Faculty of Computer Science

CS6795 Semantic Web Techniques

Project Proposal

"A Test Case Suite for Hornlog+ RuleML 1.01"

Professor: Dr. Harold Boley

Advisor: Dr. Tara Arthan

Team: Zhenzhi Cui

Radhika Yadav

Introduction:

RuleML is knowledge representation language which is contrived for the interchange of the major kinds of Web rules in an XML format that is uniform across various rule logics and platforms. It is elucidated as an extensible family of sublanguages and has eclectic coverage. Its modular methodology of schemas allows rule interchange with high accuracy. Deliberation RuleML incorporates a family of sublanguages, commencing with Datalog, Hornlog, and other Derivation Rules, and advancing to FOL and beyond. The outstanding part of Deliberation RuleML 1.01 is the potential to amalgamate one or more of the following Datalog extensions which jointly elucidate Datalog+:

- **Existential Rule:** In this rule, the "then" part of a rule has existentially quantified variables, as needed for description logics, F- logic and PSOA RuleML, Rule-Based Data Access, etc.
- **Equality Rule:** In this rule, the "then" part of a rule is the "Equal" predicate, as needed for user-defined/'semantic' equality in logics with equality and functional logic programming and this was already allowed in RuleML 1.0.
- **Integrity Rule:** In this rule, the "then" part of a rule is falsity, as an appropriate way to express negative integrity constraints.[1]

The development of Datalog+ test and use cases have been done for several objectives, which includes validation (specification), transformation (interoperability), execution (benchmarking), and prototyping (application).
Hornlog+ is the part of the Deliberation RuleML 1.01 family which has not been explored much. Hornlog+ enlarges Datalog+ with constructor or functions, which is similar to the role Hornlog plays for Datalog. In particular, we can customize the Hornlog+ RuleML 1.01 language features via MYNG 1.01.[2]

Objective:

In our project, we will construct the Hornlog+ test rulebases and test queries. There are three approaches to the construction of Hornlog+ test rulebases that we will combine in our project:

A1. By augmenting the current examples of Datalog+ rulebases with functions.

A2. By augmenting the current Hornlog examples with the three defining extensions.

A3. By creating new Hornlog+ examples (from scratch).

All of these Hornlog+ RuleML 1.01 instance documents will be formatted in an easy-to-read standard (pretty-print-indented) manner. Like the Datalog+ RuleML 1.01 examples, we will modify and validate them in Relax NG step by step. We will make the documented Hornlog+ test rulebases and the results of their validations available (including with a few screenshots).

Project Outline:

