

# Logical reconstruction of normative RDF

Jos de Bruijn<sup>1</sup>, Enrico Franconi<sup>2</sup>, and Sergio Tessaris<sup>2</sup>

<sup>1</sup> Digital Enterprise Research Institute,  
University of Innsbruck, Austria  
`jos.debruijn@deri.org`

<sup>2</sup> Faculty of Computer Science,  
Free University of Bozen-Bolzano, Italy  
`lastname@inf.unibz.it`

**Abstract.** In this sketchy paper we introduce a logical reconstruction of the RDF family of languages and the OWL-DL family of languages. We prove that our logical framework is equivalent to the normative W3C definitions of RDF/RDFS and OWL-DL/Lite. The main aim is to have a unified model theoretic semantics for both worlds. As a consequence we get various complexity results and a model theoretic semantics for basic SPARQL.

## 1 Introduction

The main aim of this sketchy paper is to recast the RDF and RDFS model theory in a more classical logic framework, and to use this characterisation to shed new light on the ontology languages layering in the semantic web, and to lay down the logic based semantics of SPARQL. In particular, we will show how the models of RDF can be related to the models of DL based ontology languages, without requiring any change on the existing syntactic or semantic definitions in the RDF and OWL-DL realms.

We first introduce the notion of herbrand and canonical models for RDF graphs, and we use this notion to characterise RDF entailment. RDF herbrand models can also be seen as classical first order structures, that we call FO interpretations. These structures provide the semantic bridge between RDF and classical logics, such as description logics (DL) based languages (e.g., OWL-DL). The intuition beyond FO interpretations is that it singles out the concepts and the individuals from an RDF herbrand model – possibly in a polymorphic way when the same node is given both the meaning as a class and as an individual.

Once we have characterised RDF graphs in terms of their herbrand models, it is possible to understand the notion of logical implication between RDF graphs and classical logic formulas. At the end of this paper we analyse the problem

---

This work has been partially supported by the EU projects KnowledgeWeb, Interop, Tones, Sekt, and Asg. This paper extends a paper previously published at the PPSWR-05 workshop [de Bruijn *et al.*, 2005].

of querying RDF graphs with OWL-DL ontologies. We prove an important reduction result. That is, given an RDF graph  $\mathcal{S}$  and a query  $Q$ , the answer set of  $Q$  to  $\mathcal{S}$  (as defined by W3C) is the same as the answer of  $Q$  to  $\mathcal{S}$  given the empty KB. This shows a complete interoperability between RDF and OWL-DL. For example, in absence of ontologies, it would be possible to use OWL-QL to answer queries to RDF graphs, or to use SPARQL to answer queries to ABoxes.

In this paper we assume that the reader is familiar with the definitions associated to RDF.

## 2 RDF Model Theory revisited

In this paper we consider an extended notion of RDF graph, in which we are less restrictive on the kind of triples. In particular we allow

- literals in subject positions;
- blank nodes in property positions.

Note that the first kind of extension has been already considered by W3C working groups (e.g. see Section 2.2 of [Prud'hommeaux and Seaborne, 2005]). All the results shown in this paper still holds for the standard definition of RDF graph. From now on, by *RDF graph* we intend the extended definition. Also note that reification is not considered as not being part of the normative semantic definition of RDF.

We indicate with  $\mathcal{RDF}_U$  the set of all RDF URI references together with the set of all literals in their canonical representation<sup>1</sup>. An RDF graph is said to be *well typed* if doesn't contain the triple

$\langle \text{"xxx"} \text{^^rdf:XMLLiteral, rdf:type, rdf:XMLLiteral} \rangle$

where  $\text{"xxx"} \text{^^rdf:XMLLiteral}$  is an ill-typed XML literal string (see the RDF semantic conditions in Section 3.1 of [Hayes, 2004]).

We first define the notion of herbrand and canonical models for an RDF graph.

