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] University of Chile, Santiago, Chile
[3] University of Rome La Sapienza, Rome, Italy

Abstract. The normative version of RDFS gives non-standard (intensional) interpretations to some standard notions such as classes and properties. In this paper we develop the extensional semantics for the RDFS vocabulary, which surprisingly preserves the simplicity and computational complexity of deduction of the intensional case. This result will impact current implementations in a positive sense, as reasoning in RDFS will follow common set-based intuitions and be compatible with OWL extensions; moreover, the rule system that we present is easily embeddable in existing libraries such as Jena.

1 Introduction

The Resource Description Framework (RDF) [8] is the standard data model for publishing and interlinking data on the Web. It 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 [7] give a glimpse of the magnitude of RDF data today available. Thousands of datasources covering different domains from general knowledge (e.g., DBPedia [2]) to specific domains are today interlinked and publicly accessible via the SPARQL [11] standard query language for RDF. The uptake of RDF is also witnessed by its adoption by large e-commerce Web sites such as `bestbuy.com`, which provides a query interface for posing structured queries over its RDF datastore. RDF also attracted the attention of companies like Oracle that are now providing RDF-centered data management solutions.

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 standard vocabularies are RDFS (RDF Schema) 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 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 common sense set-based meaning. For example, in Fig. 1 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 the common vocabularies. RDFS, for historical reasons, follows an *intensional*

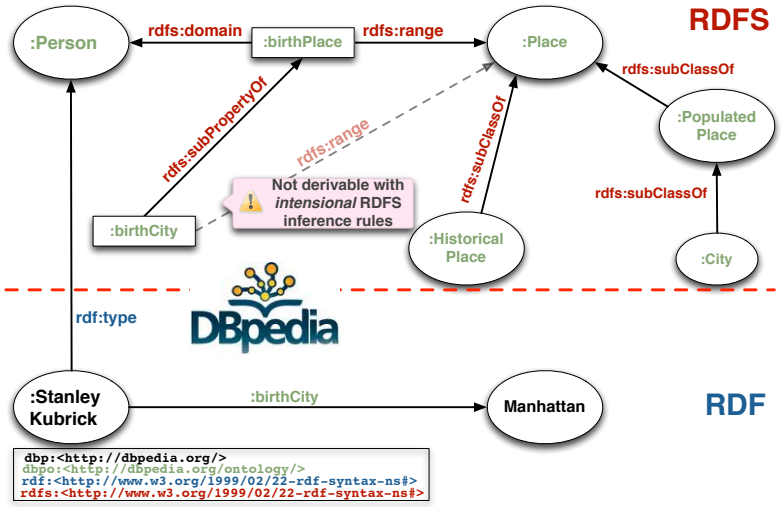


Fig. 1. An RDF(S) graph taken from `dbpedia.org`

semantics while OWL adopts a standard *extensional* set-based semantics. The intensional semantics of RDFS brings several problems, both at the level of deduction and in terms of compatibility with OWL. Consider the example shown in Fig. 1: the dotted `rdfs:range` property would be 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). Until now, there were two relevant open problems regarding this extensional RDFS semantics that have prevented its usage: *i)* which are the computational properties (decidability, complexity) for extensional RDFS?; *ii)* can we extend easily normative RDFS inference engines (based on the computation of a *completion* in a forward-chaining manner) to support completely 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. Not only this is theoretically interesting but it will impact on current implementations (for the most part based on the normative intensional semantics) in a positive

sense. Indeed, reasoning in RDFS will follow common set-based intuitions and be compatible with OWL extensions. Moreover, the rule system that we present is easily embeddable in existing libraries such as Jena.

2 Preliminaries: RDF & RDFS

The Resource Description Framework (RDF) [8] is the W3C's standard data model for the publishing and interlinking of data on the Web. As the name suggests, it is centred around the notion of *resource*, which can be essentially anything. Resources are given identifiers by using Uniform Resource Identifiers (URIs). RDF has a very simple structure; an RDF dataset contains statements, expressed by *triples*. Each triple includes a binary *predicate* (which itself is a resource) relating a *subject* with an *object*; subjects and objects may be resources or values, expressed by *literals*. *Blank nodes* play the role of generic identifiers referring to resources or literals.

