Project Proposal: POSL to d-POSL

Making the Positional-Slotted Language Defeasible

Team #1:

Daniel Latimer
Hanin Almutairi
Usman Ali

Instructor: Harold Boley

Table of Contents

Introduction

POsitional **S**lotted **L**anguage or POSL is a rule based language that provides a more human readable format for web rules. It is compatible, and convertible with the more generic xml based RuleML which has a format more suited for machine readability. The current version of POSL needs some modification to bring its syntax more in line with the new version of RuleML. Uniform Resource Identifiers (URI) has been renamed to the more appropriate Internet Resource Identifier (IRI). The syntax to type a variable or independent has been changed from a colon (:) to a double hat (^^). This will allow us to use the colon as a namespace identifier as it is being used in RuleML.

One thing POSL currently lacks is a mechanism for defining defeasible rules. Defeasible logic allows certain rules to override other rules. By including a superiority list of which rule beats which we are able to define the priority of our rules and allow knowledge-bases which have an incomplete understanding of their domain to function in an appropriate manner. This sort of prioritization of rules is important to practical applications such as knowledge-bases in the domains of law and business. It is also "useful for ontology integration, where conflicting information arises naturally" [1].

Objectives

Currently POSL is at version 0.91, our project will modify the grammar to align it with the changes that occurred between RuleML 0.91 and 1.0. We will then add functionality to OO jDrew to read POSL 1.0 and make its default output when translating RuleML to POSL to be version 1.0.

Another purpose of our project will be to write a defeasible POSL grammar. The grammar will be created by taking the updated POSL 1.0 grammar and adding support for rule labels, superiority relations, and adding syntax to denote whether a rule is strict, defeasible, or a defeater.

Methodology

- Use the ANTLRWorks to define the original POSL grammar in a way that it interacts with Java instead of Python.
- Update the grammar according to new standards for POSL 1.0 as in the use of IRI and HATHAT ^^.
- Generate the grammar`s lexer, parser and tokens and integrate them into OO jDREW to implement the new POSL standard.
- The original grammar for DPOSL project version 1 was fixed with changes however, it still lacked key features like rule labels and rule types as well as conflicting literals, that made it necessary to start the grammar from scratch.
- Develop a new grammar for DPOSL as, envisioned in reference [3], from scratch to be able to tackle DPOSL statements. However the implementation of the finished grammar into a software package will be implemented in future work.

Project Tools

ANTLR

ANother **T**ool for **L**anguage **R**ecognition (ANTLR) is a language tool that provides a framework for constructing recognizers, compilers, and translators from grammatical descriptions containing actions in a variety of target languages. ANTLR automates the construction of language recognizers. From a formal grammar, ANTLR generates a program that determines whether sentences conform to that language. In other words, it's a program that writes other programs. By adding code snippets to the grammar, the recognizer becomes a translator or interpreter.

ANTLRWorks

ANTLRWorks is a novel grammar development environment for ANTLR v3 grammars written by Jean Bovet (with suggested use cases from Terence Parr). It combines an excellent grammar-aware editor with an interpreter for rapid prototyping and a language-agnostic debugger for isolating grammar errors.

Figure 4.1 ANTLRWorks

Eclipse

Eclipse is an open source community, whose projects are focused on building an open development platform comprised of extensible frameworks, tools and runtimes for building, and for deploying and managing software across the lifecycle.

OO jDREW

OO jDREW is a deductive reasoning engine for the RuleML web rule language; it is an Object Oriented extension to jDREW. OO jDREW implements Object Oriented extensions to RuleML which include:

1. Order Sorted Types
2. Slots
3. Object Identifiers

OO jDREW is written in the Java programming language (Tested with Version 1.6.0).

OO jDrew provides both Bottom-up as well as Top-down reasoning. Bottom-up execution is used to infer all derivable knowledge from a set of clauses (forward reasoning). Top-down execution is used to solve a query on the knowledge base (backward reasoning)

Supported Languages

POSL
POSL is a Positional Slotted Language that implements RuleML shorthand (similar to prolog).

RuleML
OO jDREW supports the XML syntax of RuleML.

References

[1] DR-DEVICE: A Defeasible Logic Reasoner for the Semantic Web <http://lpis.csd.auth.gr/systems/dr-device.html>

 [2] POSL: An Integrated Positional-Slotted Language for Semantic Web Knowledge. Harold Boley. 11 May 2004 <http://ruleml.org/submission/ruleml-shortation.html>

[3] Efstratios Kontopoulos, Nick Bassiliades, Grigoris Antoniou, "Visualizing Semantic Web proofs of defeasible logic in the DR-DEVICE system, Knowledge-Based System", Volume 24, Issue 3, April 2011, Pages 406-419, ISSN 0950-7051, 10.1016/j.knosys.2010.12.001. <http://www.sciencedirect.com/science/article/pii/S0950705110001735>

[4] ANTLR. Terence Parr. <http://www.antlr.org/>

[5] Boley, Harold. "Integrating Positional and Slotted Knowledge on the Semantic Web". Journal of Emerging Technologies in Web Intelligence. Vol 2, No 4 (2010), Nov 2010, 343-353.

[6] Eclipse web site. <http://www.eclipse.org/org/>

[7] RuleML website. <http://ruleml.org/oojdrew/>