RDFS(FA): A DL-ised Sub-language of RDFS

Jeff Z. Pan and Ian Horrocks

Department of Computer Science
University of Manchester
Oxford Road, Manchester M13 9PL, UK
`{pan,horrocks}@cs.man.ac.uk`

Abstract

Description Logics (DLs), as a field of research, form a formal foundation of first-order semantic Web ontology languages, such as DAML+OIL and OWL. The Semantic Web will build on XML's ability to define customized tagging schemes and RDF(S)'s flexible approach to representing data. RDF Schema (RDFS), however, has a non-standard metamodeling architecture, which makes some elements in the model have dual roles in the RDFS specification. As a result, the specification of its semantics requires a non-standard model theory, RDF MT. This leads to semantic problems when trying to layer DL-based and other conventional first-order Web ontology languages on top of RDFS. In this paper we suggest layering Web ontology languages on top of RDFS(FA), a sub-language of RDFS, which is based on a (relatively) standard model-theoretic semantics. We will also compare this approach with the existing RDF Model Theory and show how their semantics relate to the semantics of OWL, a newly developed Web ontology language by W3C.

1 Introduction

Description Logics (DLs)[1], a family of logical formalisms for the representation of and reasoning about conceptual knowledge, are of crucial importance to the development of the Semantic Web [2]. In the Semantic Web, a vision of the next generation Web, the existing rendering markup, which specifies how to display web resources for human consumption, will be supplemented with explicit machine-understandable *semantic* markup (often called "metadata"), which will specify the meaning of web resources and services so as to make them more readily accessible to automatic processes. The role of DLs is to provide formal underpinnings and automated reasoning services for first-order semantic Web ontology languages [10] such as DAML+OIL[1] [6] and OWL[2].

In a functional architecture [2, 11] of semantic Web languages, Web ontology languages are expected to stand on top of RDFS [3], which provides rather simple facilities to define classes and properties used in annotations, to supply a richer set of modelling primitives. Unfortunately, the relationships between RDFS and Web ontology languages aren't clearly specified.

---

[1] `http://www.daml.org/`

[2] `http://www.w3.org/2001/sw/WebOnt/`

Initially it was not possible to define the semantics of Web ontology languages as extensions of the semantics of RDFS, because RDF and RDFS had no formal model theory, nor any formal meaning at all. E.g., when DAML+OIL was layering on top of RDFS, it used the syntax of RDFS only, and defined its own semantics [15], even for the ontological primitives present in RDFS.

As earlier works [9, 4] pointed out, RDFS has a non-standard and non-fixed layer metamodeling architecture, which makes some elements in the model have dual roles in the RDFS specification. As a result, the specification of its semantics requires a non-standard model theory, RDF MT [5]. This leads to semantic problems [12, 13, 7] when trying to layer DL-based (or other conventional first-order) Web ontology languages on top of RDFS.

In this paper we suggest layering Web ontology languages on top of RDFS(FA) [11][3] [4], a sub-language of RDFS, which is based on a (relatively) standard model-theoretic semantics. The implicitly represented modelling primitives in RDFS are explicitly stratified into different strata (layers) of RDFS(FA). The new modelling primitives introduced by DL-based (or other conventional first-order) Web ontology languages are syntactically and semantically located in stratum 1 and 2.

In the remainder of this paper, we will first briefly describe dual roles in RDFS (Section 2). We will then present how RDFS(FA) and RDF MT clear up any possible confusion of RDFS individually (Section 3 and 4). We will then explain the relationships between the semantics of OWL and those of RDFS(FA) and RDF MT (Section 5). Finally we will discuss what conclusions we can draw from the above comparison in Section 6.

2 Dual Roles in RDFS

The Resource Description Framework (RDF) [8] and its schema extension, RDF Schema (RDFS) [3] form the lowest two layers of the Semantic Web. RDF is intended to provide a foundation for processing metadata, which will provide interoperability between applications that exchange machine-readable information on the Semantic Web. RDFS provides a standard mechanism for declaring classes and (global) properties, as well as defining relationships between classes and properties, using RDF syntax.

RDFS, however, has a non-standard and non-fixed layer metamodeling architecture, which makes some elements in the model appear to have dual (or multiple) roles [9, 4, 11]. E.g., there is a strange situation for rdfs:Class and rdfs:Resource as discussed in [11]. On the one hand, rdfs:Resource is an instance of rdfs:Class. On the other hand, rdfs:Class is a sub-class of rdfs:Resource. Thus rdfs:Resource is an instance of its sub-class?![5] Many people find it confusing. Up to now, there are at least two ways to clear up any confusion and give a clear semantics to the schema language: RDFS(FA) and RDF MT.