Let $\mathcal{U}$, $\mathcal{L}$, $\mathcal{B}$ three pairwise disjoint sets representing URIs, literals and blank node identifiers, 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 the subject, predicate and object, respectively. A triple is ground if it does not contain blank node identifiers. A (ground) RDF graph $\mathcal{G}$ is a set of RDF (ground) triples.*

Definition 2 (Vocabulary). *The set of terms of a graph $\mathcal{G}$, that is $term(\mathcal{G})$ is the set of elements in $\mathcal{UBL}$ that occur in triples of $\mathcal{G}$. The vocabulary of a graph $\mathcal{G}$, denoted by $\mathcal{V}(\mathcal{G})$, is the set $term(\mathcal{G}) \cap \mathcal{UL}$. Given a graph $\mathcal{G}$ and a vocabulary $\mathcal{W}$ we say that $\mathcal{G}$ is a graph over $\mathcal{W}$ whenever $\mathcal{V}(\mathcal{G}) \subseteq \mathcal{W}$. The* RDFS vocabulary *is a set of reserved resource names in the* `rdf:` *namespace (such as, e.g.,* `rdf:type`, `rdf:property`, *etc) and in the* `rdfs:` *namespace (such as, e.g.,* `rdfs:subClass`, `rdfs:subProperty`, `rdfs:domain`, `rdfs:range`, *etc).*

In RDF only primitive statements about resources can be expressed: a resource may be an instance of another resource (representing a class) and/or a property of another resource. In RDFS it is possible to express also hierarchies of classes and properties and to restrict the domain and range of properties. In what follows we assume that the set $\mathcal{U}$ includes the RDF and the RDFS vocabularies. As an example of an RDFS graph, see Fig. 1.

The ρdf fragment.
In this paper we focus our attention on an abstraction of RDFS named ρdf, introduced first in [10], which has been shown to capture the essential semantics of the full fragment, while avoiding to deal with minor idiosyncrasies related to the vocabulary. The ρdf vocabulary is restricted to the following subset of the normative RDFS vocabulary: $\mathcal{V}_{\rho\text{df}} = \{\texttt{sc}, \texttt{sp}, \texttt{dom}, \texttt{range}, \texttt{type}\}$[4], and, unlike in

[4] Their meaning is `rdfs:subClass`, `rdfs:subProperty`, `rdfs:domain`, `rdfs:range`, `rdf:type`, respectively.

normative RDF, literals may appear in subject position within triples. As it has been shown in [10], ρdf is self-contained: it does not rely on the RDFS vocabulary beyond the subset, nor does the rest of the RDFS vocabulary rely on this subset. ρdf is endowed with a set of inference rules that are derived from the original RDFS semantics and extended to cope with the incompleteness of the latter [9].

The normative semantics of RDFS [6] is built upon the standard logic notions of model, interpretation and entailment. In this paper 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 – to 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 `rdfs:subClass`, `rdfs:subProperty`, `rdfs:domain`, `rdfs:range` in terms of `rdf:type` in the obvious way – as in set theory[5]:

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

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

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

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

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

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

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

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

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

The following typing axioms are also needed in normative RDFS:

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

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

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

$$\forall a, b, c\ (a, b, c) \longrightarrow P(b) \quad (12)$$

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 figure 1, according the normative RDFS semantics the statement (`:birthCity`, `rdfs:range`, `:Place`) is

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

not entailed. Such an entailment is expected since people do read the properties 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 ρdf+ fragment.
[6] introduces also an *extensional* non normative version of RDFS, in which `rdfs:subClass`, `rdfs:subProperty`, `rdfs: domain`, `rdfs:range` are exactly defined as their set theoretical counterparts, as expected. This is achieved by adding to the previous definition of the RDFS semantics the axioms (13) to (16) below, which look like axioms (1) to (4) but where the one sided material implications are replaced with if-and-only-if definitions. Axioms (1) to (16) define the semantics of the non normative extensional RDFS restricted to the ρdf vocabulary[6]. From now on we will refer to the non normative RDFS restricted to the ρdf vocabulary as **ρdf+**.

$$\forall a, b\ (a, \texttt{sc}, b) \longleftrightarrow C(a) \wedge C(b) \wedge \forall x\ (x, \texttt{type}, a) \rightarrow (x, \texttt{type}, b) \tag{13}$$

$$\forall a, b\ (a, \texttt{sp}, b) \longleftrightarrow P(a) \wedge P(b) \wedge \forall x, y\ (x, a, y) \rightarrow (x, b, y) \tag{14}$$

$$\forall a, c\ (a, \texttt{dom}, c) \longleftrightarrow \forall x, y\ (x, a, y) \rightarrow (x, \texttt{type}, c) \tag{15}$$

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

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 ρdf+ *exactly* corresponds to the DL-Lite$^{\mathcal{H}}_{\{\text{core,pos,safe}\}}$, namely the well known DL-Lite$^{\mathcal{H}}_{\{\text{core}\}}$ description logic [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.

