Reasoning in Expressive Extensions of the RDF Semantics

Michael Schneider

FZI Research Center for Information Technology, Karlsruhe, Germany
schneid@fzi.de

Abstract. The research proposed here deals with reasoning in expressive semantic extensions of the RDF Semantics specification, up to the level of OWL 2 Full. The work aims to conduct an in-depth study of the distinctive features and the degree of implementability of OWL Full reasoning. This paper describes the core problem, presents the proposed approach, reports on initial results, and lists planned future tasks.

Keywords: Semantic Web, Reasoning, RDF Semantics, OWL Full.

1 Problem Statement and State of the Art

This research deals with reasoning in expressive semantic extensions of the RDF Semantics specification [5]. The focus will specifically be on the ontology language OWL 2 Full [9], which has been standardized by the World Wide Web Consortium (W3C) in 2009 as an RDFS-compatible flavor of OWL that essentially covers all other members of the RDF and OWL language families. Several W3C languages have dependencies on OWL Full, including SKOS, RIF, and the current revision of SPARQL ("SPARQL 1.1"). So far, however, OWL Full has largely been ignored by the research community and no practically applicable reasoner has been implemented.

The current situation may have a variety of reasons. The most frequently heard technical argument against OWL Full reasoning is that OWL Full is *computationally undecidable* with regard to key reasoning tasks [7]. However, undecidability is a common theoretic problem in other fields as well, as for first-order logic reasoning, which still has many highly efficient implementations with relevant industrial applications [11]. Nevertheless, the undecidability argument and other arguments have led to strong reservations about OWL Full and have effectively prevented researchers from studying the distinctive features of the language and from searching for methods to realize at least useful *partial* implementations of OWL Full reasoning. But without a better understanding of OWL Full reasoning and its relationship to other reasoning approaches it will not even become clear what the added value of an OWL Full reasoner would be.

An OWL Full reasoner would make the full expressivity of OWL available to unrestricted RDF data on the Semantic Web. A conceivable use case for OWL Full reasoners is to complement RDF rule reasoners in reasoning-enabled

G. Antoniou et al. (Eds.): ESWC 2011, Part II, LNCS 6644, pp. 487–491, 2011.

applications operating on weakly-structured data to provide for significantly enhanced reasoning power. RDF rule reasoners, such as those implementing the OWL 2 RL/RDF rules [8], typically allow for efficient and scalable reasoning on arbitrary RDF data, but their reasoning capabilities are intrinsically limited, in particular with regard to terminological reasoning. OWL Full, on the other hand, offers very high reasoning expressivity, roughly comparable to that of OWL DL and even beyond. In a reasoning-enabled application, an RDF rule reasoner could rapidly produce the bulk of easily derivable results, while more thorough inferencing could be delegated to an OWL Full reasoner. Interoperability would often be warranted, since OWL Full applies to arbitrary RDF graphs as well, and is semantically fully compatible with RDFS and the OWL 2 RL/RDF rules. In contrast, OWL DL reasoners cannot generally be expected to work reliably in this use case due to the many syntactic restrictions and the differing semantics of OWL DL compared to the RDF Semantics.

As said, only little effort has been spent in the implementation of OWL Full reasoning so far. One general idea that can be found in the literature is to translate the native semantics of an ontology language into a first-order logic (FOL) axiomatisation and to use an automated theorem prover (ATP) for reasoning. An early application of this approach to a preliminary version of RDF and a precursor of OWL has been reported by Fikes et al. [1]. The focus of this work was, however, more on identifying technical problems in the original language specifications rather than on practical reasoning. Hayes [4] provides fairly complete translations of RDF(S) and OWL 1 Full into Common Logic, but does not report on any reasoning experiments. This gap is filled by Hawke's reasoner *Surnia* [3], which applies an ATP to an FOL axiomatisation of OWL 1 Full. For unknown reasons, however, Surnia performed rather poorly on reasoning tests [12]. Comparable studies have also been carried out for ATP-based OWL DL reasoning, which have shown more promising results [10].

This research is going to conduct an in-depth study of the features and the implementability of OWL Full reasoning. More precisely, the following research questions will be investigated: Firstly, *what are the distinctive features of OWL 2 Full compared to other approaches used for Semantic Web reasoning?* Secondly, *to which degree and how can OWL 2 Full reasoning be implemented?*

2 Proposed Approach and Methodology

The investigation of the two research questions will be carried out in the form of a *feature analysis* and an *implementability analysis*.

