.

RIFEngne parser is developed with the help of ANTLR 4. Figure 2 shows the RIFEnge components.

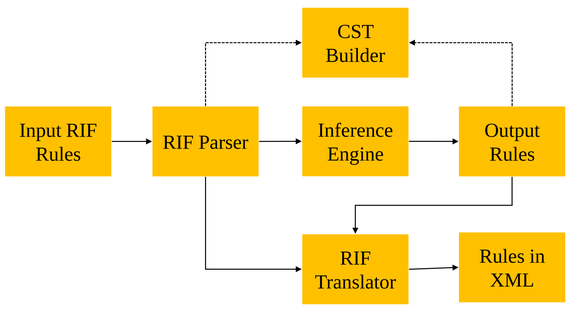

Figure 2. RIFEnge rule engine

ANTLR

ANother Tool for Language Recognition (ANTLR) [3] is a language recognizer and parser generator. Language implementers can use ANTLR to develop their domain specific language (DSL). ANTLR is a parser generator that uses LL(*) for parsing. An LL parser is called an LL(k) parser if it uses k tokens of look ahead when parsing a sentence. The following shows the ANTLR notation for defining a production rule (parser or lexer).

```
rule_name : definition ;
```

The following shows as an example definition of two production rules of RIF-BLD grammar in ANTLR notation:

```
rule  : (irimeta? 'Forall' Var+ '(' clause ')') | clause    ;
fragment UNSIGNEDINT: [0-9]+ ;
```

RIF Parser

RIF-BLD parser is developed using the ANTLR library. After parsing the input rules by the parser, they are fed to the reasoning component of RIFEnge rule engine. Consider the example in table 2. Figure 2 shows the CST tree output of the parser after parsing the rules in table 2.

Table 2. A RIF-BLD rule as input to the RIFEnge parser

```
Document (
  Group (
      Forall ?Buyer ?Item ?Seller (
        cpt:buy(?Buyer ?Item ?Seller) :- cpt:sell(?Seller ?Item  ?Buyer)  )
  )
)
```

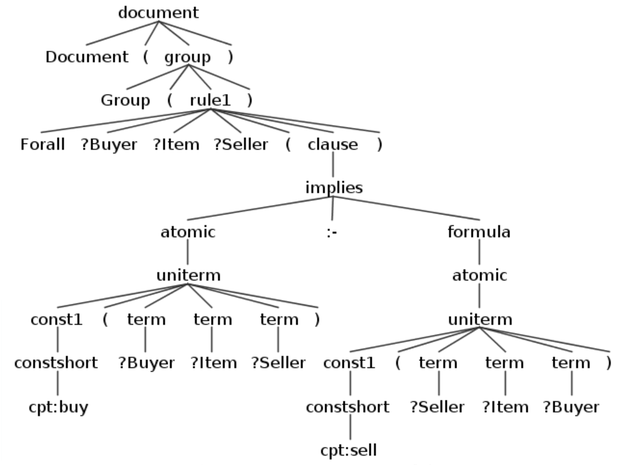

Figure 3. CST tree output of the parser

Inference Engine

The proposed RIF rule engine (RIFEnge) has both forward and backward reasoning capabilities. It can answer queries as well as infer new knowledge from available rules. Inferencing new knowledge in based on universal and existential formulae in the rule base. Figure 4 shows RIFEnge user interface. It also shows an example of running RIFEnge inference engine on a RIF rule base. Figure 5 shows the original and result (inferred) rules (from figure 4) in a tree view.

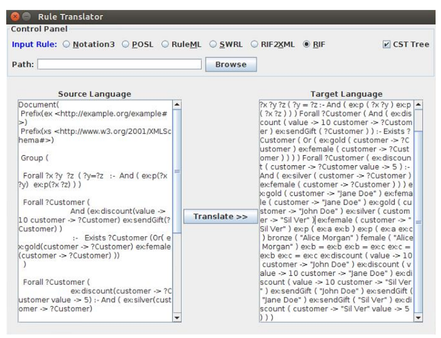

Figure 4. RIFEnge GUI

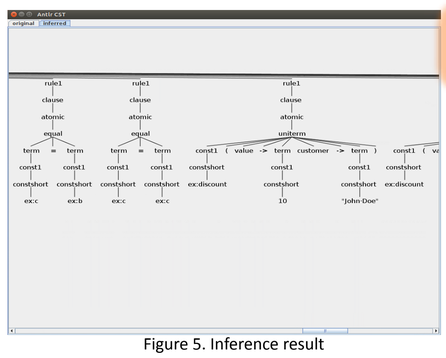

Figure 5. Inference result

References

[1] Boley, Harold, Said Tabet, and Gerd Wagner. "Design rationale of RuleML: A markup language for semantic web rules." Proceedings of the First International Conference on Semantic Web Working. CEUR-WS. org, 2001.
[2] Boley, Harold, and Michael Kifer. "RIF basic logic dialect." W3C Working Draft (July 2009) (2007).
[3] Parr, Terence. The definitive ANTLR 4 reference. Pragmatic Bookshelf, 2013.