[6] It is easy to see that axioms (1) to (8) are redundant, since they can be derived from axioms (9) to (16).

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

7. **Extensional A:**

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

8. **Extensional B:**

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

9. **Simple:**

   (a) $\frac{G}{G'}$ for a map $\mu : G' \to G$ (b) $\frac{G}{G'}$ for $G' \subseteq G$

Fig. 2. The ρdf+ rule system.

3 Reasoning with ρdf+

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 (13)-(16). The new set of rules shown in Fig. 2 correctly derive the statement (`:birthCity`, `rdfs: range`, `:Place`) in Fig. 1; this can be achieved by applying rule 4(b).
We now introduce some definitions that will be useful in the discussion.

Definition 3 (Instantiation of a rule). *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.*

In every rule in Fig. 2, letters A, B, C, X, and Y, stand for meta variables to be replaced by actual terms in $\mathcal{UBL}$. As an example, given $a, b \in \mathcal{U}$, $N \in \mathcal{B}$ and $y \in \mathcal{L}$, then (R/R') with $R = \{(a, \mathtt{sp}, b), (N, a, y)\}$ and $R' = \{(N, b, y)\}$ is an instantiation of rule 2 (a).
We now recall the notions of *Map* and *Proof* [10].

Definition 4 (Map). *A map is a function $\mu : \mathcal{UBL} \to \mathcal{UBL}$ preserving URIs and literals i.e., $\mu(u) = u \ \forall \ u \in \mathcal{UL}$. Given a graph $\mathcal{G}$ we define $\mu(\mathcal{G})$ as the set of all $(\mu(s), \mu(p), \mu(o))$ such that $(s, p, o) \in \mathcal{G}$. By abusing a little bit the above notation, in the following 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 the map μ is such that $\mu(\mathcal{G}_1)$ is a subgraph of $\mathcal{G}_2$.*

Definition 5 (Proof). *Let G and H be graphs. We say that $G \vdash_{\rho df+} H$ iff there exists a sequence of graphs $P_1, P_2, \ldots, P_k$, with $P_1 = G$ and $P_k = H$, 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 9a),*
- *$P_j \subseteq P_{j-1}$ (rule 9b),*
- *there is an instantiation $\frac{R}{R'}$ of one of the rules (2)–(6), 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 H from G.*

The ρdf+ system of rules extends the ρdf system by the six rules 3(b), 3(c), 4(b), 4(c), (7), and (8). The following theorem states the soundness and completeness of ρdf+.

Theorem 1 (Soundness and completeness). *The proof system $\rho df+$ is sound and complete for entailment under the extensional $\rho df+$ semantics ($\models_{\rho df+}$): let $\mathcal{G}$ and H be two graphs in $\rho df+$, then $\mathcal{G} \vdash_{\rho df+} H$ iff $\mathcal{G} \models_{\rho df+} H$.*

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 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 can be obtained by applying systematically the inference rules in Fig. 2 to all the triples of the graph. Unluckily, the deductive closure of a ρdf+ graph is infinite, due to rules 7, 8, and 9, so we do not get directly a constructive way to compute entailment by using the rule system.

In order to get a finite but still useful *completion* graph approximating the deductive closure, let's consider a restricted version of the ρdf+ rule system, called ρdf+ *ground* rule system, which includes only rules from 1 to 8, and restricts the meta-variable X in rules 7 and 8 to be instantiated to just elements in

$term(\mathcal{G}) \cup \mathcal{V}_{\rho\text{df}}$. Indeed, with the absence of rules 9 we avoid the over-generation of bnode names since the bnode names in the data graph are reused systematically by rules 1-8; with the restriction on the instantiation of the consequent meta-variables in rules 7 and 8 we avoid to include triples in the completion with irrelevant new elements. Let $cl_g(\mathcal{G})$ be the *ground closure* (or *completion*) of a graph $\mathcal{G}$ as the closure via the ρdf+ ground rule system.