**Definition 1.** (*Herbrand and canonical models*)

*A herbrand model of an RDF graph  $\mathcal{S}$  is a well typed ground instantiation of the graph obtained by replacing each bnode in the completed  $\mathcal{S}$  with some element in  $\mathcal{RDF}_U$ .*

*A graph is completed if it is augmented by the RDF and RDFS axiomatic triples, it is extended by applying the RDF and RDFS entailment rules (see sections 3.1, 4.1, 7.2 and 7.3 in [Hayes, 2004])<sup>2</sup> and all the literals are in their canonical representation.*

<sup>1</sup> The canonical representation of a literal is a chosen representative of all the literals associated to the same value, if the literal is non ill-typed, otherwise it is the literal itself.

<sup>2</sup> Note that, since we allow literals as subject in RDF triples, we need to add a dual rule to RDF2 and to RDFS1 acting on literals in the subject of a triple; moreover, rules RDF2 and RDFS1 should have the proper literal instead of a bnode in the "then add" part.

The canonical model  $\widehat{\mathcal{S}}$  of an RDF graph  $\mathcal{S}$  is the herbrand model of  $\mathcal{S}$  obtained by skolemisation, i.e. by replacing each distinct bnode in  $\mathcal{S}$  with a distinct fresh URI – that is, a skolem constant not appearing elsewhere in  $\mathcal{S}$  nor in the context in which  $\mathcal{S}$  is used (e.g. in queries).

Note that a herbrand model is always finite if the RDF graph is finite, that a ground RDF graph has a unique herbrand model that it is also its canonical model, and that a herbrand model is a ground RDF graph.

As the following theorem shows, the herbrand models of an RDF graph contain *explicitly* all the information entailed by the graph itself.

**Theorem 2.** (*RDF entailment*)

An RDF graph  $\mathcal{S}$  entails an RDF graph  $\mathcal{E}$  (as defined in [Hayes, 2004]), written  $\mathcal{S} \vdash \mathcal{E}$ , if and only if some herbrand model of  $\mathcal{E}$  is a subgraph of the canonical model of  $\mathcal{S}$ .

**Corollary 3.** (*Complexity of entailment*)

1. RDF entailment is NP-complete in the size of the RDF graphs.
2. RDF entailment is polynomial in the size of the entailing graph  $\mathcal{S}$ .
3. RDF entailment is polynomial in the size of the graphs if  $\mathcal{E}$  is acyclic or ground.

The proofs are based on a reduction to the problem of conjunctive query containment, and by using the interpolation lemma in [Hayes, 2004].

The above theorem and corollary (without the polynomiality results) have been already sketched in [Gutierrez *et al.*, 2004]. However, the results in [Gutierrez *et al.*, 2004] are imprecise since the role of axiomatic triples and the completion (as defined here) are neglected, and literals are not taken in careful account.

## 2.1 The Semantics of Basic SPARQL

Let's now consider SPARQL queries on RDF graphs. If we restrict our attention to SPARQL query basic graph patterns [Prud'hommeaux and Seaborne, 2005], we can define the semantics of query answering in the usual logic based way (as, e.g., is defined for classical relational databases, or for description logics). We also disallow in this paper the answer to a query to contain blank nodes. Relaxing this restriction raises several issues regarding the redundancy of answers, which are not taken into account in [Prud'hommeaux and Seaborne, 2005].

**Definition 4.** (*Semantics of basic SPARQL*)

A SPARQL query basic graph pattern to an RDF graph  $\mathcal{S}$  is a (possibly ground) RDF graph  $\mathcal{Q}_{\mathbf{x}}$  where, in addition to URIs and bnodes, variables are allowed; the elements in the set  $\mathbf{x}$  (possibly empty) of  $n$  variables of a query are called distinguished variables, and the bnodes play the role of non-distinguished variables. The answer set of  $\mathcal{Q}_{\mathbf{x}}$  is the set of all substitutions of the distinguished variables with some arbitrary URI from  $\mathcal{RDF}_U$ , such that the for each substitution the instantiated query is entailed by  $\mathcal{S}$ , i.e.,

$$\{\langle c_1 \dots c_n \rangle \in (\mathcal{RDF}_U)^n \mid \mathcal{S} \vdash \mathcal{Q}_{[x_1 \mapsto c_1, \dots, x_n \mapsto c_n]}\}.$$

Note that according to our extended definition of RDF graphs we allow blank nodes and variables in property position.