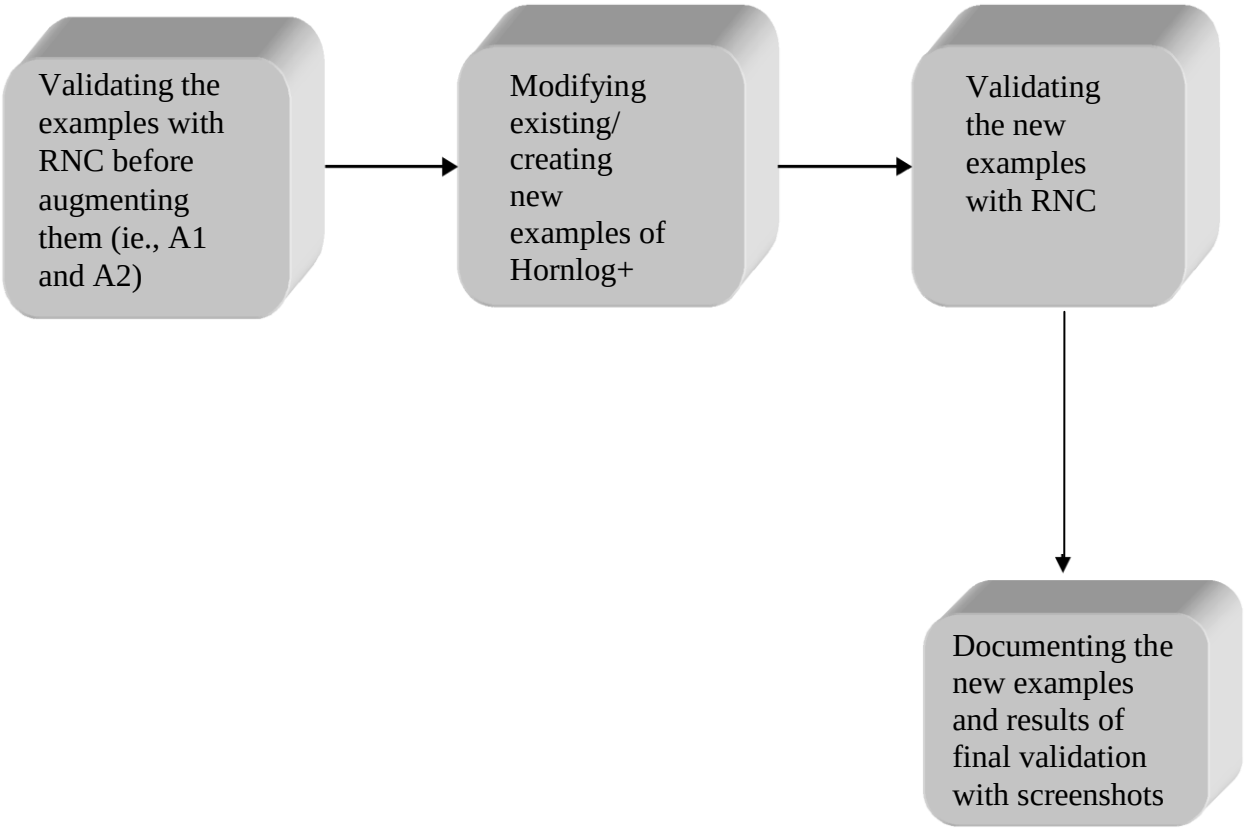

Fig. 1. Project Work Flow

Examples for the Test Case Suite:

- **Developing Datalog+ example into Hornlog+ example**

Datalog+:

```
<Assert>
   <Implies>
      <if>
         <Equal>
            <Ind>Sue</Ind>
            <Ind>Maria</Ind>
         </Equal>
      </if>
      <then>
         <Or />
      </then>
   </Implies>
</Assert>
```

Hornlog+:

```
<Assert>
   <Implies>
      <if>
         <Equal>
            <Ind>Sue</Ind>
            <Expr>
               <Fun>motherOf</Fun>
               <Ind>Sue</Ind>
            </Expr>
         </Equal>
      </if>
      <then>
         <Or />
      </then>
   </Implies>
</Assert>
```

- **Developing Hornlog example into Hornlog+ example**

Hornlog:

```
<Assert mapClosure="universal">
   <Implies>
      <if>
         <Atom>
            <op>
               <Rel>book</Rel>
            </op>
            <Expr>
               <Fun per="copy">parts</Fun>
               <Var>title</Var>
               <Var>author</Var>
               <Var>toc</Var>
               <Var>chapters</Var>
            </Expr>
         </Atom>
```

```
      </if>
      <then>
        <Atom>
          <Rel>author</Rel>
          <Ind>Bin</Ind>
        </Atom>
      </then>
    </Implies>
</Assert>
```

Hornlog+:

```
<Assert mapClosure="universal">
      <Implies>
        <if>
          <Atom>
            <op>
              <Rel>book</Rel>
            </op>
            <Expr>
              <Fun per="copy">parts</Fun>
              <Var>title</Var>
              <Var>author</Var>
              <Var>toc</Var>
              <Var>chapters</Var>
            </Expr>
          </Atom>
        </if>
        <then>
          <Equal>
            <Ind>Bin</Ind>
            <Ind>Tom</Ind>
          </Equal>
        </then>
      </Implies>
 </Assert>
```

- **Newly created Hornlog+ example**

```
 <Assert>
    <Implies>
      <Atom>
        <Rel>worksAt</Rel>
        <Var>Person</Var>
        <Var>Company</Var>
      </Atom>
      <Exists>
        <Var>Superior</Var>
        <And>
          <Atom>
            <Rel>employeeOf</Rel>
            <Var>Superior</Var>
            <Expr>
              <Fun>headquartersOf</Fun>
              <Var>Company</Var>
            </Expr>
          </Atom>
          <Atom>
            <Rel>reportsTo</Rel>
            <Var>Person</Var>
            <Var>Superior</Var>
```

```
            </Atom>
          </And>
        </Exists>
      </Implies>
    </Assert>
```

Project Tools:

1. XML editor: oXygen
   http://www.oxygenxml.com/

2. RNC validation tool: http://validator.nu/

References:

[1] Specification of Deliberation RuleML 1.01
http://wiki.ruleml.org/index.php/Specification_of_Deliberation_RuleML_1.01

[2] MYNG1.01- the Deliberation RuleML Schema Selection Form
http://deliberation.ruleml.org/1.01/myng/

[3] The MYNG 1.01 Suite for Deliberation RuleML 1.01: Taming the Language Lattice
http://ceur-ws.org/Vol-1211/paper/.pdf

[4] RuleML 1.01 Examples (grouped according to sublanguages):
http://deliberation.ruleml.org/1.01/exa

[5] W3C RIF Test Cases:
http://www.w3.org/2005/rules/wiki/Category:Test_Case

[6] Demo of MYNG 1.01:
http://wiki.ruleml.org/index.php/Demo_of_MYNG_1.01