Definition 6 (Completion). *The* completion *or* ground closure *of a graph $\mathcal{G}$ – denoted by $cl_g(\mathcal{G})$ – is obtained by taking the union of all the graphs for which there exists a proof from $\mathcal{G}$ via the $\rho df+$ ground rule system.*

Theorem 2 (Completion complexity). *The ground closure of an $\rho df+$ graph $cl_g(\mathcal{G})$ is polynomially larger than the size of the graph $\mathcal{G}$, and it can be computed in polynomial time by the non-deterministic exhaustive application of the rules in the $\rho df+$ ground rule system.*

Proof. By inspecting the form of all possible instantiations of the inference rules, we first prove – by induction over the length of proofs – that graphs for which there exists a proof from $\mathcal{G}$ will only include elements from $term(\mathcal{G}) \cup \mathcal{V}_{\rho\text{df}}$. It is easy to see then that the number of triples in the ground closure is within the order of $\mathcal{O}(\|term(\mathcal{G})\|^3)$ □

We can now state the main theorem of this paper, which states how ρdf+ entailment can be constructively reduced to computing (possibly offline) and materializing 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.

Theorem 3 (Entailment for ρdf+). *Consider a data RDFS graph $\mathcal{G}$ and a query RDFS graph H (i.e., a SPARQL basic graph pattern) such that either (1) `rdf:type` never appears in subject or object position within triples of $\mathcal{G}$, or (2) $term(H) \subseteq term(\mathcal{G})$. Then $\mathcal{G} \models_{\rho df+} H$ iff $cl_g(\mathcal{G}) \models_{RDF_{simple}} H$.*

Proof. Due to the completeness theorem, all the graphs entailed by $\mathcal{G}$ are in its completion, with the exception of (a) the entailed graphs using elements outside $term(\mathcal{G}) \cup \mathcal{V}_{\rho\text{df}}$, due to the restriction to the application of rules 7 and 8, and with the exception of (b) some entailed graphs where some elements of a graph in the completion are consistently replaced by bnodes, due to the lack of rule 9(a). If condition (1) applies, it can be seen that rules 7 and 8 are never applicable – because there is no way in which `rdf:type` could go from property position into the subject position: in all rules, all subject positions come either from subject positions or from object positions; therefore the lacking entailed graphs of type (a) do not play any role for checking entailment. Similarly, the lacking entailed graphs of type (a) do not play any role for checking entailment if condition (2) applies, since no new elements are expected to appear in H. The lacking entailed graphs of type (b) are recovered 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 H is a ground graph, and NP-hard otherwise [10]. On the other hand, the data complexity of entailment (that is, only in the size of the data graph $\mathcal{G}$) is polynomial [4].

The theorem states very realistic alternative restrictions on the form of the data graph or of the query: either the data graph does not redefine the `rdf:type` property – something that we have never seen happening in any real RDFS knowledge base, or the query makes use of only URIs already appearing in the data graph. If the first condition is met, then the ρdf+ ground rule system can be easily implemented in any RDF rule engine – such as Jena. As a matter of facts, rules 7 and 8 can't be implemented within conventional rule engines, since they imply the assertion of their consequents for all the URI and blank node names appearing in the graph. But if condition (1) of the theorem is met, then rules 7 and 8 would never fire, and so they do not need to be implemented. We can also observe that whenever condition (1) is met, then according to the ρdf+ extensional semantics no additional `rdf:type` triple is entailed with respect to the `rdf:type` triples entailed according to the intensional ρdf semantics.

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 and it does not undergo the restrictive conditions of Theorem 3. Such a system, which we do not present here, is used to check entailment on the fly whenever it is not convenient to materialize the completion.

4 Experiments

In Section 3 we have proved that the ρdf+ is a sound and complete set of rules capturing the extensional non normative RDFS semantics. The aim of this section is to show the practical impact of ρdf+; we discuss how the ρdf+ 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.