The complexity results presented in Corollary 3 can be rephrased in the context of SPARQL. We consider the graph to be queried as the *data* against which the given query is verified; so, in this way, we can define the data complexity of the problem of query answering.

**Corollary 5.** (*Complexity of SPARQL*)

*Query answering for SPARQL query basic graph patterns is polynomial in data complexity.*

Note that the above definition together with the correspondence stated in Theorem 2 supports the idea of implementing SPARQL by means of a relational DBMS. In fact, the theorem suggests a straightforward query answering technique based on canonical models.

The same technique can be extended in order to provide blank nodes in the query answers. However, it can be shown that guaranteeing non redundant answers increase the complexity of the problem, which rises to be NP-complete in the size of the answer set (i.e. data complexity).

The result of Corollary 5 has been already sketched in [Gutierrez *et al.*, 2004] for a richer query language, with the same imprecision we mentioned before.

## 2.2 The FO Model Theory for RDF

A *FO interpretation* (first order interpretation) of an RDF graph shows how models of RDF can be seen as interpretations of classical first order logic.

**Definition 6.** (*FO interpretation of an RDF herbrand model*)

A FO interpretation  $\mathcal{I}(\cdot)$  of an RDF herbrand model  $\mathcal{I}_{RDF}$  is a first order type structure  $\mathcal{I}(\mathcal{I}_{RDF}) = \langle \Delta, \cdot^{\mathcal{I}_O}, \cdot^{\mathcal{I}_C}, \cdot^{\mathcal{I}_R} \rangle$ , where  $\Delta$  is a non-empty abstract domain corresponding to  $\mathcal{RDF}_U$ . An RDF herbrand model  $\mathcal{I}_{RDF}$  with RDFS vocabulary and containing an XML clash has no FO interpretation. The interpretation of the elements of  $\mathcal{I}_{RDF}$  is given by the interpretation functions  $\cdot^{\mathcal{I}_O}$ ,  $\cdot^{\mathcal{I}_C}$ ,  $\cdot^{\mathcal{I}_R}$ , whose domain is  $\mathcal{RDF}_U$ , and the range is respectively all elements of  $\Delta$ , all subsets of  $\Delta$ , and all binary relations over  $\Delta$ . The interpretation functions state which of the elements of the graph play the role of individuals, concepts, and roles.

For each  $u \in \mathcal{RDF}_U$ ,  $\mathcal{I}(\mathcal{I}_{RDF})$  should be such that:

$$\begin{aligned} u^{\mathcal{I}_O} &= u \\ u^{\mathcal{I}_C} &= \{o \mid \langle o, \mathit{rdf:type}, u \rangle \in \mathcal{I}_{RDF}\} \\ u^{\mathcal{I}_R} &= \{(o_1, o_2) \mid \langle o_1, u, o_2 \rangle \in \mathcal{I}_{RDF}\} \end{aligned}$$

An URI reference is associated to more than one syntactic type, e.g., an URI may refer to an individual and to a class at the same time: polymorphic meanings of

URIs are allowed. However note that, just like in the case of contextual predicate calculus (as defined in [Chen *et al.*, 1993]) and of  $\pi$ -semantics of [Hustadt *et al.*, 2005], in the above definition there is no semantic interaction between the distinct occurrences of the same URI as a concept name, or as a role name, or as an individual. This absence of interaction is required for classical first order (description) logic fragments such as OWL-Lite or OWL-DL. For example, given the triple  $\langle \text{ex:o}, \text{rdf:type}, \text{ex:o} \rangle$  within an RDF herbrand model, in the FO interpretation associated to it the URI  $\text{ex:o}$  is interpreted as both an individual and a concept, and the individual  $\text{ex:o}$  is in the extension of the concept  $\text{ex:o}$ .