---

[3] `http://DL-Web.man.ac.uk/rdfsfa/`

[4] Note that RDFS(FA) predates RDF MT.

[5] This might partially explain why Brickley and Guha [3] didn't define a semantics for RDFS.

3 RDFS(FA)

In [11] we proposed a sub-language of RDFS - RDFS(FA), which provides a Fixed layer metamodeling Architecture for RDFS. RDFS(FA) eliminates dual roles by defining the modelling primitives *explicitly*, instead of implicitly.

The universe of discourse is divided up into different strata (layers). Built-in modelling primitives of RDFS are stratified into different strata of RDFS(FA), so that certain modelling primitives belong to a certain stratum (layer).

Let V be a vocabulary, which is a set of urirefs. V is divided into disjoint sets $V_0, V_1, V_2, \ldots$, the vocabularies used in strata 0,1,2 ... respectively. Let $\mathbf{R}_i$, $\mathbf{C}_i$, $\mathbf{P}_i$ be the modelling primitives which are interpreted as the sets of all elements, all classes and all properties respectively in stratum i. Let $\mathbf{D}_i$ be the domain in stratum i and **IE** be an interpretation function.

The pair $\langle IR, IE \rangle$ is an interpretation for RDFS(FA)[6], where

$$IR = \mathbf{D}_0 \cup \mathbf{D}_1 \cup \mathbf{D}_2 \cup \cdots = IE(\mathbf{R}_0) \cup IE(\mathbf{R}_1) \cup IE(\mathbf{R}_2) \cdots .$$

We start from stratum 0. Every individual name $x \in V_0$ is mapped to an object in the domain $\mathbf{D}_0$:

$$IE(x) \in \mathbf{D}_0.$$

In stratum $i+1$ (where $i = 0, 1, 2, \ldots$), the domain $\mathbf{D}_{i+1}$ is equal to the union of the set of all classes and the set of all properties in stratum $i+1$:

$$\mathbf{D}_{i+1} = IE(\mathbf{R}_{i+1}) = IE(\mathbf{C}_{i+1}) \cup IE(\mathbf{P}_{i+1})$$

Each class primitive $c_{i+1} \in V_{i+1}$ is interpreted as a set of elements in stratum i:

$$IE(c_{i+1}) \subseteq IE(\mathbf{R}_i),$$

and each property primitive $p_{i+1} \in V_{i+1}$ is interpreted as a set of pairs of elements in stratum i:

$$IE(p_{i+1}) \subseteq IE(\mathbf{R}_i) \times IE(\mathbf{R}_i).$$

The $type_{i+1}$ property is interpreted as a set of pairs, where the first element is in stratum i, and the second element is a class in stratum $i+1$:

$$IE(type_{i+1}) \subseteq IE(\mathbf{R}_i) \times IE(\mathbf{C}_{i+1}).$$

Since $IE(c_{i+1}) \subseteq IE(\mathbf{R}_i)$, we have $IE(c_{i+1}) \in 2^{IE(\mathbf{R}_i)}$, i.e. $IE(c_{i+1}) \in 2^{\mathbf{D}_i}$. According to the definition of $\mathbf{C}_{i+1}$, we have $IE(\mathbf{C}_{i+1}) = 2^{\mathbf{D}_i}$. Similarly, we have $IE(\mathbf{P}_{i+1}) = 2^{\mathbf{D}_i \times \mathbf{D}_i}$. Since $IE(\mathbf{R}_{i+1}) = \mathbf{D}_{i+1} = IE(\mathbf{C}_{i+1}) \cup IE(\mathbf{P}_{i+1})$,we have

$$\mathbf{D}_{i+1} = 2^{\mathbf{D}_i} \cup 2^{\mathbf{D}_i \times \mathbf{D}_i}.$$

We can see that the above interpretation of RDFS(FA) is very similar to that of DLs, except that not only sets of objects and pairs of objects are considered, but also sets of sets of objects and sets of sets of pairs of objects etc. Figure 1 illustrates the

---

[6] The complete semantics of RDFS(FA) is available at `http://DL-Web.man.ac.uk/rdfsfa/semantics.htm`