Jena inference engine.
Jena is a comprehensive Semantic Web library providing a set of features for data management and reasoning in OWL and RDF(S). 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

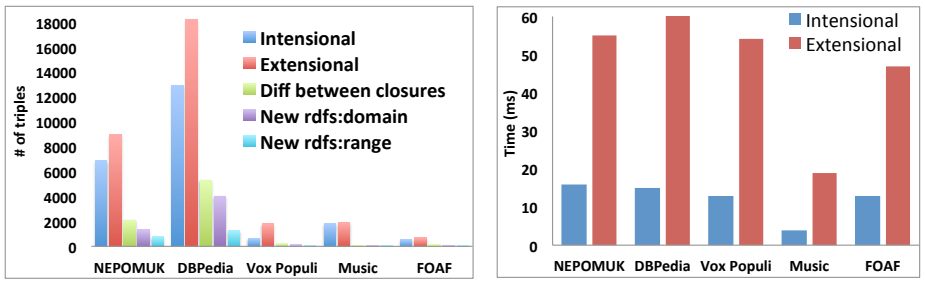
Ontology	#Classes	#Properties	#dom	#range	#sc	#sp
DBpedia	359	1775	1505	1553	369	-
FOAF	24	51	47	46	15	10
NEPOMUK	399	628	535	561	460	258
MusicOnto	70	97	97	97	68	25
VoxPopuli	140	66	61	78	140	-

Fig. 3. Statistics about the ontologies considered.

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 basis of the integration of the *ground ρdf+ rule system*; as discussed in the previous section, we haven't implemented rules 7 and 8, since we assume that data graphs do not have `rdf:type` neither in subject nor in object position. As an example the rule 3 (c) in Fig. 2 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 to mention that the reasoner can be configured to log derivations so that each triples obtained after the reasoning task has associated an "explanation", that is, the reasoning steps (in terms of rules triggered) that led to the triple.

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 analyse derived `rdf:type` triples, since they would be the same as the `rdf:type` triples derived by a normative RDFS reasoner. The table in Figure 3 provides some information about the ontologies considered. As for the DBpedia ontology, we have replaced OWL datatype and object properties with the corresponding RDF properties and OWL classes with RDFS classes. The considered ontologies have different sizes; they range from small ontologies such as FOAF (Friend-od-a-Friend) or MusicOnto (Music Ontology) to relatively large ontologies like NEPOMUK and DBpedia. None of these (real-life) ontologies



(a) Size of the completions. (b) Times for computing the completions.

Fig. 4. Comparing extensional and intensional completions at schema level.

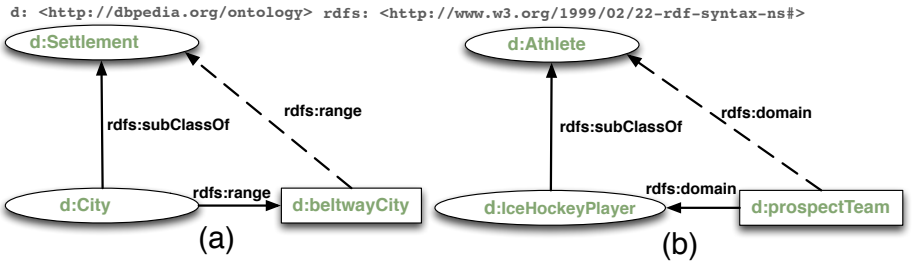


Fig. 5. Examples of new derivations with ρdf+

includes RDF triples redefining the RDFS vocabulary, that is, containing the ρdf vocabulary in subject or object position.

Fig. 4 shows some statistics about the completion of the ontologies by considering the ρdf (intensional RDFS) and ground ρdf+ (extensional RDFS) rule systems. Fig. 4 (a) shows the comparison between the completions in terms of number of triples. 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 Fig. 2 in the system that enable to derive new `rdfs:domain` and `rdfs:range` relations. The largest number was obtained when considering DBpedia ($\sim$4000 `rdfs:domain` and $\sim$ 1200 `rdfs:range`). The extensional completion contains an increasing of triples in the order of 30% for DBpedia and NEPOMUK, 60% fir VoxPopuli, 20% for FOAF and 5% for MusicOnto. Fig.4 (b) reports the times (in ms) that the reasoning engine took to compute the completion.

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. 5 shows two examples from DBpedia. In Fig. 5 (a) it is shown the new `rdfs:range` for the property `:beltwayCity` obtained by applying rule 4 (c). Fig. 5 (b) shows the derivation of a new `rdfs:domain` for the property `:prospectTeam` obtained via rule 3(c).