We say that the FO interpretations of an RDF graph are the FO interpretations of its herbrand models. The main theorem of this Section states that we can correctly define RDF entailment and queries using a classical logic with FO interpretations.

**Theorem 7.** (*FO entailment and query*)

1. An RDF graph  $\mathcal{S}$  entails an RDF graph  $\mathcal{E}$  (as defined in [Hayes, 2004]), written as  $\mathcal{S} \vdash \mathcal{E}$ , if and only if the set of all the FO interpretations of  $\mathcal{S}$  is included in the set of all the FO interpretations of  $\mathcal{E}$ , written  $\mathcal{S} \models \mathcal{E}$ .
2. The answer set of a SPARQL query basic graph pattern  $\mathcal{Q}_x$  to an RDF graph  $\mathcal{S}$ , as defined in Definition 4, is equal to

$$\{\langle c_1 \dots c_n \rangle \in (\mathcal{RDF}_U)^n \mid \mathcal{S} \models \mathcal{Q}_{[x_1 \mapsto c_1, \dots, x_n \mapsto c_n]}\}.$$

### 3 Classical Logic Interoperability

In this Section we define the interoperability between RDF graphs and first order classical logics. We show how a tower of classical logics (e.g., from OWL-Lite to OWL-DL, to full first order logic, or any arbitrary logic equipped with classical first order models) can be built on top of the language of RDF: the interoperability is grounded on the notion of FO interpretations.

First, we need to define the notion of non high order graphs, that basically do not have bnodes in any property or class position.

**Definition 8.** (*Non-high order RDF graph*)

An RDF graph is non-high order if bnodes and variables are not in property position of any triple, nor in object position of `rdf:type` triples, nor in any triple with `rdfs:subClassOf`, `rdfs:subPropertyOf`, `rdfs:domain`, `rdfs:range` predicates.

Note that herbrand models and canonical models are always non-high order RDF graphs, since they are always ground graphs.

Given a classical first order logic  $\mathcal{C}$ , we define now the translation of a non-high order graph into some formula of  $\mathcal{C}$ . We require that in  $\mathcal{C}$  the interpretation of well-typed literals is subject to the Unique Name Assumption.

**Definition 9.** (Classical logic translation)

The classical logic translation  $\text{FO}(\mathcal{S})$  of a non-high order well-typed RDF graph  $\mathcal{S}$  is a predicate logic formula, where URIs and literals (in their canonical representation) are constants and blank nodes are existentially quantified variables, and the body is a conjunction of the ground binary atomic formulas in correspondence with the triples of  $\mathcal{S}$ , where to each binary atomic formula of the kind “ $\text{rdf:type}(a, b)$ ” a ground unary atomic formula of the kind “ $b(a)$ ” is added, to each binary atomic formula of the kind “ $\text{rdfs:subClassOf}(a, b)$ ” a formula of the kind “ $\forall x. a(x) \rightarrow b(x)$ ” is added, to each binary atomic formula of the kind “ $\text{rdfs:subPropertyOf}(a, b)$ ” a formula of the kind “ $\forall xy. a(x, y) \rightarrow b(x, y)$ ” is added, to each binary atomic formula of the kind “ $\text{rdfs:domain}(a, b)$ ” a formula of the kind “ $\forall xy. a(x, y) \rightarrow b(x)$ ” is added, to each binary atomic formula of the kind “ $\text{rdfs:range}(a, b)$ ” a formula of the kind “ $\forall xy. a(x, y) \rightarrow b(y)$ ” is added. If  $\mathcal{S}$  has RDFS vocabulary and contains an XML clash, then  $\text{FO}(\mathcal{S})$  is equal to  $\perp$ .

