The Logic of Extensional RDFS

Enrico Franconi[1], Claudio Gutierrez[2], Alessandro Mosca[1],
Giuseppe Pirrò[1], Riccardo Rosati[3]

[1] KRDB, Free University of Bozen-Bolzano, Bolzano, Italy
[2] Department of Computer Science, University of Chile, Santiago, Chile
[3] University of Rome La Sapienza, Rome, Italy

Abstract. The normative version of RDF Schema (RDFS) gives non-standard (intensional) interpretations to some standard notions such as classes and properties, thus departing from standard set-based semantics. In this paper we develop a standard set-based (extensional) semantics for the RDFS vocabulary while preserving the simplicity and computational complexity of deduction of the intensional version. This result can positively impact current implementations, as reasoning in RDFS can be implemented following common set-based intuitions and be compatible with OWL extensions.

1 Introduction

The Resource Description Framework (RDF) [9] is the standard data model for publishing and interlinking data on the Web. Its associated vocabulary RDF Schema (RDFS) (classes, properties, hierarchies) gives non-standard (intensional) interpretations to some standard set theoretical notions such as classes and properties. This brings some difficulties to the reasoning systems based on classical first-order logic (FOL). RDF enables the making of *statements* about (Web) resources in the form of triples including a *subject*, a *predicate* and an *object* expressed in manifold vocabularies. Efforts like the Linked Open Data project [8] give a glimpse of the magnitude of RDF data today available.

In many application scenarios, there is the need to have on top of RDF data a language to structure knowledge domains. To cope with this aspect, the Web Consortium developed standard vocabularies such as RDF Schema (RDFS) and OWL. RDFS was designed with a minimalist philosophy and it includes essentially the machinery for expressing subclass, subproperty, type and such. On the other hand, OWL is a more expressive language that includes a much richer set of features.

From a standardization point of view the current normative RDFS has two weaknesses. First, the interpretations of basic notions such as subclass and subproperty do not have the usual set-based meaning. For example, in in Fig. 1, even though `:birthCity` is a subproperty `:birthPlace`, one cannot derive the fact that the range of the property `:birthCity` must be `:Place`. Second, the normative semantics of RDFS and OWL differ for some of their common vocabularies. RDFS, for historical reasons, follows an *intensional* semantics while OWL

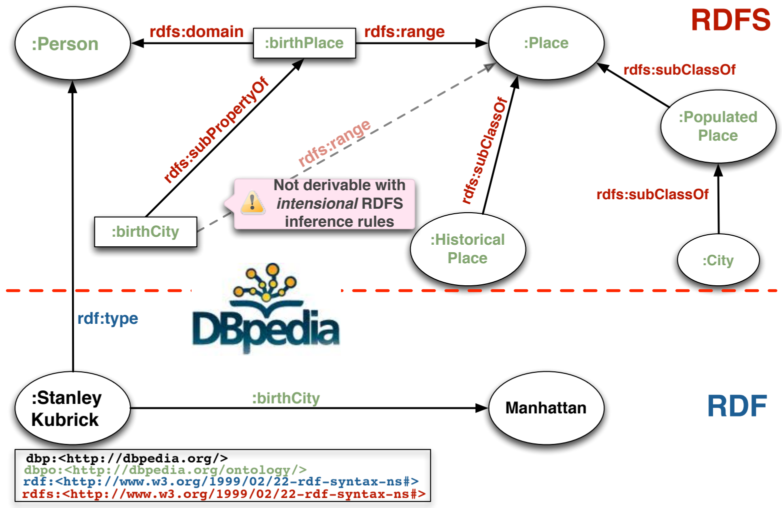

Fig. 1. An RDFS graph taken from `dbpedia.org` showing compatibility problems between OWL and RDFS. The dotted arrow is valid in OWL while not in RDFS.

adopts a standard *extensional* set-based semantics. This intensional semantics of RDFS brings compatibility problems with OWL. In the example considered shown in Fig. 1, the dotted `rdfs:range` property is a valid set-based deduction, thus valid in OWL, while *not* derivable in RDFS.

The designers of RDFS were aware of this problem, and added in a "non-normative" status the standard set-based semantics and some sound inference rules for it. This so-called "extensional" version of RDFS corresponds exactly to the standard set-based interpretation of the vocabulary (and thus is fully compatible with OWL). The rationale for keeping a weaker (intensional) semantics for RDFS was efficiency: "In some ways the extensional versions provide a simpler semantics, but they require more complex inference rules. The 'intensional' semantics [...] provides for most common uses of subclass and subproperty assertions, and allows for simpler implementations of a complete set of RDFS entailment rules." (W3C RDFS Semantics Spec., [7]). According to this specification, RDFS inference engines develop following the intensional semantics.

Thus, two relevant problems regarding the natural extensional RDFS semantics have prevented its usage: *i)* What is the complexity overload associated to the extensional semantics for RDFS?; *ii)* Can normative RDFS inference engines (based on the computation of a completion in a forward-chaining manner) be easily extended to support extensional RDFS, and at which cost?

Contributions. This paper answer both question in the positive. First, we provide a simple sound and complete proof system for the extensional semantics of RDFS. Second, we show that a meaningful completion of the graph computed by using the rules in a forward-chaining manner can still be computed in polynomial case (as for intensional RDFS) thus spurring on current system that use completion. These two results can be seen as founding the ground for the developing of the extensional semantics for the RDFS vocabulary while preserving the simplicity and computational complexity of deduction of the intensional case.

Our results can be considered as an extension of intensional RDFS. Our results address not only an interesting theoretical open problem, but could impact on current implementations (for the most part based on the normative intensional semantics) in a positive sense. Indeed, we show that reasoning in RDFS can follow common set-based intuitions and be compatible with OWL extensions. Moreover, we show that the rule system that we present is easily embeddable in existing libraries such as Jena.

2 Preliminaries

The Resource Description Framework (RDF) [9] and RDF Schema (RDFS) are the W3C's standard data model for the publishing and interlinking of data on the Web. In RDF only simple statements about resources can be expressed via triples: a resource may be an instance of another resource (representing a class typing the instance) and/or a property of another resource. RDFS augments RDF with some minimal *vocabulary*, allowing to express hierarchies of classes and properties and to restrict the domain and range of properties. As an example of an RDFS graph, see Fig. 1. In what follow we will give a simple presentation abstracted from implementation (e.g., namespace) details.