5 Related Work

There has been a solid body of research around RDFS. A formalization of RDF and its links with databases is due to Gutierrez et al. [5]. Marin [9] and ter Horst [12] came up with counterexample (see Fig. 1) pointing out the incompleteness of RDFS (intensional) inference rules. The merit of Marin was to overcome the issue while keeping the original rules, adding two additional rules and show 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. [10] 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, authors introduce *minimal* RDFS, which imposes restrictions on the position of triples of the RDFS vocabulary (they can only appear in predicate position). The advantage of minimal RDFS is that there exists a logarithm algorithm to check graph entailment in the case of ground graphs. Authors 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 other ontology languages such as OWL (e.g., RDFS(DL) [3]) and, as we have seen, the family of description logics DL-Lite [1]. The aim of this paper is to study RDFS from the logical point of view. Differently from the above mentioned approaches, we provide a bridge between the normative and non normative part of the RDFS specification. In particular, we investigated two important open problems: *i)* whether there exists a sound and complete set of inference rules for extensional RDFS; *ii)* whether the completion of an graph under extensional RDFS can still be computed in polynomial time. We pointed out the both problems can be answered in the positive.

6 Conclusions

In this paper we investigated two relevant open problems regarding RDFS. These two problems stem from the non-standard definition of the normative semantics of RDF [6], which although being based on logic notions does not follow a standard set-theoretical approach. Indeed, the normative semantics of RDF is *intensional* while the (natural and set-theory-compliant) extensional semantics has been included in the non normative part of the specification, and it is used, e.g., in OWL. We showed that providing a set of sound and complete inference rules for extensional RDFS is possible. Moreover, in this new setting, the complexity of computing the completion of an RDFS graph remains the same as in the normative case. We also discussed how the complexity of the entailment for RDFS remains the same as that of the normative case. Our results will impact on current reasoning libraries for RDFS that now can obtain more inferences at no additional cost, as emphasised by our evaluation.

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. C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia-a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154–165, 2009.
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. 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.
6. P. Hayes. RDF semantics. W3C recommendation. 2004.
7. 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.
8. G. Klyne, J. J Carroll, and B. McBride. Resource description framework (RDF): Concepts and abstract syntax. *W3C recommendation*, 10, 2004.
9. D. Marin. A formalization of rdf. Technical report, Technical Report TR/DCC-2006-8, TR Dept. Computer Science, Universidad de Chile, 2006.
10. S. Muñoz, J. Pérez, and C. Gutierrez. Simple and efficient minimal RDFS. *Journal of Web Semantics*, 7(3):220–234, 2009.
11. E. PrudHommeaux, A. Seaborne, et al. SPARQL query language for rdf. *W3C recommendation*, 15, 2008.
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 sketch of Theorem 1

In the following we provide a sketch of the argument that proves the completeness of the ρdf+ rule system. Assuming that the graphs $\mathcal{G}$ and H are in the ρdf+ vocabulary, the main statement was:

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

While the soundness theorem (from left to right) follows straightforwardly from the observation that each rule in ρdf+ preserves validity (i.e., given the validity of the antecedent, the validity of the consequent is guaranteed), the completeness theorem (from right to left) requires a bit of effort to be proved.

First, we need the following auxiliary notion of extended closure.

Definition 7. *The extended closure of a graph $\mathcal{G}$ – called $\widehat{cl}(\mathcal{G})$ – is made by all the triples entailed by $\mathcal{G}$ under the ρdf entailment together with the axioms (13) - (16).*

We now rephrase $\widehat{cl}(\mathcal{G})$ using the ρdf rule system instead of ρdf entailment.

Lemma 1. *The* extended closure *of a graph $\mathcal{G}$ is made by all the triples derived from $\mathcal{G}$ using the ρdf rule system plus the triples entailed from those by the axioms (13) - (16).*

Proof. Use the known fact (Theorem 8 from [10]) that, if graphs $\mathcal{G}$ and H are in the ρdf vocabulary, $\mathcal{G} \vdash_{\rho\text{df}} H$ iff $\mathcal{G} \models_{\rho\text{df}} H$. □

The next lemma is at the core of the proof of the theorem.