The goal of the **feature analysis** is to create a systematic and comprehensive *catalogue of distinctive pragmatic features* of OWL 2 Full. Both syntactic and semantic aspects of the language will be taken into account. A feature will be called *distinctive*, if it is not supported by either OWL 2 DL or by typical RDF rule reasoners, as those implementing the OWL 2 RL/RDF rules. For example, a distinctive *syntactic-aspect feature* might be the ability to assert a disjointness axiom between two annotation properties (as for SKOS lexical labels), while a distinctive *semantic-aspect feature* might be the ability to draw

logical conclusions from such an axiom; neither is supported by OWL 2 DL. For each identified feature, a precise description, a motivation, an explanation for its distinctiveness, and one or more concrete examples will be given. Identification of the features may make use of any kind of source, including literature, ontologies, forum discussions, or technical aspects of the language. For each candidate feature, concrete evidence will be searched in order to support its validity.

The **implementability analysis** will concentrate on studying the *FOL translation approach*, as mentioned in Sec. 1. This approach has the advantage that it applies to arbitrary extensions of the RDF Semantics and it enjoys strong and mature reasoning tool support through existing ATPs. The idea is to create a corresponding FOL formula for every model-theoretic *semantic condition* of the OWL 2 Full semantics. For example, the semantic condition for class subsumption, as given in Sec. 5.8 of [9], can be translated into the FOL formula

$$\forall c, d : \mathrm{iext}(\mathrm{rdfs{:}subClassOf}, c, d) \Leftrightarrow \mathrm{ic}(c) \wedge \mathrm{ic}(d) \wedge \forall x : [\mathrm{icext}(c, x) \Rightarrow \mathrm{icext}(d, x)] \,.$$

An *RDF triple* '`:s :p :o`' is mapped to an atomic FOL formula '$\mathrm{iext}(p, s, o)$'. An *RDF graph* is translated into a conjunction of such 'iext' atoms, with existentially quantified variables representing blank nodes. An *entailment query* is represented by the conjunction of the OWL Full axiomatisation, the translation of the premise graph, and the negated translation of the conclusion graph. *Entailment checking* can then be performed by means of an ATP.

The implementability analysis will be based on a prototypical reasoner that is going to be built from an ATP, an FOL axiomatisation of the OWL 2 Full semantics, and a converter for translating RDF graphs into FOL formulas. The reasoner will then be evaluated based on the identified distinctive OWL Full features. This will be done by using the created concrete feature examples as test cases for conformance and performance testing of the parsing and the reasoning capabilities of the reasoner. Conversely, this method will help ensuring that the identified distinctive features will be technically valid. The evaluation results will be compared to those for OWL DL reasoners and RDF rule reasoners.

To the author's knowledge, the proposed research will be the first in-depth study of the features and the implementability of OWL 2 Full reasoning. No analysis of the distinctive pragmatic features of OWL 2 Full has been done so far. Also, there has been no rigorous analysis of OWL 2 Full reasoning based on the FOL translation approach yet.

3 Current Status and Initial Results

The **syntactic-aspect feature analysis** has been partially completed for the OWL 1 subset of OWL 2 Full. This work resulted in a catalogue of 90 features that have been grouped into 14 categories. The features were often motivated by data on the public Semantic Web. Example ontologies for all the features were used in the EU project *SEALS* (`http://seals-project.eu`) as test cases for the evaluation of ontology engineering tools. Project deliverable D10.3 reports on various problems that have been observed when using the OWL DL-centric OWL API (`http://owlapi.sourceforge.net`) with these test cases.

Table 1. Results for a test suite of 32 characteristic OWL Full conclusions

Reasoner / Mode	Success	Failure	Unknown	System Error
Pellet 2.2.2 / OWL-API 3.1	9	22	0	1
BigOWLIM 3.4 / owl2-rl	9	23	0	0
iProver 0.8 / all OWL Full axioms	28	0	4	0
iProver 0.8 / sufficient axioms only	32	0	0	0

For the **semantic-aspect feature analysis**, work on the development of several reasoning test suites has been started. The test suites are designed according to diverse criteria, such as language coverage, reasoning difficulty, and realism. One of these test suites consists of "characteristic" OWL 2 Full conclusions from e.g. meta-modeling, annotation properties, unrestricted use of complex properties, and "vocabulary reflection". They have been collected from forum discussions and other sources. The first two rows of Table 1 show the results of applying the test suite to the OWL 2 DL reasoner *Pellet* (`http://clarkparsia.com/pellet`) and the OWL 2 RL/RDF rule reasoner *OWLIM* (`http://ontotext.com/owlim`). Both reasoners succeeded on only a small fraction of the test suite, and they did so conjointly only on *two* of the test cases.