We now introduce the general problem of reasoning and query answering in a classical first order logic  $\mathcal{C}$  given an RDF graph.

**Definition 10.** (Classical logic RDF extension)

1. The logical implication problem in a classical logic  $\mathcal{C}$  given an RDF graph  $\mathcal{S}$  is defined as follows:

$$\text{FO}(\widehat{\mathcal{S}}), \phi \models_{\mathcal{C}} \psi$$

where  $\phi$  and  $\psi$  are formulas in  $\mathcal{C}$ ,  $\phi$  does not contain any symbol from the RDF and RDFS vocabularies, and  $\models_{\mathcal{C}}$  is entailment in  $\mathcal{C}$ .

2. The query answering problem in a classical logic  $\mathcal{C}$  given an RDF graph  $\mathcal{S}$  is defined as follows:

$$\{\langle c_1 \dots c_n \rangle \in (\mathcal{RDF}_U)^n \mid \text{FO}(\widehat{\mathcal{S}}), \phi \models_{\mathcal{C}} \psi_{[x_1 \mapsto c_1, \dots, x_n \mapsto c_n]}\}.$$

where  $\phi$  is a formula in  $\mathcal{C}$  which does not contain any symbol from the RDF and RDFS vocabularies, and  $\psi_{\mathbf{x}}$  is an open formula in  $\mathcal{C}$  (expressing the query) with  $\mathbf{x}$  being the free (distinguished) variables, and  $\models_{\mathcal{C}}$  is entailment in  $\mathcal{C}$ .

The above general definition of reasoning and querying given an RDF graph is actually an abstraction of basic reasoning and querying for RDF graphs only, as the following reduction theorem shows.

**Theorem 11.** (Reduction theorem)

1. Given an RDF graph  $\mathcal{S}$  and a non-high order graph  $\mathcal{E}$ ,  $\mathcal{S} \vdash \mathcal{E}$  if and only if  $\text{FO}(\widehat{\mathcal{S}}) \models_{\mathcal{C}} \text{FO}(\mathcal{E})$
2. Given an RDF graph  $\mathcal{S}$  and a SPARQL non-high order query basic graph pattern  $\mathcal{Q}_{\mathbf{x}}$ , its answer set is equal to

$$\{\langle c_1 \dots c_n \rangle \in (\mathcal{RDF}_U)^n \mid \text{FO}(\widehat{\mathcal{S}}) \models_{\mathcal{C}} \text{FO}(\mathcal{Q}_{[x_1 \mapsto c_1, \dots, x_n \mapsto c_n]})\}.$$

The proof of the reduction theorem is based on the following lemma.

**Lemma 12.** (*Canonical entailment*)

An RDF graph  $\mathcal{S}$  entails an RDF graph  $\mathcal{E}$  (as defined in [Hayes, 2004]), i.e.,  $\mathcal{S} \vdash \mathcal{E}$ , if and only if the FO interpretation corresponding to the canonical model of  $\mathcal{S}$  is in the set of all the FO interpretations of  $\mathcal{E}$ .

This lemma together with the reduction theorem justifies the use of datalog-like implementations for SPARQL.

We believe that the classical logic RDF extension presented here is a meaningful way to build up logical languages on top of RDF, and it is a formal justification of the semantic web tower of languages proposed by Tim Berners-Lee. As a matter of fact, in our proposed framework it is possible to add (first-order) logic-based knowledge on top of RDF graphs, written as a knowledge base  $\phi$  in the logic  $\mathcal{C}$ . Note that such knowledge base  $\phi$  should not contain any RDF and RDFS vocabularies; this restriction is not really limiting, since – as it is evident by looking at the FO translation of RDF graphs – it is possible to write directly in  $\mathcal{C}$  itself the RDF/RDFS properties.

