RDF and Logic: Reasoning and Extension

Jos de Bruijn
Faculty of Computer Science,
Free University of Bozen-Bolzano, Italy
debruijn@inf.unibz.it

Stijn Heymans
Digital Enterprise Research Institute (DERI),
University of Innsbruck, Austria
stijn.heymans@deri.org

Abstract

In this paper we explore embeddings of the various kinds of RDF entailment in F-Logic. We show that the embeddings of simple, RDF, and RDFS entailment, as well as a large fragment of extensional RDFS entailment, fall in the Datalog fragment of F-Logic, allowing the use of optimization techniques from the area of deductive databases for reasoning with RDF. Using earlier results on the relationship between F-Logic and Description Logics (DLs), we define an embedding of a large fragment of extensional RDFS in a tractable description logic, namely DL-Lite, allowing efficient reasoning over the ontology vocabulary. We show how, using these embeddings, RDFS can be extended with rules and/or general axioms.

1. Introduction

The Resource Description Framework RDF [8], together with RDFS, constitutes the basic language for the semantic Web. The RDF semantics specification [8] defines four increasingly expressive types of entailment, namely simple, RDF, RDFS, and extensional RDFS (eRDFS) entailment[1]. We refer to these kinds of entailment as *entailment regimes*.

The standard knowledge representation and reasoning paradigms of Description Logics (DL) [1] and Logic Programming (LP) [12], which are both based on classical first-order logic, are used on the the semantic Web (e.g. [9, 6]). However, so far, little research has been done into the formal relationships[2] between RDF and the logical languages which are being considered for the semantic Web. In this paper we try to bridge the gap between these formalisms by demonstrating several embeddings of the RDF(S) entailment regimes in logic, and showing how RDF(S) can be extended with (LP) rules and (DL) logical axioms.

[1]Note that the definition of extensional RDFS entailment is not normative.

[2]A notable exception is [4].

We use F-Logic [11], a syntactical extension of standard first-order logic, for our embeddings. It turns out that the attribute value construct in F-Logic is exactly equivalent to the triple construct in RDF, and the typing and subclassing constructs in F-Logic are very close to those in RDF. Additionally, F-Logic, like RDFS, has the possibility of using the same identifier as a class, an instance, or a property identifier.

The contributions of this paper can be summarized as follows.

– In Section 3 we define embeddings of the simple, RDF, RDFS, and eRDFS entailment regimes in F-Logic, and show that these embeddings preserve entailment. It turns out that deductive database technology can be immediately applied for reasoning with the simple, RDF, RDFS, and a subset of the eRDFS entailment regimes. Using earlier results about the relationship between F-Logic and DL/FOL [5] we demonstrate the embedding of a very expressive subset of eRDFS in $DL\text{-}Lite_{\mathcal{R}}$ [2], enabling the use of DL reasoning techniques for eRDFS.

– In Section 4 we use these embeddings in logic, together with existing results about the complexity of reasoning in subsets of logic, to establish several novel complexity results for RDF(S). Table 2 on page 4 summarizes the existing and novel complexity results for RDF.

– Finally, in Section 5 we show how RDF graphs can be extended with F-Logic rules or FOL axioms using the notions of F-Logic and FOL extended RDF graphs, and exhibit several complexity results for reasoning with such extended graphs.

Note that we do not consider RDF literals and datatypes in this paper. We consider this future work.

2. Frame Logic

We follow the treatment of F-Logic[3] in [5]. For the full definition of the F-Logic semantics, we refer the reader

[3]Note that F-Logic is also often used as an extension of nonmonotonic logic programming; however, we follow the original definition which is strictly first-order.

to [11, 5].

The signature of an F-language $\mathcal{L}$ is of the form $\Sigma = \langle \mathcal{F}, \mathcal{P} \rangle$ with $\mathcal{F}$ and $\mathcal{P}$ disjoint sets of function and predicate symbols, each with an associated arity $n \geq 0$. Terms and atomic formulas are defined in the usual way. A molecule in F-Logic is one of the following statements: (i) an *is-a* assertion of the form $C : D$, (ii) a *subclass-of* assertion of the form $C :: D$, or (iii) a data molecule of the form $C[D \twoheadrightarrow E]$, with C, D, E terms. An F-Logic molecule is *ground* if it does not contain variables.