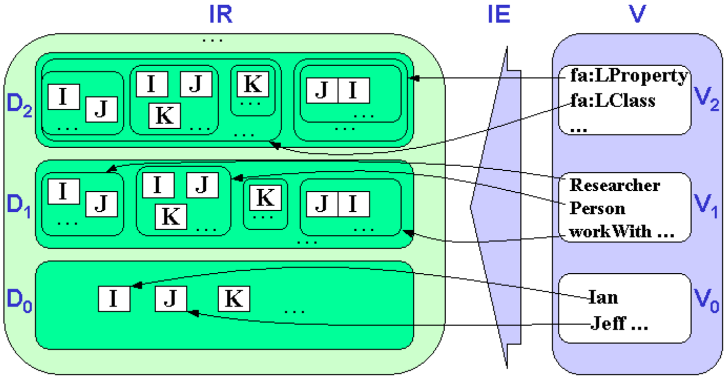

Figure 1: Interpretation of RDFS(FA)

interpretation of RDFS(FA). Vocabularies in stratum 0 (the Instance Layer), e.g. Ian and Jeff, are interpreted as objects (i.e., elements of D_0). Vocabularies for ontology classes (in V_1), such as eg:Researcher and eg:Person, are interpreted as sets of objects. Vocabularies for ontology properties (in V_1), such as eg:workWith, are interpreted as sets of pairs of objects. In stratum 2 (the Language Layer), fa:LClass is interpreted as a set of sets of objects (a set of ontology classes), and fa:LProperty is interpreted as a set of sets of pairs of objects (a set of ontology properties).

There are no dual roles in RDFS(FA). E.g., rdfs:Resource and rdfs:Class are stratified into different layers in RDFS(FA), such that fa:OResource is an instance of fa:LClass, and fa:LClass is a sub-class of fa:LResource, while fa:LResource is an instance of fa:MClass.

4 RDF MT

Another way to clear up the kinds of confusion of RDFS is RDF Model Theory (RDF MT) [11], which gives a precise semantic theory for RDF and RDFS. It is a W3C working draft when this paper is being written.

An interpretation in the RDF model theory is a triple $\langle \mathbf{IR}, IEXT, IS \rangle$, where $\mathbf{IR}$ is the domain (of resources); IS is a function that maps URI references to resources (including classes and properties) in $\mathbf{IR}$, and $IEXT$ is an extension function from $\mathbf{IR}$ to $\mathbf{IR} \times \mathbf{IR}$.

In RDF MT, meaning is given to properties by first mapping the property URI

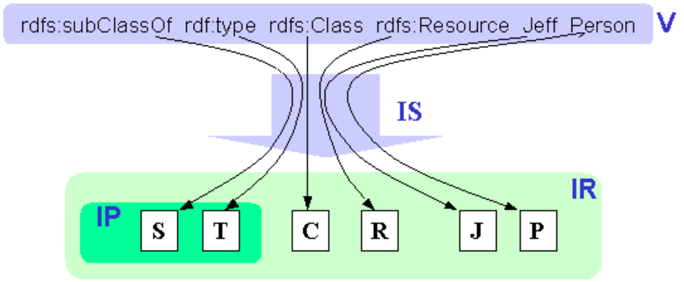

Figure 2: Resources in RDF MT

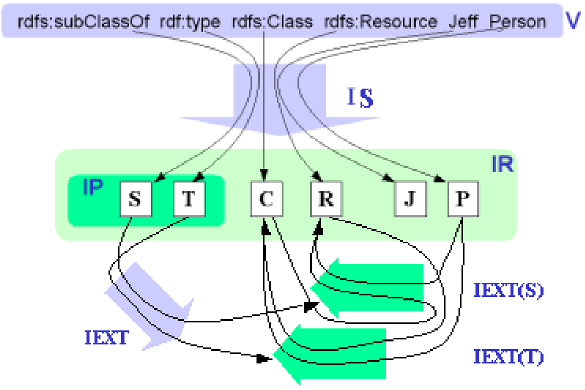

Figure 3: Interpretation of RDF MT

references to an object (resource) of the domain of discourse via IS. E.g., in Figure 2, IS maps rdfs:subClassOf to object S, or IS(rdfs:subClassOf) , rdf:type to object T, or IS(rdf:type), rdfs:Class to object C, or IS(rdfs:Class) etc. Note that **IP** is a special sub-set of **IR**. It is a set of all property objects.