Let $\mathcal{U}$, $\mathcal{L}$, $\mathcal{B}$ three pairwise disjoint sets representing the set of URIs, literals and blank nodes, respectively. For simplicity, we denote unions of these sets by simply concatenating their names.

Definition 1 (RDF triple, graph). *An RDF triple t is a tuple of the form $(s, p, o) \in (\mathcal{UB}) \times \mathcal{U} \times (\mathcal{UBL})$, where s, p, o are called subject, predicate and object, respectively. A triple is ground if it does not contain blank nodes. A (ground) RDF graph $\mathcal{G}$ is a set of RDF (ground) triples. The vocabulary of $\mathcal{G}$, denoted* $\mathrm{voc}(\mathcal{G})$, *is the set of elements in $\mathcal{UBL}$ that occurs in its triples.*

The $\boldsymbol{\rho}$df fragment

In this work we will concentrate on a simple and small fragment of RDFS, which includes only the special RDFS vocabulary `type`, `property`, `subClass`, `subProperty`, `domain` and `range`. This fragment is called **$\boldsymbol{\rho}$df** and was introduced first in [11]. It has been shown to capture the essential semantics of the full fragment, while avoiding to deal with minor idiosyncrasies. In the following, we will denote its vocabulary as $\mathcal{V}_{\rho\mathrm{df}} = \{\mathtt{sc}, \mathtt{sp}, \mathtt{dom}, \mathtt{range}, \mathtt{type}\}$. As it has been shown in [11], ρdf is self-contained as it does not rely on the RDFS vocabulary beyond this subset. ρdf is endowed with a set of inference rules that preserves the original RDFS semantics restricted to this vocabulary [10].

The Intensional (Normative) Semantics

The normative semantics of RDFS [7] is built upon the standard logic notions of model, interpretation and entailment. In the following we rephrase the normative model theory of RDFS using first-order logic (FOL) in the spirit of [4]. The signature of the language includes a ternary predicate T –to represent RDF triples – and two unary predicates C and P that will represent the membership of individuals to "`rdfs:Class`" and "`rdf:Property`", respectively. It can be proved that, given a ρdf graph $\{(s_1, p_1, o_1), \cdots, (s_n, p_n, o_n)\}$, its models according the the normative RDFS model theory in the W3C specification [4] are the same as the models of the FOL formula $\exists \mathbf{b}\ T(s_1, p_1, o_1) \wedge \cdots \wedge T(s_n, p_n, o_n)$, where $\mathbf{b}$ is the set of blank node symbols appearing in the graph, under the FOL theory specified by the axioms listed below.

The basic axioms primitively define `subClass`, `subProperty`, `domain`, `range` in terms of `type` in the obvious way –as in set theory[4]:

$$\forall a, b\ (a, \mathtt{sc}, b) \longrightarrow C(a) \wedge C(b) \wedge \forall x\ (x, \mathtt{type}, a) \rightarrow (x, \mathtt{type}, b) \tag{1}$$

$$\forall a, b\ (a, \mathtt{sp}, b) \longrightarrow P(a) \wedge P(b) \wedge \forall x, y\ (x, a, y) \rightarrow (x, b, y) \tag{2}$$

$$\forall a, c\ (a, \mathtt{dom}, c) \longrightarrow \forall x, y\ (x, a, y) \rightarrow (x, \mathtt{type}, c) \tag{3}$$

$$\forall a, d\ (a, \mathtt{range}, d) \longrightarrow \forall x, y\ (x, a, y) \rightarrow (y, \mathtt{type}, d) \tag{4}$$

To cope with reflexivity and transitivity of the subclass and subproperty relations we have also the following axioms:

$$\forall a, b, c\ (a, \mathtt{sc}, b) \wedge (b, \mathtt{sc}, c) \longrightarrow (a, \mathtt{sc}, c) \tag{5}$$

$$\forall a\ C(a) \longrightarrow (a, \mathtt{sc}, a) \tag{6}$$

$$\forall a, b, c\ (a, \mathtt{sp}, b) \wedge (b, \mathtt{sp}, c) \longrightarrow (a, \mathtt{sp}, c) \tag{7}$$

$$\forall a P(a)\ \longrightarrow (a, \mathtt{sp}, a) \tag{8}$$

The following typing axioms are also needed in normative RDFS:

$$\forall a, b\ (a, \mathtt{dom}, b) \longrightarrow P(a) \wedge C(b) \tag{9}$$

$$\forall a, b\ (a, \mathtt{range}, b) \longrightarrow P(a) \wedge C(b) \tag{10}$$

$$\forall a, b\ (a, \mathtt{type}, b) \longrightarrow C(b) \tag{11}$$

$$\forall a, b, c\ (a, b, c) \longrightarrow P(b) \tag{12}$$

$$P(\mathtt{sc}) \wedge P(\mathtt{sp}) \wedge P(\mathtt{dom}) \wedge P(\mathtt{range}) \wedge P(\mathtt{type}) \tag{13}$$

The above axioms define the semantics for the `subClass`, `subProperty`, `domain` and `range` predicates.

It is important to observe that `rdfs:subClass`, `rdfs:subProperty`, `rdfs:domain`, `rdfs:range` are defined only by means of *necessary* properties according to the above axioms: the semantics of normative RDFS is a quite weak one, since the RDFS vocabulary does not express fully the corresponding relations in set theory. As a matter of facts, given the RDFS graph from Fig. 1, according the normative RDFS semantics the statement (`:birthCity`, `rdfs:range`, `:Place`) is not entailed. Such an entailment is expected since people do read the properties

[4] Note that for simplicity we may omit the T symbol in FOL formulas.

in the RDFS vocabulary as the corresponding set-based relations – just like in OWL. The normative RDFS semantics is called *intensional*, since it is unable to define sets in terms of their elements.