Formulas of an F-language $\mathcal{L}$ are either atomic formulas, molecules, or compound formulas, which are constructed in the usual way from atomic formulas, molecules, and the logical connectives $\neg, \wedge, \vee, \supset$, the quantifiers $\exists, \forall$ and the auxiliary symbols), (. The Horn and Datalog subset of F-Logic are defined in the usual way (see e.g. [5]).

An *F-structure* is a tuple $\mathbf{I} = \langle U, \prec_U, \in_U, \mathbf{I}_F, \mathbf{I}_P, \mathbf{I}_{\twoheadrightarrow} \rangle$. Here, $\prec_U$ is an irreflexive partial order on the domain U and $\in_U$ is a binary relation over U. We write $a \preceq_U b$ when $a \prec_U b$ or $a = b$, for $a, b \in U$. For each F-structure must hold that if $a \in_U b$ and $b \preceq_U c$ then $a \in_U c$. An n-ary function symbol $f \in F$ is interpreted as a function over the domain U: $\mathbf{I}_F(f) : U^n \rightarrow U$. An n-ary predicate symbol $p \in P$ is interpreted as a relation over the domain U: $\mathbf{I}_P(p) \subseteq U^n$. $\mathbf{I}_{\twoheadrightarrow}$ associates a binary relation over U with each $k \in U$: $\mathbf{I}_{\twoheadrightarrow}(k) \subseteq U \times U$. Variable assignments are defined as usual.

Given an F-structure $\mathbf{I}$, a variable assignment B, and a term t of $\mathcal{L}$, $t^{\mathbf{I},B}$ is defined as: $x^{\mathbf{I},B} = x^B$ for variable symbol x and $t^{\mathbf{I},B} = \mathbf{I}_F(f)(t_1^{\mathbf{I},B}, \ldots, t_n^{\mathbf{I},B})$ for t of the form $f(t_1, \ldots, t_n)$.

Satisfaction of atomic formulas and molecules ϕ in $\mathbf{I}$, given the variable assignment B, denoted $(\mathbf{I}, B) \models_{\mathsf{f}} \phi$, is defined as: $(\mathbf{I}, B) \models_{\mathsf{f}} p(t_1, \ldots, t_n)$ iff $(t_1^{\mathbf{I},B}, \ldots, t_n^{\mathbf{I},B}) \in \mathbf{I}_P(p)$, $(\mathbf{I}, B) \models_{\mathsf{f}} t_1 : t_2$ iff $t_1^{\mathbf{I},B} \in_U t_2^{\mathbf{I},B}$, $(\mathbf{I}, B) \models_{\mathsf{f}} t_1 :: t_2$ iff $t_1^{\mathbf{I},B} \preceq_U t_2^{\mathbf{I},B}$, $(\mathbf{I}, B) \models_{\mathsf{f}} t_1[t_2 \twoheadrightarrow t_3]$ iff $\langle t_1^{\mathbf{I},B}, t_3^{\mathbf{I},B} \rangle \in \mathbf{I}_{\twoheadrightarrow}(t_2^{\mathbf{I},B})$, and $(\mathbf{I}, B) \models_{\mathsf{f}} t_1 = t_2$ iff $t_1^{\mathbf{I},B} = t_2^{\mathbf{I},B}$. This extends to arbitrary formulas in the usual way.

The notions of a model and of validity are defined as usual. A theory $\Phi \subseteq \mathcal{L}$ *F-entails* a formula $\phi \in \mathcal{L}$, denoted $\Phi \models_{\mathsf{f}} \phi$, iff for all F-structures $\mathbf{I}$ such that $\mathbf{I} \models_{\mathsf{f}} \Phi, \mathbf{I} \models_{\mathsf{f}} \phi$.