Also note that any use of the RDF and RDFS vocabularies in the entailed formula  $\psi$  (or in the query) is affected *only* by the RDF graph and not by the knowledge base expressed by  $\phi$ . This observation suggests the definition of a *pure* classical logic RDF extension by considering the classical logic translation of the simple skolemisation (and not of the canonical model) of  $\mathcal{S}$ , and by restricting both  $\phi$  and  $\psi$  to contain no symbol from the RDF and RDFS vocabularies. The encoding of the RDF graph in the logic  $\mathcal{C}$  would be obtained by only skolemising the bnodes in  $\mathcal{S}$  without completing the graph itself, i.e. without adding the axiomatic triples and without applying the entailment rules.

**Definition 13.** (*Pure classical logic RDF extension*)

1. The pure logical implication problem in a classical logic  $\mathcal{C}$  given an RDF graph  $\mathcal{S}$  is defined as follows:

$$\text{FO}(\text{SK}(\mathcal{S})), \phi \models_{\mathcal{C}} \psi$$

where  $\phi$  and  $\psi$  are formulas in  $\mathcal{C}$  which do not contain any symbol from the RDF and RDFS vocabularies, and  $\models_{\mathcal{C}}$  is entailment in  $\mathcal{C}$ .

2. The query answering problem in a classical logic  $\mathcal{C}$  given an RDF graph  $\mathcal{S}$  is defined as follows:

$$\{(c_1 \dots c_n) \in (\mathcal{RDF}_{\bar{U}}^-)^n \mid \text{FO}(\text{SK}(\mathcal{S})), \phi \models_{\mathcal{C}} \psi_{[x_1 \mapsto c_1, \dots, x_n \mapsto c_n]}\}.$$

where  $\mathcal{RDF}_{\bar{U}}^-$  does not include the RDF and RDFS vocabularies,  $\phi$  is a formula in  $\mathcal{C}$  which does not contain any symbol from the RDF and RDFS vocabularies, and  $\psi_{\mathbf{x}}$  is an open formula in  $\mathcal{C}$  (expressing the query) with  $\mathbf{x}$  being the free (distinguished) variables and not containing any symbol from the RDF and RDFS vocabularies, and  $\models_{\mathcal{C}}$  is entailment in  $\mathcal{C}$ .

As the following lemma shows, the classical logic translation of an RDF graph seems to be mostly insensitive to the computation of the canonical model.

**Lemma 14.** (*Minimal models of classical logic translations*)

Given an RDF graph  $\mathcal{S}$ ,

1.  $\text{FO}(\mathbb{SK}(\mathcal{S}))$ , i.e. the classical logic translation of the skolemisation of  $\mathcal{S}$ , has a unique minimal model;
2.  $\text{FO}(\widehat{\mathcal{S}})$ , i.e. the classical logic translation of the canonical model of  $\mathcal{S}$ , has a unique minimal model;
3. the minimal models of  $\text{FO}(\mathbb{SK}(\mathcal{S}))$  and  $\text{FO}(\widehat{\mathcal{S}})$  are isomorphic up to renaming of the skolem constants, and by not considering the parts involving the RDF and RDFS vocabularies.

The following important equivalence result can now be proved.

**Theorem 15.** (*Equivalence theorem*)

The pure classical logic RDF extension and the classical logic RDF extension are equivalent if  $\psi$  does not contain any symbol from the RDF and RDFS vocabularies, and the query answers containing symbols from the RDF and RDFS vocabularies are discarded.

## 4 Interoperability between RDF and OWL-DL

The results presented so far have several immediate consequences when considering the interoperability between OWL-DL/Lite with RDF.

Note that in Section 3 we assume that the theory written in logic  $\mathcal{C}$ , which interoperates with  $\text{RDF}(\mathcal{S})$ , is not encoded in the graph itself (see Definition 10). For this reason, we do not run into the problems of dealing with triples which correspond to the peculiar serialisation of OWL into RDF. However, nothing prevents any specific implementation to use RDF to represent syntactically the formulae of the logic  $\mathcal{C}$ .

