A Perfect Match for Reasoning, Explanation, and Reason Maintenance: OWL 2 RL and Semantic Wikis

Jakub Kotowski and François Bry

Institute for Informatics, University of Munich
http://pms.ifi.lmu.de

Abstract. Reasoning in wikis has focused so far mostly on expressiveness and tractability and neglected related issues of updates and explanation. In this demo, we show reasoning, explanation, and incremental updates in the KiWi wiki and argue that it is a perfect match for OWL 2 RL reasoning. Explanation nicely complements the "work-in-progress" focus of wikis by explaining how which information was derived and thus helps users to easily discover and remove sources of inconsistencies. Incremental updates are necessary to minimize reasoning times in a frequently changing wiki environment.

1 Introduction

One of the main goals of the semantic web [1] is to facilitate processing of information on the web for example by means of reasoning. Semantic wikis [2] are sometimes seen as semantic webs in small; they enhance traditional wikis with semantic annotations in order to make more information directly amenable to machine processing. On the web and even more in wikis, it is natural that inconsistencies arise during work in progress. Users should be supported by a system that not only tolerates inconsistencies but is also able to explain them. The focus of reasoning in KiWi[1] is therefore on a rule-based inconsistency tolerant reasoning that can be explained to users and that also allows for efficient knowledge base updates by the means of reason maintenance. This article describes the state of implementation as of KiWi version 0.8.

Current semantic web applications and frameworks such as Sesame [3], Semantic MediaWiki [4], or IkeWiki [5] implement either specialized RDF/S [2] reasoning or connect a specialized OWL-DL [3] reasoner such as Pellet [6] to provide more expressive reasoning. For example Sesame aims to be a general platform for semantic software based on RDF/S and therefore provides reasoning optimized for RDF/S data. For the Semantic MediaWiki, scalability is one of the top priorities which is why it limits its reasoning to the most efficient forms. In contrast,

[1] http://www.kiwi-project.eu/
[2] http://www.w3.org/TR/rdf-mt/
[3] http://www.w3.org/TR/owl2-profiles/

the Jena [7] framework, also provides custom rules and some support for dealing with inconsistencies in a dataset via so called validation rules. Both Sesame and Jena have only limited support for incremental updates. Jena employs a general purpose RETE-based forward-chaining reasoner which supports incremental additions but no incremental removals[4] [5]. Sesame itself has a limited support for custom rule reasoning and does not offer incremental removals in the general case. There is a reason-maintenance-inspired implementation of incremental updates for Sesame [8] but it is specific to RDF/S reasoning. For Sesame, there is also the OWLIM [6] reasoner which, however, also does not support incremental removals[7]. A contribution of the described implementation is a system capable of incremental processing of fact removals for a general monotonic rule-based reasoner. In addition, it can be easily extended for incremental rule updates.

2 Introduction to KiWi and sKWRL

KiWi is a social semantic platform that features four advanced enabling technologies: reasoning and reason maintenance, querying, information extraction, and personalization and has a wiki as its main application. See [9] for details about the KiWi conceptual model.

sKWRL is a simple KiWi rule language the syntax of which resembles the syntax of the N3 [10] language. It can express a subset[8] of OWL 2 RL – a partial axiomatization of the OWL 2 RDF-based semantics using rules[9] – and has been implemented to provide a starting point for reasoning, explanation, and reason maintenance and also as a step towards the full featured KWRL rule language. sKWRL is a triple pattern based rule language for RDF with two distinctive features: constraint rules and new resource creation in rule heads.

Triple pattern is a generalized RDF triple that can contain a variable in place of subject, property, and object. Bodies of a sKWRL rule consist of a conjunction of triple patterns. Head of a sKWRL rule contains either a conjunction of triple patterns or the "*inconsistency*" keyword. Rules with the "*inconsistency*" keyword in the head are called constraint rules, rules with a conjunction of triple patterns are called construction rules. All variables are implicitly universally quantified and there is no explicit quantification. Variables occurring in a rule head that do not occur in the body of a rule are allowed and construct a new URI reference for each variable binding of the rule body.