Classical first-order logic (classical FOL) is F-Logic without molecules. *Contextual first-order logic* (contextual FOL) is classical FOL where $\mathcal{F}$ and $\mathcal{P}$ are not required to be disjoint, function and predicate symbols do not have an associated arity, and for every structure $\mathbf{I} = \langle U, \prec_U, \in_U, \mathbf{I}_F, \mathbf{I}_P, \mathbf{I}_{\twoheadrightarrow} \rangle$, $\mathbf{I}_F$ assigns a function $\mathbf{I}_F(f) : U^n \rightarrow U$ to every $f \in \mathcal{F}$ for every nonnegative integer n and $\mathbf{I}_P$ assigns a relation $\mathbf{I}_P(p) \subseteq U^n$ to every $p \in \mathcal{P}$ for every nonnegative integer n. We denote satisfaction and entailment in classical and contextual FOL using the symbols $\models$ and $\models_c$, respectively.

3. RDF Embedding and Extension

In this section we first consider an embedding of the RDF entailment regimes in F-Logic, after which we are consider embeddings of the eRDFS entailment regime in FOL and DL. For the definition of the different entailment regimes we refer the reader to [8].

Let S, E be RDF graphs, $x \in \{s, rdf, rdfs, erdfs\}$ be the simple (resp., RDF, RDFS, eRDFS) entailment regime, we denote entailment, i.e. S x-entails E, with $S \models_x E$.

3.1. Embedding RDF in F-Logic

We first define the embedding of RDF graphs in F-Logic, without taking into account the specific entailment regime, using an embedding function tr. The RDF graph is translated to a conjunction of data molecules, where URIs are constants, and blank nodes are existentially quantified variables. In the remainder, we assume that every RDF graph is finite. Given a triple $\langle s, p, o \rangle$ (resp., graph S), $bl(\langle s, p, o \rangle)$ (resp., $bl(S)$) denotes the set of blank nodes occurring in the triple (resp., graph).

Definition 1. *Let $\langle s, p, o \rangle$ be a triple and S a graph.*

$$\begin{aligned} tr(\langle s, p, o \rangle) &= s[p \twoheadrightarrow o] \\ tr(S) &= \exists\, bl(S)(\bigwedge \{tr(\langle s, p, o \rangle) \mid \langle s, p, o \rangle \in S\}) \end{aligned}$$

Depending on the entailment regime x, we add a set of formulas Ψ^x to the embedding of the graph. Ψ^x, defined in Table 1, axiomatizes the semantics of the entailment regime x.[4]

If ϕ is an F-Logic formula in prenex normal form with only existential quantifiers, then ϕ^{sk} denotes the *Skolemization* of ϕ, i.e. every existentially quantified variable is replaced by a new constant not occurring in the formula or its context (the theory in which it occurs, or any of the surrounding theories, e.g. those participating in an entailment relation). If Φ is an F-Logic theory, then Φ^{sk} denotes a skolemization of Φ.

The following Proposition follows immediately from the definition of the translations.

Proposition 1. *Let S be an RDF graph. Then, $tr(S)^{sk} \cup \Psi^x$, with $x \in \{s, rdf, rdfs\}$, can be equivalently rewritten to a set of F-Logic Datalog formulas.*

Note that Ψ^{erdfs} cannot be equivalently rewritten to a set of Datalog formulas, due to the universal quantification in the antecedents of the implications in Ψ^{erdfs}.

We now show the correspondence between entailment in the original RDF semantics and entailment in the F-Logic embedding.

[4] For brevity, we leave out the namespace of the RDF vocabulary; for example, `type` is short for `rdf:type`.