The Extensional (Non-normative) Semantics

The W3C specification [7] introduces in a "non-normative"status an *extensional* version of RDFS, in which `subClass`, `subProperty`, `domain`, `range` are defined precisely as having the usual set theoretical meaning. This is achieved by adding to the previous definition of the RDFS semantics the missing implication (left-direction arrows) in axioms (1) to (4), thus getting axioms (14) to (17). Thus, axioms (1) to (17) define the semantics of the non-normative extensional RDFS restricted to the ρdf vocabulary. Note that axioms (1) to (8) are redundant, since they can be derived from axioms (9) to (17). From now on we will refer to the non-normative version of RDFS restricted to the ρdf vocabulary as $\boldsymbol{\rho}$**df+**.

$$\forall a, b\ (a, \texttt{sc}, b) \longleftrightarrow C(a) \wedge C(b) \wedge \forall x\ (x, \texttt{type}, a) \rightarrow (x, \texttt{type}, b) \tag{14}$$

$$\forall a, b\ (a, \texttt{sp}, b) \longleftrightarrow P(a) \wedge P(b) \wedge \forall x, y\ (x, a, y) \rightarrow (x, b, y) \tag{15}$$

$$\forall a, c\ (a, \texttt{dom}, c) \longleftrightarrow \forall x, y\ (x, a, y) \rightarrow (x, \texttt{type}, c) \tag{16}$$

$$\forall a, d\ (a, \texttt{range}, d) \longleftrightarrow \forall x, y\ (x, a, y) \rightarrow (y, \texttt{type}, d) \tag{17}$$

This (extensional) semantics – which follows exactly the obvious extensional definitions of the corresponding set-based operators – has been disregarded by the W3C working group because of some computational problems that were conjectured during the definition of the specification. In the non normative section of the W3C specification only a set of *incomplete* inference rules for extensional RDFS is provided.

As for the relations with other KR formalisms, and with the family of description logics in particular, notice that it is easy to see that ρdf+ without typing *exactly* corresponds to the DL-Lite$^{\mathcal{H}}_{\{\text{core,pos,safe}\}}$, namely the well known DL-Lite$^{\mathcal{H}}_{\{\text{core}\}}$ description logic [2, 1] without negation and unqualified existential restrictions on the right-hand side of the inclusion axioms. Obviously, DL-Lite$^{\mathcal{H}}_{\{\text{core,pos,safe}\}}$ *includes* the normative RDFS. It is easy to see that the usual unqualified number restrictions of DL-Lite$_{core}$, once on the left-hand side of the inclusion axioms, can be used to encode the `rdfs:domain` and `rdfs:range` statements, while `rdfs:subClass` and `rdfs:subProperty` are nothing but usual *DL* concept and role inclusion axioms, respectively.

Although the semantics of RDFS dates back to 2004 and despite the large amount of research around it, there were still some important open problems concerning extensional RDFS: i) whether a sound and complete system of inference rules existed; ii) whether a polynomial algorithm for computing the completion according to these extensional rules existed; iii) whether the problem of entailment checking, crucial for query answering, can still be done in the same complexity bound as for intensional RDFS. In this paper we tackle these three problems and provide positive answers to each of them.

3 Reasoning with ρdf+: A forward-chaining system

This section presents a set of sound and complete inference rules for ρdf+ that captures the extensional semantics of RDFS. Our findings complement the set of rules in the ρdf fragment with additional rules derived from the analysis of axioms (14)-(17). The complete set of rules is presented in Table 1. For example, the missing deduction in Fig. 1 can be done now with rule 4(b) with the instantiations A =`birthCity`, B =`birthPlace` and C =`Place`.

We will need some definitions for the discussion that follows. We follow the notations of [11].

Definition 2 (Instantiations and maps).

1. *An* instantiation *of a rule is a uniform replacement of the meta variables occurring in the triples of the rule with elements in $\mathcal{UBL}$, such that all the triples obtained after the replacement are well-formed RDF triples.*
2. *A map is a function $\mu : \mathcal{UBL} \to \mathcal{UBL}$ preserving URIs and literals i.e., $\mu(u) = u$ for all $u \in \mathcal{UL}$. Given a graph $\mathcal{G}$ we define $\mu(\mathcal{G}) = \{(\mu(s), \mu(p), \mu(o)) : (s, p, o) \in \mathcal{G}\}$. By abusing notation, we speak of a map μ from a graph $\mathcal{G}_1$ to a graph $\mathcal{G}_2$ and write $\mu : \mathcal{G}_1 \to \mathcal{G}_2$ if μ is such that $\mu(\mathcal{G}_1)$ is a subgraph of $\mathcal{G}_2$.*

Definition 3 (Proof). *Let $\mathcal{G}$ and $\mathcal{H}$ be graphs. We say that $\mathcal{G} \vdash_{\rho df+} \mathcal{H}$ iff there exists a sequence of graphs $P_1, P_2, \ldots, P_k$, with $P_1 = \mathcal{G}$ and $P_k = \mathcal{H}$, and for each j $(2 \leq j \leq k)$ one of the following cases hold:*

- *there exists a map $\mu : P_j \to P_{j-1}$ (rule 8),*
- *there is an instantiation $\frac{R}{R'}$ of one of the rules (1)–(7) in Table 1 such that $R \subseteq P_{j-1}$ and $P_j = P_{j-1} \cup R'$.*

The sequence of rules used at each step (plus its instantiation or map), is called a proof *of $\mathcal{H}$ from $\mathcal{G}$.*

The ρdf+ system of rules extends the ρdf system [11] by the rules 3(b), 3(c), 4(b), 4(c) and (7). The following theorem states the soundness and completeness of $\vdash_{\rho\text{df+}}$.

Theorem 1 (Soundness and completeness). *Let $\models_{\rho df+}$ denote the entailment relation for the extensional ρdf+ semantics obtained from the axioms (1)-(17). Then, the proof system $\vdash_{\rho df+}$ (rules in Table 1) is sound and complete for this extensional semantics; that is, for $\mathcal{G}$ and $\mathcal{H}$ graphs in ρdf+, then $\mathcal{G} \vdash_{\rho df+} \mathcal{H}$ iff $\mathcal{G} \models_{\rho df+} \mathcal{H}$.*

Proof. The proof is available in the Appendix. □

Although the natural consequence of Theorem 1 would be that of dropping the intensional (weaker) semantic conditions in the normative semantics and replacing them with the extensional (stronger), it is still necessary to investigate whether ρdf+ brings in some source of complexity when applied to the

1. **Subclass**:

   (a) $\frac{(A,\texttt{sc},B)\ \ (X,\texttt{type},A)}{(X,\texttt{type},B)}$ (b) $\frac{(A,\texttt{sc},B)\ \ (B,\texttt{sc},C)}{(A,\texttt{sc},C)}$

2. **Subproperty**:

   (a) $\frac{(A,\texttt{sp},B)\ \ (X,A,Y)}{(X,B,Y)}$ (b) $\frac{(A,\texttt{sp},B)\ \ (B,\texttt{sp},C)}{(A,\texttt{sp},C)}$

3. **Domain:**

   (a) $\frac{(A,\texttt{dom},B)\ \ (X,A,Y)}{(X,\texttt{type},B)}$ (b) $\frac{(A,\texttt{sp},B)\ \ (B,\texttt{dom},C)}{(A,\texttt{dom},C)}$ (c) $\frac{(A,\texttt{dom},B)\ \ (B,\texttt{sc},C)}{(A,\texttt{dom},C)}$

4. **Range:**

   (a) $\frac{(A,\texttt{range},B)\ \ (X,A,Y)}{(Y,\texttt{type},B)}$ (b) $\frac{(A,\texttt{sp},B)\ \ (B,\texttt{range},C)}{(A,\texttt{range},C)}$ (c) $\frac{(A,\texttt{range},B)\ \ (B,\texttt{sc},C)}{(A,\texttt{range},C)}$

5. **Subclass Reflexivity:**

   (a) $\frac{(A,\texttt{sc},B)}{(A,\texttt{sc},A)\ \ (B,\texttt{sc},B)}$ (b) $\frac{(X,p,A)}{(A,\texttt{sc},A)}$ for $p \in \{\texttt{dom}, \texttt{range}, \texttt{type}\}$

6. **Subproperty Reflexivity:**

   (a) $\frac{(X,A,Y)}{(A,\texttt{sp},A)}$ (c) $\frac{}{(p,\texttt{sp},p)}$ for $p \in \rho\text{df}$

   (b) $\frac{(A,\texttt{sp},B)}{(A,\texttt{sp},A)\ \ (B,\texttt{sp},B)}$ (d) $\frac{(A,p,X)}{(A,\texttt{sp},A)}$ for $p \in \{\texttt{dom}, \texttt{range}\}$

7. **Extensional:**

   $\frac{(\texttt{type},\texttt{sp},A)\ \ (A,\texttt{dom},B)\ \ (X,\texttt{sc},X)}{(X,\texttt{sc},B)}$

8. **Simple:**

   $\frac{\mathcal{G}}{\mathcal{G}'}$ for a map $\mu : \mathcal{G}' \to \mathcal{G}$

Table 1. The $\vdash_{\rho\text{df+}}$ rule system for ρdf+. Capital letters A, B, C, X, and Y, stand for meta-variables to be replaced by actual terms in $\mathcal{UBL}$.

following important reasoning tasks: i) computation of the closure; ii) checking of entailment, crucial for query answering.

Computational properties of ρdf+

The *deductive closure* of a graph $\mathcal{G}$ is the graph obtained by adding to $\mathcal{G}$ all triples that are derivable from $\mathcal{G}$. It can be computed by applying systematically and recursively the inference rules in Table 1 to all the triples of $\mathcal{G}$. The deductive closure of a ρdf+ graph is in principle infinite, due to the rule 8, which possibly introduces new blank nodes. In order to get a finite but still useful *completion* of the graph we can consider the closure of $\mathcal{G}$ over the same vocabulary of $\mathcal{G}$, that is, by adding only triples derivable from $\mathcal{G}$ which have elements in $\text{voc}(\mathcal{G}) \cup \mathcal{V}_{\rho\text{df}}$. We will denote this restricted closure by $cl_g(\mathcal{G})$ be the *ground closure* (or *completion*) of a graph $\mathcal{G}$ as the closure via the $\vdash_{\rho\mathrm{df}+}$ ground rule system (rules (1)-(7) in Table 1).

By observing that the number of existing triples with vocabulary in $\mathrm{voc}(\mathcal{G}) \cup \mathcal{V}_{\rho\mathrm{df}}$ is of the order $O(|\mathcal{G}|^3)$, and that all new triples in the closure of $\mathcal{G}$ will be obtained by a successive applications of the rules of the proof system, we obtain the following result:

Proposition 1 (Closure complexity). *The size of the ground closure of a ρdf graph $cl_g(\mathcal{G})$ is at most $O(|\mathcal{G}|^3)$ and it can be computed in polynomial time.*

We will now present a result which states how ρdf+ entailment can be constructively reduced to computing (possibly offline) and materialising the finite polynomial completion of the data graph and then by querying the completion with a standard RDF *simple entailment* query engine. Note that this is the very same procedure which is used in real systems for the standard normative RDFS entailment – of course with the reduced set of normative RDFS inference rules.

Proposition 2 (Entailment for ρdf+). *Consider two RDFS graphs $\mathcal{G}$ (data) and $\mathcal{H}$ (pattern). Then $\mathcal{G} \models_{\rho df+} \mathcal{H}$ iff $cl_g(\mathcal{G}) \models_{\mathrm{RDF}_{\mathrm{simple}}} \mathcal{H}$.*

Proof. By the completeness theorem, $\mathcal{G} \vdash_{\rho\mathrm{df}+} \mathcal{H}$, which by definition of the closure is equivalent to $cl_g(\mathcal{G}) \vdash_{\rho\mathrm{df}+} \mathcal{H}$, which means that $\mathcal{H}$ is in the completion $cl_g(\mathcal{G})$, unless there is an application of rule 8. In this case, $\mathcal{H}$ is got by using the RDF simple entailment in the entailment checking –because of the homomorphism checking. □

It can be easily seen that the combined complexity of entailment (in the size of both graphs) is exactly the same as for normative RDFS and the ρdf system, which is polynomial if $\mathcal{H}$ is a ground graph, and NP-hard otherwise [11]. On the other hand, the data complexity of entailment (that is, only in the size of the data graph $\mathcal{G}$) is polynomial [4].

Materializing all data by computing the completion may cause a waste of space if most of it is never really used. Deciding whether applying materialization or checking entailment on the fly with a specific algorithm depends on different factors such as: i) size of the graph: some graphs may not fit in the main memory and then the completion cannot be avoided; ii) updates: removing a triple from the graph, causes implicit data to still exist if no special care is taken to remove it. Hence, materialization vs. on the fly checking is a trade-off between the better performance of updates, or better performance of look-ups. For this purpose we have studied a refutation proof system provably sound and complete for ρdf+ based on tableaux calculus, which in addition to ρdf+ deals also with negative atoms in the data graph. Such a system, which we do not present here, is used to check entailment on the fly whenever it is not convenient to materialise the completion (see [5] for further details).