Combination	Rule obtained	Rule in ρdf+	Rule in intensional RDFS
$13a \curvearrowright 13a$	$\frac{(A,\mathtt{sc},B)\ \ (B,\mathtt{sc},C)}{(A,\mathtt{sc},C)}$	1b	rdfs 11
$14a \curvearrowright 14a$	$\frac{(P,\mathtt{sp},Q)\ \ (Q,\mathtt{sp},R)}{(P,\mathtt{sp},R)}$	2b	rdfs 5
$14a \curvearrowright 15a$	$\frac{(P,\mathtt{sp},Q)\ \ (Q,\mathtt{dom},A)}{(P,\mathtt{dom},A)}$	3b	not available
$14a \curvearrowright 16a$	$\frac{(P,\mathtt{sp},Q)\ \ (Q,\mathtt{range},A)}{(P,\mathtt{range},A)}$	4b	not available
$15a \curvearrowright 13a$	$\frac{(P,\mathtt{dom},A)\ \ (A,\mathtt{sc},B)}{(P,\mathtt{dom},B)}$	3c	not available
$16a \curvearrowright 13a$	$\frac{(P,\mathtt{range},A)\ \ (A,\mathtt{sc},B)}{(P,\mathtt{range},B)}$	4c	not available

Fig. 6. Inference rules obtained by combining rules (13a)-(16a)

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

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

Proof. From Lemma 1 above it follows that we only have to show how each triple derived with the axioms (13) - (16) can be also derived with the ρdf+ rule system and viceversa.
Since of the presence of the axiomatic system defining the semantics of the non normative RDFS (see axioms from (13) to (16)), the strategy we introduce here aims at showing, through an *exhaustive combinatorics analysis* that whatever can be derived by the axioms can be derived in the the ρdf+ rule system as well. No surprisingly, what can be done using the introduced finite axiomatisation is basically driven by two operations working at the syntactic level: *axiom instantiation*, and *pattern matching*. By means of these two operations, one can start combining together the axioms, until no more new syntactically well formed sentences is 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 ρdf+, and this proved our claim. Note that we can restrict to the case when H is one atom. In fact, for ground atoms p, q it holds $\Sigma \models p \wedge q$ iff $\Sigma \models p$ and $\Sigma \models q$.

To this aim, we introduce for convenience auxiliary extended deductive rules allowing "implications" in the antecedent or in the consequent. This allows to codify formulas (13)-(16) as follows:

13a $\dfrac{(A,\mathtt{sc},B)}{(x,\mathtt{type},A) \xrightarrow{\forall x} (x,\mathtt{type},B)}$ 13b $\dfrac{(x,\mathtt{type},A) \xrightarrow{\forall x} (x,\mathtt{type},B)}{(A,\mathtt{sc},B)}$ (`rdfs:subClassOf`)

14a $\dfrac{(P,\mathtt{sp},Q)}{(x,P,y) \xrightarrow{\forall xy} (x,Q,y)}$ 14b $\dfrac{(x,P,y) \xrightarrow{\forall xy} (x,Q,y)}{(P,\mathtt{sp},Q)}$ (`rdfs:subPropertyOf`)

15a $\dfrac{(P,\mathtt{dom},A)}{(x,P,y) \xrightarrow{\forall xy} (x,\mathtt{type},A)}$ 15b $\dfrac{(x,P,y) \xrightarrow{\forall xy} (x,\mathtt{type},A)}{(P,\mathtt{dom},A)}$ (`rdfs:domain`)

16a $\dfrac{(P,\mathtt{range},A)}{(x,P,y) \xrightarrow{\forall xy} (y,\mathtt{type},A)}$ 16b $\dfrac{(x,P,y) \xrightarrow{\forall xy} (y,\mathtt{type},A)}{(P,\mathtt{range},A)}$ (`rdfs: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 G because of the presence of the *open world assumption*, we cannot know from G if it is valid or not.
2. Two implications can be matched if the meaning of the formulas allow so. For example, $(x, \texttt{type}, A) \xrightarrow{\forall x} (x, \texttt{type}, B)$ and $(y, \texttt{type}, B) \xrightarrow{\forall y} (y, \texttt{type}, C)$ would produce another rule:

$$\frac{(x, \texttt{type}, A) \xrightarrow{\forall x} (x, \texttt{type}, B) \qquad (y, \texttt{type}, B) \xrightarrow{\forall y} (y, \texttt{type}, C)}{(z, \texttt{type}, A) \xrightarrow{\forall z} (z, \texttt{type}, C)} \tag{17}$$