The domain object is then mapped via $IEXT$, the extension function, into their extensions, a set of pairs. Property objects are special in the sense that they can have non-empty extensions. E.g. in Figure 3, $IEXT$ maps S to $IEXT$(S), which is a set of pairs $\{\langle$P,R$\rangle$, $\langle$C,R$\rangle\}$. $IEXT$ maps T to $IEXT$(T), which is a set of pairs $\{\langle$P,C$\rangle$, $\langle$R,C$\rangle\}$.

Class primitives are not fundamental primitives in RDF MT. The class extension $ICEXT$ is defined through the extension of IS(rdf:type):

$$ICEXT(x) = \{y \mid \langle y, x\rangle \text{ is in } IEXT(IS(\text{rdf:type}))\}$$

In Figure 3, $IEXT$(T)=$\{\langle$P,C$\rangle$, $\langle$R,C$\rangle\}$, so P and R are in ICEXT(C).

RDF MT justifies dual roles in RDFS by treating classes and properties as objects. rdfs:Class and rdfs:Resource are mapped to objects C and R in the domain of resource by IS, therefore rdfs:Class is rdfs:subClassOf rdfs:Resource means the pair of R and C is in the extension of the rdfs:subClassOf object S

$$\langle \text{C,R}\rangle \in IEXT(\text{S})$$

while rdfs:Resource is instance of rdfs:Class means the pair of R and C is in the extension of the rdf:type object T

$$\langle \text{R,C}\rangle \in IEXT(\text{T}).$$

According to the definition of $ICEXT$, we have

$$\text{R} \in ICEXT(\text{C}).$$

In this way, the situation between rdfs:Class and rdfs:Resource is given a well defined meaning.

5 OWL and RDFS

The OWL Web Ontology Language provides three increasingly expressive sub-languages: OWL Lite, OWL DL and OWL Full. OWL DL is so named due to its correspondence with an expressive description logics $\mathcal{SHOIQ}(\mathbf{D})$. OWL Lite and OWL DL are actually quite similar and are both decidable, while OWL Full is not decidable, but meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. OWL Full, e.g., supports treating classes as individuals.

There are two formal semantics [14] for OWL. The first one is a direct, standard model-theoretic semantics for OWL DL ontologies. The second one is a RDF-compatible semantics for OWL ontologies. Two versions of this second semantics are provided, one for OWL DL and the other for OWL Full.

From the perspective of RDFS(FA), OWL DL introduces new modelling primitives in stratum 1 and 2, its direct semantics naturally extends the semantics of RDFS(FA) to give meaning to the new primitives, and it preserves the interpretation of RDFS(FA) primitives. E.g., owl:Thing is equivalent to fa:ObjectResource[7].

From the perspective of RDF MT, there are at least three known problems when layering First Order Logics (FOLs) on top of it:

- too few entailments [12]: since classes are objects, it should be guaranteed that all the expected class objects exist in the universe;
- contradiction classes [12, 13]: it should also be guaranteed that no contradiction class objects exist in the universe[8];
- size of the universe [7]: since classes and properties are also objects, problems arise (in certain situations) when one restricts the number of the objects in the universe.

In order to make the RDF-compatible semantics for OWL DL be equivalent to the direct semantics, the domain of discourse is divided into several disjoint parts. In particular, the interpretations of classes, properties, individuals and OWL/RDF vocabulary are strictly separated. Given such a separation, there is a direct correspondence between RDF MT models and standard first-order models.[9]

As far as the RDF-compatible semantics for OWL Full is concerned, the above disjointness restriction is not required. However, it has yet to be proved that this semantics give a coherent meaning to OWL Full. Firstly, it has yet to be proved the RDF-compatible semantics for OWL Full has overcome the first two problems mentioned above. Note that proofs presented in [14] concern only ontologies with the above disjointness restriction. Secondly, the size of universe problem has shown that the interpretation of OWL Full has different features from the interpretation of standard FOL model theoretic semantics. Furthermore, no evidence has been provided to show that no new problems will arise in the future.

6 Discussion

As we have seen, dual roles in the RDFS specification can be confusing and difficult to understand and, more importantly, the specification of its semantics requires a non-

---

[7] The specification of RDFS(FA) is available at `http://DL-Web.man.ac.uk/rdfsfa/specification.htm`

[8] A contradiction class is a class whose membership is impossible to be determined, see [12, 13].

[9] Note that classes and properties thus can't be treated as ordinary objects.

standard model theory, RDF MT. This leads to semantic problems when trying to layer DL and other conventional first-order Web ontology languages on top of RDFS.