4 Reasoning with extensional RDFS in practice

The aim of this section is to illustrate with simple examples the practical impact of extensional RDFS reasoning. We discuss how the $\vdash_{\rho\text{df}+}$ system of rules can be embedded into the Apache Jena library and the impact that it has on the computation of the completion of an RDFS graph.

The Jena inference engine
Jena is a comprehensive Semantic Web library providing a set of features for data management and reasoning in OWL and RDFS. The library features four predefined reasoning engines: i) *transitive reasoner*, which just considers transitive and reflexive properties of RDFS `sc` and `sp`; ii) a configurable *RDFS rule reasoner*; iii) a configurable *OWL reasoner*; iv) *a custom reasoner*. This latter reasoner enables to provide a custom set of inference rules; it supports three reasoning strategies: i) one implementing the *RETE algorithm*; ii) a *forward reasoner*; iii) a *backward reasoner*.

The availability of the custom reasoner is at the core of the integration of the *ground ρdf+ rule system*; we have not implemented rule 7, since we assume that data graphs do not redefine `rdf:type`, that is, they do not have it in subject nor object position. As an example the rule 3 (c) in Table 1 is specified in Jena as: `[3c: (?a dom ?b), (?b sc ?c)->(?a dom ?c)]`. The specification follows the pattern `[label: Ant ->Cons]` where `label` is a name assigned to the rule, `Ant` is the antecedent and `Cons` the consequent. It is also worth mentioning that the reasoner can be configured to log derivations so that each triple obtained after the reasoning task has associated an "explanation", that is, the reasoning steps (in terms of rules triggered) that led to the triple. The reader can consult the Web page `https://jena.apache.org/documentation/inference` for further details.

Comparing inferences at schema level
We investigated the impact of ρdf+ on the completion of five existing ontologies. This experiment only considers triples at schema level; as discussed previously, we do not need to analyze derived `rdf:type` triples, since they would be the same as the `rdf:type` triples derived by a normative RDFS reasoner. Table 2 provides some information about the ontologies considered.

Ontology	#Classes	#Properties	#`dom`	#`range`	#`sc`	#`sp`
DBpedia	359	1775	1505	1553	369	-
FOAF	24	51	47	46	15	10
NEPOMUK	399	628	535	561	460	258
MusicOnto	70	97	97	97	68	25
VoxPopuli	140	66	61	78	140	-

Table 2. Statistics about the ontologies considered.

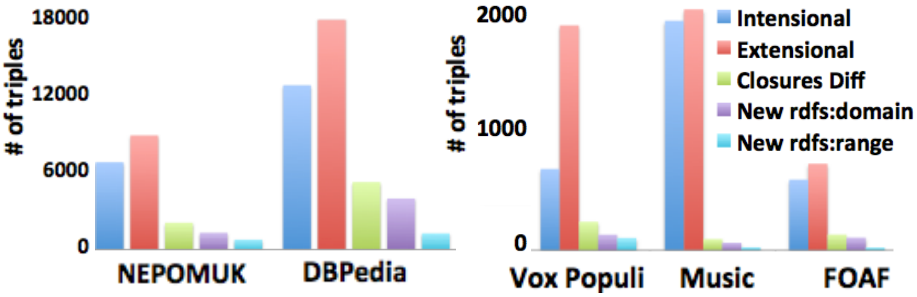

Fig. 2. Size of the completions.

The considered ontologies have different sizes; they range from small ontologies such as FOAF (Friend-of-a-Friend) or MusicOnto (Music Ontology) to relatively large ontologies like NEPOMUK and DBpedia. None of these (real-life) ontologies includes RDF triples redefining the RDFS vocabulary, that is, containing the ρdf vocabulary in subject or object position. Fig. 2 shows some statistics about the completion of the ontologies by considering the ρdf (intensional RDFS) and ground ρdf+ (extensional RDFS) rule systems. The comparison between the completions in terms of number of triples is also shown . As it can be observed with ρdf+ we obtain a larger number of triples. This is due to the presence of the rules 3(b), 3(c), 4(b) and 4(c) in Table 1 that enable to derive new `rdfs:domain` and `rdfs:range` relations. The largest number was obtained when considering DBpedia (∼4000 `rdfs:domain` and ∼ 1200 `rdfs:range`). The extensional completion contains an increase of triples of the order of 30% for DBpedia and NEPOMUK, 60% for VoxPopuli, 20% for FOAF and 5% for MusicOnto. Fig. 3 reports the times (in ms) taken to compute the completion.

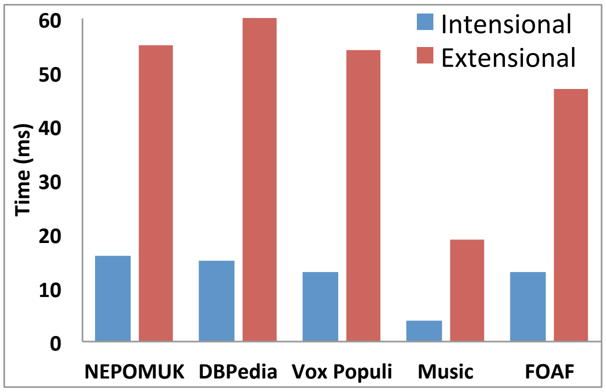

Fig. 3. Times for computing the completions.

In the extensional case more time is needed because of the presence of additional inference rules. However, it can be observed that the time remains around 60ms with a large schema like DBpedia.

In order to give a hint on the kind of derivations enabled via ρdf+, Fig. 4 shows two examples from DBpedia. In Fig. 4 (a) it is shown the new `rdfs:range` for the property `:beltwayCity` obtained by applying rule 4 (c). Fig. 4 (b) shows the derivation of a new `rdfs:domain` for the property `:prospectTeam` obtained via rule 3(c).