First of all, it is possible to have an implementation for free of a query evaluation engine for SPARQL non-high order query basic graph patterns over RDF graphs, using any of the existing description logics based query system available. In fact, it is enough to encode the (arbitrary) RDF graph to query as an ABox in the system (by considering its canonical model), and to query it by encoding the SPARQL non-high order query basic graph pattern as a standard conjunctive query. The reduction theorem above guarantees that we will get the correct answer.

Moreover, it is possible to extend the query problem of an RDF graph in SPARQL to the query problem of an RDF graph (possibly with RDFS vocabulary) given an ontology in OWL-DL, again by exploiting standard description logics based query systems. This is achieved by just adding the encoding of the RDF graph to query as an OWL-DL knowledge base corresponding to the FO translation of its canonical model or of its simple skolemisation (depending whether the pure classical logic RDF extension has been chosen or not).

This work also shows how it is possible to give a semantics to OWL-DL based on RDF, generalising the recommended semantics given by [Patel-Schneider *et al.*, 2004]. Our proposal is fully compatible with the W3C recommended semantics, but removes some of the non necessary limitations related to the polymorphism of URIs, enforced in [Patel-Schneider *et al.*, 2004] by the “vocabulary partitioning” and “explicit typing” restrictions. As a matter of fact, [Patel-Schneider *et al.*, 2004] would allow interoperation and queries only with RDF graphs *not* containing any meta information (for example, of the kind represented by the triple  $\langle \text{ex:o}, \text{rdf:type}, \text{ex:o} \rangle$ ); a slightly more liberal restriction has been proposed in [Pan and Horrocks, 2003] where acyclic regularly stratified meta-classes are allowed but still disallowing the above cyclic example.

## References

- [Chen *et al.*, 1993] Weidong Chen, Michael Kifer, and David Warren. HILOG: a foundation for higher-order logic programming. *Journal of Logic Programming*, 15(3):187–230, February 1993.
- [de Bruijn *et al.*, 2005] Jos de Bruijn, Sergio Tessaris, and Enrico Franconi. Logical reconstruction of RDF and ontology languages. In *Proc. of PPSWR-05*, 2005.
- [Gutierrez *et al.*, 2004] C. Gutierrez, C. Hurtado, and A. Mendelzon. Foundations of semantic web databases. In *Proceedings of the 2004 ACM SIGART SIGMOD SIGART Symposium on Principles of Database Systems (PODS'04)*, 2004.
- [Hayes, 2004] Patrick Hayes. RDF semantics. Technical report, W3C, February 2004. W3C recommendation, URL <http://www.w3.org/TR/rdf-mt/>.
- [Hustadt *et al.*, 2005] U. Hustadt, Boris Motik, and U. Sattler. Reasoning for description logics around  $\mathcal{SHIQ}$  in a resolution framework. Technical Report 3-8-04/04, FZI, Karlsruhe, Germany, 2005. URL <http://www.fzi.de/ipe/publikationen.php?id=1172>.
- [Pan and Horrocks, 2003] Jeff Pan and Ian Horrocks. RDFS(FA) and RDF MT: Two semantics for RDFS. In *Proc. 2003 International Semantic Web Conference (ISWC 2003)*, 2003.
- [Patel-Schneider *et al.*, 2004] Peter F. Patel-Schneider, Patrick Hayes, and Ian Horrocks. OWL web ontology language semantics and abstract syntax. Technical report, W3C, February 2004. W3C recommendation, URL <http://www.w3.org/TR/owl-semantics/>.
- [Prud'hommeaux and Seaborne, 2005] Eric Prud'hommeaux and Andy Seaborne. SPARQL query language for RDF. Technical report, W3C, July 2005. W3C working draft, URL <http://www.w3.org/TR/rdf-sparql-query/>.