In Section 5, we have examined how RDF MT relates to the semantics of OWL, a newly developed Web ontology language by W3C. The RDF MT compatible semantics for OWL DL is equivalent to the direct semantics of OWL DL *only* when a certain domain disjointness restriction is satisfied. Moreover, it has yet to be proved that the RDF MT compatible semantics for OWL Full gives a coherent meaning to OWL Full.

Therefore, we suggest layering Web ontology languages on top of RDFS(FA), a DL-ised sub-language of RDFS, which is based on a (relatively) standard model-theoretic semantics. The direct semantics of OWL DL is consistent with the semantics of RDFS(FA).

This solution could make it possible to have another sub-language of OWL, OWL FA, which allows meta-classes and meta-properties to be used in ontologies. Note that, strictly speaking, OWL Full doesn't support meta-classes. Because the class hierarchies in RDFS (therefore OWL Full) are not trees - any class can be a sub-class, a type or an instance of any class (including itself) simultaneously. While in many Object Oriented systems, classes are sets of objects and meta-classes are the types of classes with certain templates. Classes can never be meta-classes and vice versa.

Such a solution would raise new research issues, such as whether OWL FA is decidable and how to provide DL reasoning services for it.

7 Acknowledgments

We would like to thank Peter Patel-Schneider for helpful discussion on the stratification of RDFS(FA).

References

[1] F. Baader, D. L. McGuiness, D. Nardi, and P. Patel-Schneider, editors. *Description Logic Handbook: Theory, implementation and applications.* Cambridge University Press, 2002.

[2] Tim Berners-lee. Semantic Web Road Map. W3C Design Issues. `URL http://www.w3.org/DesignIssues/Semantic.html`, Oct. 1998.

[3] Dan Brickley and R.V. Guha. Resource Description Framework (RDF) Schema Specification 1.0. W3C Recommentdation, Mar. 2000.

[4] J. Broekstra, M. Klein, S. Decker, D. Fensel, F. van Harmelen, and I. Horrocks. Enabling knowledge representation on the web by extending rdf schema. Hong Kong, May 2001.

[5] Patrick Hayes. RDF Model Theory. Apr 2002. W3C Working Draft, URL `http://www.w3.org/TR/rdf-mt/`.

[6] Ian Horrocks and Peter F. Patel-Schneider. The generation of DAML+OIL. In *Proc. of the 2001 Description Logic Workshop (DL 2001)*, pages 30–35. CEUR Electronic Workshop Proceedings, `http://ceur-ws.org/Vol-49/`, 2001.

[7] Ian Horrocks and Peter F. Patel-Schneider. Three theses of representation in the semantic web. In *Proc. of the Twelfth International World Wide Web Conference (WWW 2003)*, pages 39–47, 2003.

[8] Ora Lassila and Ralph R.Swick. Resource Description Framework (RDF) Model and Syntax Specification – W3C Recommendation 22 February 1999. Technical report, World Wide Web Consortium, 1999.

[9] W. Nejdl, M. Wolpers, and C. Capella. The RDF Schema Specification Revisited. In *Modelle und Modellierungssprachen in Informatik und Wirtschaftsinformatik, Modellierung 2000*, Apr. 2000.

[10] J. Pan and I. Horrocks. Metamodeling Architecture of Web Ontology Languages. In Isabel Cruz, Stefan Decker, Jérôme Euzenat, and Deborah McGuinness, editors, *The Emerging Semantic Web*, Frontiers in artificial intelligence and applications. IOS press, Amsterdam (NL), 2002.

[11] Jeff Z. Pan and Ian Horrocks. Metamodeling Architecture of Web Ontology Languages. In *Proceeding of the Semantic Web Working Symposium (SWWS)*, July 2001. URL `http://www.cs.man.ac.uk/~panz/Zhilin/download/Paper/Pan-Horrocks-rdfsfa-2001.pdf`.

[12] Peter F. Patel-Schneider. Layering the Semantic Web: Problems and Directions. . In *2002 International Semantic Web Conference*, Jun 2002.

[13] Peter F. Patel-Schneider. Two Proposals for a Semantic Web Ontology Language. In *2002 International Description Logic Workshop*, Apr 2002.

[14] Peter F. Patel-Schneider, Patrick Hayes, and Ian Horrocks. OWL Web Ontology Language Semantics and Abstract Syntax. Technical report, Mar. 2003. W3C Working Draft.

[15] Frank van Harmelen, Peter F. Patel-Schneider, and Ian Horrocks. A Model-Theortis Semantics of DAML+OIL(March 2001). Mar. 2001.