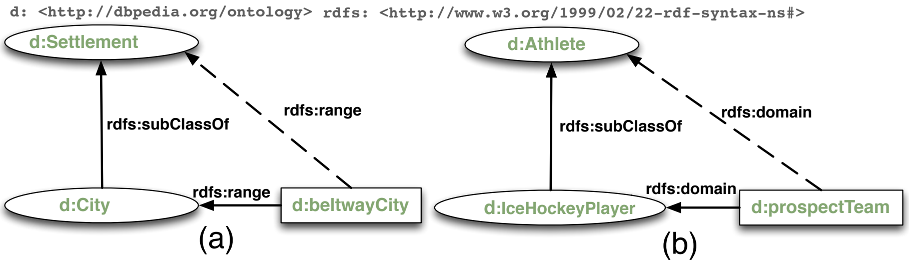

Fig. 4. Examples of new derivations with ρdf+

5 Related Work

There is a solid body of research on RDFS. A formalisation of RDF regarding databases issues was done by Gutierrez et al. [6]. Marin [10] and ter Horst [12] came up with counterexamples (see Fig. 1) which, though pointing to the incompleteness of the W3C RDF Semantics specification rules, showed an issue belonging to the intensional approach to RDFS. The merit of Marin was to overcome the issue keeping the original rules, and adding two additional ones, and proved that the new set of rules was sound and complete. ter Horst instead modified the rule system by allowing non-legal RDFS triples within the rule system by using blank nodes in the predicate position. The formalization of the semantics of RDF in FOL has been studied by de Brujin et al. [4]. Muñoz et al. [11] introduced the ρdf fragment; this paper also discusses the quadratic lower bound for the size of the completion of a graph $\mathcal{G}$ pointing out how such size is impractical from a database point of view. To cope with this issue, the authors introduce *minimal* RDFS, which imposes restrictions on the occurrence of the RDFS vocabulary (it can only occur in predicate position). The advantage of minimal RDFS is that there exists an efficient algorithm to check graph entailment in the case of ground graphs. They also showed that if triples contain at most one blank node the bound remains the same.

The common ground of these approaches is that they stick with the normative specification, that is, intensional RDFS. Other approaches such as RDF-F-Logic [13] depart from the normative specification. Finally, yet other approaches focus on the interplay between RDFS and other ontology languages such as OWL (e.g., RDFS(DL) [3]) and the family of description logics DL-Lite [2, 1]. In contrast to the above approaches, our goal in this paper is to provide a bridge between the normative (intensional) and non normative (extensional) parts of the RDFS specification, and study systematically the latter.

6 Conclusions

In this paper we investigated the extensional semantics for RDFS. Based on the non-normative specification given in the standard W3C RDF semantics specification [7], we develop proof systems that show that one can get a practical, efficient and simple system for the extensional version of RDFS.

We answered an open problem since the publishing of the W3C RDF Semantics [7], which asked for the existence of a simple and efficient system of rules to codify extensional RDFS entailment. The results presented in the paper showed that providing a set of sound and complete inference rules for extensional RDFS is possible, and the complexity of computing the completion of an RDFS graph remains the same as in the normative case.

Our results will impact on current reasoning libraries (e.g., Jena) for RDFS that now can obtain more inferences at no significantly additional cost, as emphasized by our evaluation. Last, but not least, this extensional version aligns the semantics of RDFS and OWL, which previously were inconsistent due to the different meanings given by each of them to set-based notions such as subclass and subproperty.

Acknowledgments

We thank the anonymous referees that provided helpful suggestions. C. Gutierrez thanks the EMCL program for his stay at FUB. Franconi, Pirrò, Mosca and Gutierrez were supported by Marie Curie action IRSES - Net2 (Grant No. 24761). Gutierrez was supported by FONDECYT (Grant No. 1110287). R. Rosati was partially supported by the EU by FP7 project Optique – Scalable End-user Access to Big Data (Grant No. FP7-318338).

References

1. Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev. The DL-Lite family and relations. *J. Artif. Intell. Res. (JAIR)*, 36:1–69, 2009.
2. Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *Journal of Automated Reasoning*, 39(3):385–429, 2007.
3. B. Cuenca Grau. A possible simplification of the semantic web architecture. In *WWW*, pages 704–713. ACM, 2004.

4. J. De Bruijn, E. Franconi, and S. Tessaris. Logical reconstruction of normative RDF. In *OWL: Experiences and Directions Workshop (OWLED-2005), Galway, Ireland*, 2005.
5. E. Franconi, C. Gutierrez, A. Mosca, G. Pirrò, and R. Rosati. A Refutation System for Extensional RDFS. Technical report, KRDB, Free University of Bozen-Bolzano, 2013 - http://www.inf.unibz.it/krdb/pub/tech-rep.php.
6. C. Gutierrez, C. A Hurtado, A. O. Mendelzon, and J. Pérez. Foundations of semantic web databases. *Journal of Computer and System Sciences*, 77(3):520–541, 2011.
7. P. Hayes and B. McBride. RDF semantics. http://www.w3.org/tr/rdf-mt, W3C Reccomandation, 2004.
8. T. Heath and C. Bizer. Linked data: Evolving the web into a global data space. *Synthesis lectures on the semantic web: theory and technology*, 1(1):1–136, 2011.
9. G. Klyne, J. J Carroll, and B. McBride. Resource description framework (RDF): Concepts and abstract syntax. *W3C recommendation*, 10, 2004.
10. D. Marin. A formalization of rdf. Technical report, Technical Report TR/DCC-2006-8, TR Dept. Computer Science, Universidad de Chile, 2006.
11. S. Muñoz, J. Pérez, and C. Gutierrez. Simple and efficient minimal RDFS. *Journal of Web Semantics*, 7(3):220–234, 2009.
12. H. J ter Horst. Completeness, decidability and complexity of entailment for RDF schema and a semantic extension involving the OWL vocabulary. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3(2):79–115, 2005.
13. G. Yang and M. Kifer. Reasoning about anonymous resources and meta statements on the semantic web. In *Journal on Data Semantics I*, pages 69–97. Springer, 2003.

Appendix: Proof of Theorem 1

The following provides a sketch of the argument that proves the completeness of the $\vdash_{\rho\mathrm{df}+}$ rule system: For graphs $\mathcal{G}$ and $\mathcal{H}$ in the ρdf+ vocabulary:

$$\mathcal{G} \vdash_{\rho\mathrm{df}+} \mathcal{H} \text{ iff } \mathcal{G} \models_{\rho\mathrm{df}+} \mathcal{H}.$$

While the soundness theorem (from left to right) follows straightforwardly from the observation that each rule in $\vdash_{\rho\mathrm{df}+}$ preserves validity, the completeness theorem (from right to left) requires more effort to be proved. The proof is heavily based in the completeness theorem for the similar (intensional) $\vdash_{\rho\mathrm{df}}$ system given in [11]. The notions of $\models_{\rho\mathrm{df}}$ and $\vdash_{\rho\mathrm{df}}$ can be found in that paper. First, we need some auxiliary notion of extended closure.