For the **implementability analysis**, an FOL axiomatisation has been created for a major fragment of the OWL 2 Full semantics, the main omission being support for datatype reasoning. The used FOL dialect is the language for encoding *TPTP* problems (`http://tptp.org`). A converter from RDF graphs to TPTP formulas has also been implemented. This enables the use of the OWL Full axiomatisation with a large number of existing ATPs.

The third row of Table 1 shows the results from applying the same test suite to the ATP *iProver* (`http://code.google.com/p/iprover`). When using the complete OWL Full axiomatisation, iProver succeeded on most of the test cases, but for a few test cases it did not terminate within a time limit of 300 seconds. Further analysis suggested that this issue may be due to the large number of complex OWL 2 Full axioms. The complete axiomatisation was then, separately for each test case, manually reduced to small sub-axiomatisations that were just sufficient to entail the expected result. This allowed iProver to succeed on *all* test cases (fourth row). In this scenario, reasoning typically finished within a few hundredth of a second on a standard personal computer, compared to often several seconds for the complete axiomatisation. These and additional results from experiments concerning OWL 2 Full language coverage, scalability and model-finding have been submitted to CADE 2011.

4 Conclusions and Future Work

This paper proposed a first in-depth study of the distinctive features and the degree of implementability of OWL 2 Full reasoning. First results indicate that one can use ATPs and an FOL axiomatisation of the OWL 2 Full semantics for genuine OWL Full reasoning beyond the capabilities of typical RDF rule and OWL DL reasoners. The approach is flexible enough to implement arbitrary extensions of the RDF Semantics or to add features such as rule-style reasoning.

It has been observed that acceptable reasoning performance can often only be achieved by reducing the whole OWL Full axiomatisation to a small sufficient sub-axiomatisation. A next step will therefore be to search for an automated method to *eliminate redundant axioms* with regard to the given input ontology.

So far, the implementability analysis was restricted to entailment and inconsistency checking. To fulfil the discussed use case of complementing RDF rule reasoners in reasoning-enabled applications, OWL Full reasoners should also support flexible *query answering* on arbitrary RDF data. This will specifically be needed to realize the *OWL 2 RDF-Based Semantics entailment regime* of SPARQL 1.1 [2]. Some ATPs have been reported to offer query answering on FOL knowledgebases [6]. It will be analyzed to what extent these capabilities can be exploited for OWL Full query answering.

For the *syntactic-aspect feature analysis*, which has already been finished for OWL 1 Full, the remaining work will be to extend the analysis to the whole of OWL 2 Full. The *semantic-aspect feature analysis* is in a less-complete state and still requires the development of a feature categorization similar to that of the syntactic-aspect feature analysis. The started work of building reasoning test suites will be continued and will eventually lead to a collection of concrete examples for the still to-be-identified semantic-aspect features.

References

1. Fikes, R., McGuinness, D., Waldinger, R.: A First-Order Logic Semantics for Semantic Web Markup Languages. Tech. Rep. KSL-02-01, Knowledge Systems Laboratory, Stanford University, Stanford, CA 94305 (January 2002)
2. Glimm, B., Ogbuji, C. (eds.): SPARQL 1.1 Entailment Regimes. W3C Working Draft (October 14, 2010)
3. Hawke, S.: Surnia (2003), `http://www.w3.org/2003/08/surnia`
4. Hayes, P.: Translating Semantic Web Languages into Common Logic (July 18, 2005), `http://www.ihmc.us/users/phayes/CL/SW2SCL.html`
5. Hayes, P. (ed.): RDF Semantics. W3C Recommendation (February 10, 2004)
6. Horrocks, I., Voronkov, A.: Reasoning Support for Expressive Ontology Languages Using a Theorem Prover. In: Dix, J., Hegner, S.J. (eds.) FoIKS 2006. LNCS, vol. 3861, pp. 201–218. Springer, Heidelberg (2006)
7. Motik, B.: On the Properties of Metamodeling in OWL. Journal of Logic and Computation 17(4), 617–637 (2007)
8. Motik, B., Grau, B.C., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C. (eds.): OWL 2 Web Ontology Language: Profiles. W3C Recommendation (October 27, 2009)
9. Schneider, M. (ed.): OWL 2 Web Ontology Language: RDF-Based Semantics. W3C Recommendation (October 27, 2009)
10. Tsarkov, D., Riazanov, A., Bechhofer, S., Horrocks, I.: Using Vampire to Reason with OWL. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 471–485. Springer, Heidelberg (2004)
11. Voronkov, A.: Automated Reasoning: Past Story and New Trends. In: Proc. IJCAI 2003, pp. 1607–1612 (2003)
12. W3C WebOnt OWL Working Group: OWL 1 Test Results (March 9, 2004), `http://www.w3.org/2003/08/owl-systems/test-results-out`