3. The only way to use an implication in a combination of rules is, either:
   (a) to combine it with another implication to derive a third implication (e.g., to form rules of the form (17)). The table in Figure 6 summarises 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). Notice that the only possible relevant formula one could get with this procedure is a formula of the type $\forall x(x, \texttt{type}, A) \rightarrow (x, \texttt{type}, B)$, thus, to deduce a triple of the form $(u, \texttt{sc}, v)$ using rule (13b). Note also that one cannot use the rules (14b), (15b) or (16b), because they need both variables universally quantified.
   (b) To instantiate the implication in the consequent, and using the Deduction Theorem ($p \vdash q \rightarrow r$ iff $p, q \vdash r$). Consider for instance rule (13a); we have: $(A, \texttt{sc}, B) \vdash (x, \texttt{type}, A) \xrightarrow{\forall x} (x, \texttt{type}, B)$. By using the deduction theorem, we obtain: $(A, \texttt{sc}, B)\ (x, \texttt{type}, A) \vdash (x, \texttt{type}, B)$. By systematically applying this process to rules (13a)-(16a), we obtain the rules in the table of Figure 7.
   (c) To use instantiation that make it possible to combine rules. For example the new rule 'Extensional B' directly follows from rule (15a) instantiated with $P = \texttt{type}$, which combined with rule for subclass (13b), gives the implication $\forall x(x, \texttt{type}, y) \rightarrow (x, \texttt{type}, B)$ which using rule (13a) gives $(y, \texttt{sc}, B)$. The table in Figure 8 is about the results of the application of the instantiation-plus-combination operation.

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

Fig. 7. Set of inference rules obtained by instantiating rules (13a)-(16a)

Instantiation/Combination	Rule obtained	Rule in ρdf+	Rule in RDFS
(14a-inst)$\curvearrowright$15a$\curvearrowright$13b	$\frac{(\texttt{type},\texttt{sp},A),(A,\texttt{dom},B)}{(X,\texttt{sc},B)}$	7a	not available
(15a-inst)$\curvearrowright$13b$\curvearrowright$13a	$\frac{(\texttt{type},\texttt{dom},A)}{(X,\texttt{sc},A)}$	8a	not available

Fig. 8. Inference rules obtained by instantiating and combining rules (13a)-(16a)

1. Notice that the only possible relevant formula one could get with the introduced procedure is a formula of the type $\forall x(x, \texttt{type}, A) \rightarrow (x, \texttt{type}, B)$, thus, to deduce a triple of the form $(u, \texttt{sc}, v)$ using rule (13b). Note that one cannot use the other rules (14b), (15b) or (16b), 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 (14a), (15a) or (16a) instantiated with $P = \texttt{type}$. As for Rule (14a), we should instantiate also $y = C$, but in this case the rule will give $\forall x(x, \texttt{type}, C) \rightarrow (x, Q, C)$, whose consequent cannot be further combined unless $Q = \texttt{type}$, which gives nothing. As for rule (15a), it gives our rule Extensional B, while rule (16a) is useless for this argument (notice that in (16a) the y in the implication changes its position from third to first thus making impossible the combination with (13b)).
   (b) Those that end with $(x, \texttt{type}, y)$: here rule (15a) is relevant once y is instantiated to a constant; and rules (15a) and (16a) 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 (14a) weakened to $x = y$. □

We will need a simple and trivial corollary to Lemma 2.

Lemma 3. *For a ρdf graph $\mathcal{G}$: $\mathcal{G} \vdash_{\rho df+} \widehat{cl}(\mathcal{G})$.*

Now we have the proof of the main theorem 1.

Proof.
$\mathcal{G} \models_{\rho\text{df+}} H$
iff $\mathcal{G} \models_{\text{RDFS+}} H$ (by definition of ρdf+)
iff $\mathcal{G} \cup \{\text{axioms } 13-16\} \models_{\text{RDFS+}} H$ (by definition of ρdf+)
iff $\mathcal{G} \cup \{\text{axioms } 13-16\} \models_{\rho\text{df}} H$ (Theorem 5 from [10], because left and right hand sides have only ρdf vocabulary)
iff $\widehat{cl}(\mathcal{G}) \models_{\rho\text{df}} H$ (by Definition 7)
iff $\widehat{cl}(\mathcal{G}) \vdash_{\rho\text{df}} H$ (Theorem 8 from [10], because there is only ρdf vocabulary)
iff $\widehat{cl}(\mathcal{G}) \vdash_{\rho\text{df+}} H$ (by Lemma 2)
iff $\mathcal{G} \vdash_{\rho\text{df+}} H$ (by Lemma 3 and transitivity of $\vdash_{\rho\text{df+}}$). □