Definition 4. *The extended closure of a graph $\mathcal{G}$, denoted $\widehat{cl}(\mathcal{G})$, is the set of triples entailed from $\mathcal{G}$ under ρdf entailment ($\models_{\rho df}$) plus the axioms (14) - (17).*

We now rephrase $\widehat{cl}(\mathcal{G})$ using the $\vdash_{\rho\mathrm{df}}$ rule system instead of $\models_{\rho\mathrm{df}}$ entailment.

Lemma 1. *The* extended closure *of a graph $\mathcal{G}$ is the set of triples derived from $\mathcal{G}$ using $\vdash_{\rho df}$ plus the axioms (14) - (17).*

Proof. Use the known fact (Theorem 8 from [11]) that, if graphs $\mathcal{G}$ and $\mathcal{H}$ are in the ρdf vocabulary, $\mathcal{G} \vdash_{\rho\mathrm{df}} \mathcal{H}$ iff $\mathcal{G} \models_{\rho\mathrm{df}} \mathcal{H}$. □

The next lemma is at the key to the proof of the theorem:

Lemma 2 (Main). *If graphs $\mathcal{G}$ and $\mathcal{H}$ are in the ρdf vocabulary, then*

$$\widehat{cl}(\mathcal{G}) \vdash_{\rho df} \mathcal{H} \textit{ iff } \mathcal{G} \vdash_{\rho df+} \mathcal{H}.$$

From Lemma 1 above it follows that we only have to show how each triple derived with the axioms (14) - (17) can be also derived with $\vdash_{\rho\mathrm{df}+}$ and vice-versa.

The strategy aims at showing, through an *exhaustive combinatoric analysis*, that whatever can be derived by the axioms (14) to (17) can be derived with the $\vdash_{\rho\mathrm{df}+}$ rule system as well. There are two operations working at the syntactic level: *axiom instantiation* and *pattern matching.* By means of these operations one can start combining together the axioms, until no more new syntactically well formed sentences are derivable. The proof strategy then is grounded on the fact that the only significant ways the axioms can be combined together give rise to nothing but the atoms that are present in the $\vdash_{\rho\mathrm{df}+}$ system. Note that we can restrict to the case when $\mathcal{H}$ is one atom, because for ground atoms p, q it holds $\Sigma \models p \wedge q$ iff $\Sigma \models p$ and $\Sigma \models q$.

Proof. We will introduce for convenience auxiliary extended deductive rules allowing "implications" in the antecedent or in the consequent. This allows to codify formulas (14)-(17) as follows:

$$\text{14a} \quad \frac{(A,\mathtt{sc},B)}{(x,\mathtt{type},A)\xrightarrow{\forall x}(x,\mathtt{type},B)} \quad \text{14b} \quad \frac{(A,\mathtt{sc},A)\wedge(B,\mathtt{sc},B)\wedge(x,\mathtt{type},A)\xrightarrow{\forall x}(x,\mathtt{type},B)}{(A,\mathtt{sc},B)} \quad (\mathtt{sc})$$

$$\text{15a} \quad \frac{(P,\mathtt{sp},Q)}{(x,P,y)\xrightarrow{\forall xy}(x,Q,y)} \quad \text{15b} \quad \frac{(A,\mathtt{sc},A)\wedge(B,\mathtt{sc},B)\wedge(x,P,y)\xrightarrow{\forall xy}(x,Q,y)}{(P,\mathtt{sp},Q)} \quad (\mathtt{sp})$$

$$\text{16a} \quad \frac{(P,\mathtt{dom},A)}{(x,P,y)\xrightarrow{\forall xy}(x,\mathtt{type},A)} \quad \text{16b} \quad \frac{(x,P,y)\xrightarrow{\forall xy}(x,\mathtt{type},A)}{(P,\mathtt{dom},A)} \quad (\mathtt{domain})$$

$$\text{17a} \quad \frac{(P,\mathtt{range},A)}{(x,P,y)\xrightarrow{\forall xy}(y,\mathtt{type},A)} \quad \text{17b} \quad \frac{(x,P,y)\xrightarrow{\forall xy}(y,\mathtt{type},A)}{(P,\mathtt{range},A)} \quad (\mathtt{range})$$

The following are a few remarks to be made on the usage of this new system:

1. Rules with an implication in the antecedent (being universally quantified) cannot be fired from the graph $\mathcal{G}$ because of the presence of the *open world assumption*, we cannot know from $\mathcal{G}$ if it is valid or not.
2. Two implications can be matched if the meaning of the formulas allow so. For example, $(x, \mathtt{type}, A) \xrightarrow{\forall x} (x, \mathtt{type}, B)$ and $(y, \mathtt{type}, B) \xrightarrow{\forall y} (y, \mathtt{type}, C)$ would produce another rule:

$$\frac{(x, \mathtt{type}, A) \xrightarrow{\forall x} (x, \mathtt{type}, B) \qquad (y, \mathtt{type}, B) \xrightarrow{\forall y} (y, \mathtt{type}, C)}{(z, \mathtt{type}, A) \xrightarrow{\forall z} (z, \mathtt{type}, C)} \tag{18}$$

3. The only way to use an implication in a combination of rules is, either:

Instantiation/Combination	**Rule obtained**	**Rule in ρdf+**	**Rule in RDFS**
(15a-inst)$\curvearrowright$16a$\curvearrowright$14b	$\frac{(\mathtt{type},\mathtt{sp},A),(A,\mathtt{dom},B),(X,\mathtt{sc},X)}{(X,\mathtt{sc},B)}$	7	not available
(16a-inst)$\curvearrowright$14b$\curvearrowright$14a	$\frac{(\mathtt{type},\mathtt{dom},A),(X,\mathtt{sc},X)}{(X,\mathtt{sc},A)}$	7 bis	not available

Table 3. Inference rules obtained by instantiating and combining rules (14a)-(17a). Rule 7bis can be obtained in turn from 7 and 6c, thus does not appear in Table 1

(a) To combine it with another implication to derive a third implication (e.g., to form rules of the form (18)). Table 4 summarizes the only admissible results one can obtain out the combination operation (we use the notation $r_1 \curvearrowright r_2$ to indicate that rule r_1 is combined with rule r_2). Note that the only possible relevant formula one could get with this procedure is a formula of the type $\forall x(x, \mathtt{type}, A) \to (x, \mathtt{type}, B)$, thus, to deduce a triple of the form $(u, \mathtt{sc}, v)$ using rule (14b). Note also that one cannot use the rules (15b), (16b) or (17b), because they need both variables universally quantified.