One or more sKWRL rules form together a sKWRL program. sKWRL programs can optionally use namespace definitions in a Turtle-like style. An example of a simple sKWRL program is a program deriving the RDF/S subclass and type hierarchy:

[4] http://tech.groups.yahoo.com/group/jena-dev/message/43618
[5] The focus of this paper the focus is on forward-chaining methods.
[6] http://www.ontotext.com/owlim/
[7] http://www.mail-archive.com/owlim-discussion@ontotext.com/msg00496.html
[8] Datatypes and OWL 2 RL rules that use a LIST[] expression are not supported yet.
[9] http://www.w3.org/TR/owl2-profiles/#OWL_2_RL

```
@prefix rdf : <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>

rdf-type: ($1 rdf:type $2), ($2 rdfs:subClassOf $3)
                                     -> ($1 rdf:type $3)
rdf-subclass: ($1 rdfs:subClassOf $2), ($2 rdfs:subClassOf $3)
                                     -> ($1 rdfs:subClassOf $3)
```

Following is an example of a constraint rule:

```
prp-irp: ($p rdf:type owl:IrreflexiveProperty), ($x $p $x)
                                   -> inconsistency
```

The *prp-irp* rule is one of the OWL 2 RL rules with "false" in the head. Currently, rules are loaded from an external file and cannot be modified from within the application.

Internally, the keyword "inconsistency" is replaced by a triple pattern conjunction constructing an annotation for every derived inconsistency. This is needed for explanation purposes for it allows to "track inconsistencies", see [11] for more about tracking. The inconsistency annotation is assigned to the default RDF graph and, in future, optionally to a graph specified by the user. Inconsistencies are displayed as "inconsistency" tags and are highlighted.

3 Implementation

sKWRL is implemented as a component of KiWi which is an enterprise Java application built using Seam[10] and the JBoss application server. The implemented reasoning strategy is semi-naive forward-chaining, also called materialization, which has already been argued to be feasible [8] for applications in the area of semantic web.

sKWRL reasoning is implemented by translating sKWRL rule bodies into JQL (a Java Peristence API version of SQL). The advantage of this approach is the database flexibility provided by JPA, the disadvantage is the inability to use a native database access, which hinders efficiency. The KiWi implementation should therefore be seen as a proof of concept not aiming for high efficiency.

The reasoner also stores derivations of each new derived triple in the form of a *justification* in Doyle's sense [12], i.e. a record of which triples and rules were used in the derivation of a triple. Justifications are then used by reason maintenance and explanation.

4 Reason maintenance

Reason maintenance is a technique originally devised by Jon Doyle [12] for use in problem solvers. A reason maintenance system works closely with a reasoner.

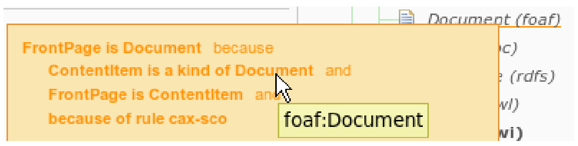
[10] http://seamframework.org/

The reasoner notifies reason maintenance about each derivation it makes and reason maintenance stores derivations in the form of a derivation graph. In the original systems, this graph was used as a kind of computation cache which helped to avoid the need of recomputing in case some base facts changed (i.e. were removed and later added again). Therefore, these systems never removed justifications. In contrast, KiWi uses justifications to determine what facts can possibly be affected by a fact removal thus avoids the inefficient, not incremental approach which is to remove all inferred facts and to do all reasoning anew.

Reason maintenance in KiWi is implemented using the Neo4j [11] NoSQL graph database which natively supports graph structured data.

5 Explanation

Explanation is important for supporting trust of users and it provides a way to determine the root cause of derived inconsistencies. Currently, explanation explains the origin of a derived triple simply by rendering its justification records. There are two renderings available: textual tooltips and an interactive JavaScript explanation tree.



FrontPage is Document because
ContentItem is a kind of Document and
FrontPage is ContentItem and
because of rule cax-sco

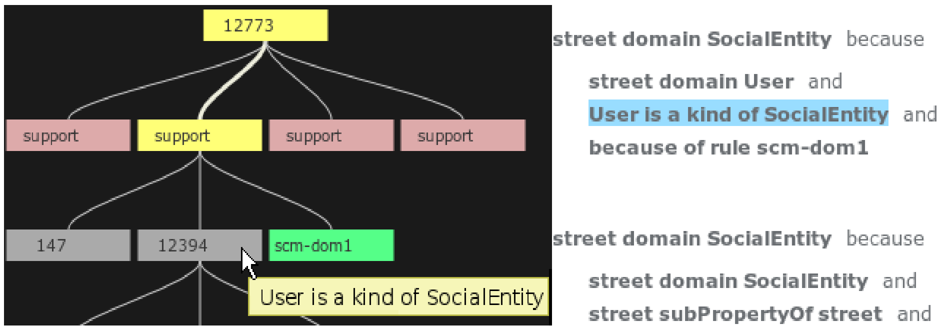
Fig. 1. Part of an explanation tooltip for the triple (localhost:FrontPage rdf:type foaf:Document).

Explanation tooltips present a simple textual explanation of a derived triple, see Fig. 1. The tooltip shows the last step of each possible derivation of the triple "*localhost:FrontPage rdf:type foaf:Document*". The implementation uses a minimal vocabulary to translate common properties into a more readable form.

The explanation tree, see Fig. 2, enables users to explore a graph of all possible derivations and to traverse them until explicit triples are reached. The explanation tree is complemented by a textual explanation, parts of which are highlighted by pointing to a tree node.

Acknowledgements. The research leading to these results is part of the project "KiWi - Knowledge in a Wiki" and has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 211932.

[11] http://neo4j.org/



street domain SocialEntity because

street domain User and
User is a kind of SocialEntity and
because of rule scm-dom1

street domain SocialEntity because

street domain SocialEntity and
street subPropertyOf street and

Fig. 2. An interactive explanation tree and a part of textual explanation of the triple *(kiwi:street rdfs:domain swap:SocialEntity)*. Numbers in graph nodes are triple ids of the corresponding KiWi triples. Green nodes represent rules, support nodes represent justifications and the currently selected derivation path is highlighted in yellow. Support nodes can be expanded and collapsed.

References

1. Berners-Lee, T., Hendler, J.: Scientific publishing on the semantic web. Nature **410** (2001) 1023–1024
2. Schaffert, S., Bry, F., Baumeister, J., Kiesel, M.: Semantic wikis. IEEE (2008)
3. Broekstra, J., Kampman, A., Van Harmelen, F.: Sesame: A generic architecture for storing and querying rdf and rdf schema. LNCS (2002)
4. Krötzsch, M., Vrandecic, D., Völkel, M.: Semantic mediawiki. In: ISWC. Volume 6., Springer (2006) 935–942
5. Schaffert, S.: IkeWiki: A semantic wiki for collaborative knowledge management. In: 1st International Workshop on Semantic Technologies in Collaborative Applications (STICA06), Manchester, UK, Citeseer (2006)
6. Sirin, E., Parsia, B., Grau, B., Kalyanpur, A., Katz, Y.: Pellet: A practical owl-dl reasoner. Journal of Web Semantics **5**(2) (2007) 51–53
7. McBride, B.: Jena: A semantic web toolkit. IEEE Internet Computing (2002)
8. Broekstra, J., Kampman, A.: Inferencing and truth maintenance in rdf schema - exploring a naive practical approach. Workshop on Practical and Scalable Semantic Systems (PSSS) (2003)
9. Bry, F., Eckert, M., Kotowski, J., Weiand, K.: What the user interacts with: Reflections on conceptual models for sematic wikis. In: Proceedings of Semantic Wiki Workshop, ESWC, Greece, 2009. (2009)
10. Berners-Lee, T., Connolly, D., Kagal, L., Scharf, Y., Hendler, J.: N3Logic: A logical framework for the World Wide Web. Theory and Practice of Logic Programming **8**(03) (2008) 249–269
11. Bry, F., Kotowski, J.: A social vision of knowledge representation and reasoning. In: Proceedings of SOFSEM 2010: 36th International Conference on Current Trends in Theory and Practice of Computer Science, Špindlerův Mlýn, Czech Republic (23rd–29th January 2010). (2010)
12. Doyle, J.: Truth maintenance systems for problem solving. Technical Report AI-TR-419, Dep. of Electrical Engineering and Computer Science of MIT (1978)