(b) To instantiate the implication in the consequent, and using the Deduction Theorem ($p \vdash q{\to}r$ iff $p, q \vdash r$). Consider for instance rule (14a); we have: $(A, \mathtt{sc}, B) \vdash (x, \mathtt{type}, A) \xrightarrow{\forall x} (x, \mathtt{type}, B)$. By using the deduction theorem, we obtain: $(A, \mathtt{sc}, B)\ (x, \mathtt{type}, A) \vdash (x, \mathtt{type}, B)$. By systematically applying this process to rules (14a)-(17a), we obtain the rules in Table 5.

(c) To use instantiation that make it possible to combine rules. For example the new rule 7 Extensional follows from rule (15a) instantiated with $P = \mathtt{type}$, which combined with the rule for domain (16a), gives the implication $\forall x(x, \mathtt{type}, y) \to (x, \mathtt{type}, B)$, which using rule (14b) gives $(y, \mathtt{sc}, B)$ for y class. Table 3 shows the results of the application of the instantiation-plus-combination operation.

Combination	**Rule obtained**	**Rule in $\vdash_{\rho\mathbf{df}+}$**	**Rule in intensional RDFS**
$14a{\curvearrowright}14a$	$\frac{(A,\mathtt{sc},B)\quad(B,\mathtt{sc},C)}{(A,\mathtt{sc},C)}$	1b	rdfs 11
$15a{\curvearrowright}15a$	$\frac{(P,\mathtt{sp},Q)\quad(Q,\mathtt{sp},R)}{(P,\mathtt{sp},R)}$	2b	rdfs 5
$15a{\curvearrowright}16a$	$\frac{(P,\mathtt{sp},Q)\quad(Q,\mathtt{dom},A)}{(P,\mathtt{dom},A)}$	3b	not available
$15a{\curvearrowright}17a$	$\frac{(P,\mathtt{sp},Q)\quad(Q,\mathtt{range},A)}{(P,\mathtt{range},A)}$	4b	not available
$16a{\curvearrowright}14a$	$\frac{(P,\mathtt{dom},A)\quad(A,\mathtt{sc},B)}{(P,\mathtt{dom},B)}$	3c	not available
$17a{\curvearrowright}14a$	$\frac{(P,\mathtt{range},A)\quad(A,\mathtt{sc},B)}{(P,\mathtt{range},B)}$	4c	not available

Table 4. Inference rules obtained by combining rules (14a)-(17a)

The presented proof system is the collection of all rules obtained. In particular, an exhaustive combinatorics indicates that *the only possible cases* are those considered in ρdf+. The idea is as follows:

Rule Instantiated	**Rule obtained**	**Rule in ρdf+**	**Rule in intensional RDFS**
13a	$\frac{(A,\texttt{sc},B)\ \ (X,\texttt{type},A)}{(X,\texttt{type},B)}$	1a	rdfs 9
14a	$\frac{(P,\texttt{sp},Q)\ \ (X,P,Y)}{(X,Q,Y)}$	2a	rdfs 7
15a	$\frac{(P,\texttt{dom},A)\ \ (X,P,Y)}{(X,\texttt{type},A)}$	3a	rdfs 2
16a	$\frac{(A,\texttt{range},B)\ \ (X,A,Y)}{(Y,\texttt{type},B)}$	4a	rdfs 3

Table 5. Set of inference rules obtained by instantiating rules (14a)-(17a)

1. Note that the only possible relevant formula one could get with the introduced procedure is a formula of the type $\forall x(x, \texttt{type}, A) \to (x, \texttt{type}, B)$, thus, to deduce a triple of the form $(u, \texttt{sc}, v)$ using rule (14b). Note that one cannot use the other rules (15b), (16b) or (17b), because they need both variables universally quantified.
2. With (1) in mind, one should start looking for the successful combinations.
   (a) Those that begin with $(x, \texttt{type}, y)$: could be rules (15a), (16a) or (17a) instantiated with $P = \texttt{type}$. As for Rule (15a), we should instantiate also $y = C$, but in this case the rule will give $\forall x(x, \texttt{type}, C) \to (x, Q, C)$, whose consequent cannot be further combined unless $Q = \texttt{type}$, which gives nothing. As for rule (16a), it gives our rule 7bis, while rule (17a) is useless for this argument (notice that in (17a) the y in the implication changes its position from third to first thus making impossible the combination with (14b)).
   (b) Those that end with $(x, \texttt{type}, y)$: here rule (16a) is relevant once y is instantiated to a constant; and rules (16a) and (17a) with the restriction $x = y$. It is not difficult to note that the first case is useful only for the instantiation $P = \texttt{type}$. In the second case, the only productive combination is to combine it with rule (15a) weakened to $x = y$. □

Now are read to prove the statement of Theorem 1:

Proof. $\mathcal{G} \models_{\rho\text{df}+} \mathcal{H}$
iff $\mathcal{G} \models_{\text{RDFS}+} \mathcal{H}$ (by definition of $\models_{\rho\text{df}+}$)
iff $\mathcal{G} \cup \{\text{axioms } 14 - 17\} \models_{\text{RDFS}} \mathcal{H}$ (by definition of RDFS+)
iff $\widehat{cl}(\mathcal{G}) \models_{\text{RDFS}} \mathcal{H}$ (by Definition 4)
iff $\widehat{cl}(\mathcal{G}) \models_{\rho\text{df}} \mathcal{H}$ (Theorem 5 from [11]) because left and right hand sides have only ρdf vocabulary)
iff $\widehat{cl}(\mathcal{G}) \vdash_{\rho\text{df}} \mathcal{H}$ (Soundness and completeness of ρdf –Theorem 8 from [11]– because there is only ρdf vocabulary)
iff $\mathcal{G} \vdash_{\rho\text{df}+} \mathcal{H}$ (by Lemma 2).

□