$\Psi^{s} = \emptyset$
$\Psi^{rdf} = \Psi^{s} \cup \{tr(\langle s,p,o\rangle) \mid \langle s,p,o\rangle \text{ is an RDF axiomatic triple}\} \cup$
$\{\forall x(\exists y,z(y[x \twoheadrightarrow z]) \supset x[\mathtt{type} \twoheadrightarrow \mathtt{Property}])\}$
$\Psi^{rdfs} = \Psi^{rdf} \cup \{tr(\langle s,p,o\rangle) \mid \langle s,p,o\rangle \text{ is an RDFS axiomatic triple}\} \cup$
$\{\forall x,y,z(x[y \twoheadrightarrow z] \supset x[\mathtt{type} \twoheadrightarrow \mathtt{Resource}] \wedge z[\mathtt{type} \twoheadrightarrow \mathtt{Resource}]),$
$\forall u,v,x,y(x[\mathtt{domain} \twoheadrightarrow y] \wedge u[x \twoheadrightarrow v] \supset u[\mathtt{type} \twoheadrightarrow y]),$
$\forall u,v,x,y(x[\mathtt{range} \twoheadrightarrow y] \wedge u[x \twoheadrightarrow v] \supset v[\mathtt{type} \twoheadrightarrow y]),$
$\forall x(x[\mathtt{type} \twoheadrightarrow \mathtt{Property}] \supset x[\mathtt{subPropertyOf} \twoheadrightarrow x]),$
$\forall x,y,z(x[\mathtt{subPropertyOf} \twoheadrightarrow y] \wedge y[\mathtt{subPropertyOf} \twoheadrightarrow z] \supset$
$x[\mathtt{subPropertyOf} \twoheadrightarrow z]),$
$\forall x,y(x[\mathtt{subPropertyOf} \twoheadrightarrow y] \supset x[\mathtt{type} \twoheadrightarrow \mathtt{Property}] \wedge$
$y[\mathtt{type} \twoheadrightarrow \mathtt{Property}] \wedge \forall z_1,z_2(z_1[x \twoheadrightarrow z_2] \supset z_1[y \twoheadrightarrow z_2])),$
$\forall x(x[\mathtt{type} \twoheadrightarrow \mathtt{Class}] \supset x[\mathtt{subClassOf} \twoheadrightarrow \mathtt{Resource}]),$
$\forall x,y(x[\mathtt{subClassOf} \twoheadrightarrow y] \supset x[\mathtt{type} \twoheadrightarrow \mathtt{Class}] \wedge$
$y[\mathtt{type} \twoheadrightarrow \mathtt{Class}] \wedge \forall z(z[\mathtt{type} \twoheadrightarrow x] \supset z[\mathtt{type} \twoheadrightarrow y])),$
$\forall x(x[\mathtt{type} \twoheadrightarrow \mathtt{Class}] \supset x[\mathtt{subClassOf} \twoheadrightarrow x]),$
$\forall x,y,z(x[\mathtt{subClassOf} \twoheadrightarrow y] \wedge y[\mathtt{subClassOf} \twoheadrightarrow z] \supset$
$x[\mathtt{subClassOf} \twoheadrightarrow z]),$
$\forall x(x[\mathtt{type} \twoheadrightarrow \mathtt{ContainerMembershipProperty}] \supset$
$x[\mathtt{subPropertyOf} \twoheadrightarrow \mathtt{member}])\}$
$\Psi^{erdfs} = \Psi^{rdfs} \cup \{\forall x,y(\forall u,v(u[x \twoheadrightarrow v] \supset u[\mathtt{type} \twoheadrightarrow y]) \supset$
$x[\mathtt{domain} \twoheadrightarrow y]),$
$\forall x,y(\forall u,v(u[x \twoheadrightarrow v] \supset v[\mathtt{type} \twoheadrightarrow y]) \supset x[\mathtt{range} \twoheadrightarrow y]),$
$\forall x,y(x[\mathtt{type} \twoheadrightarrow \mathtt{Property}] \wedge y[\mathtt{type} \twoheadrightarrow \mathtt{Property}] \wedge$
$\forall u,v(u[x \twoheadrightarrow v] \supset u[y \twoheadrightarrow v]) \supset x[\mathtt{subPropertyOf} \twoheadrightarrow y]),$
$\forall x,y(x[\mathtt{type} \twoheadrightarrow \mathtt{Class}] \wedge y[\mathtt{type} \twoheadrightarrow \mathtt{Class}] \wedge$
$\forall u(u[\mathtt{type} \twoheadrightarrow x] \supset u[\mathtt{type} \twoheadrightarrow y]) \supset x[\mathtt{subClassOf} \twoheadrightarrow y])\}$

Table 1. Axiomatization of the RDF semantics

Theorem 1. *Let S, E be RDF graphs and $x \in \{s, rdf, rdfs, erdfs\}$ an entailment regime. Then,*

$$S \models_x E \text{ if and only if } tr(S) \cup \Psi^x \models_{\mathsf{f}} tr(E).$$

The following corollary follows immediately from Theorem 1 and the classical results about Skolemization.

Corollary 1. *Let S, E be RDF graphs and $x \in \{s, rdf, rdfs, erdfs\}$ be an entailment regime. Then,*

$$S \models_x E \text{ if and only if } tr(S)^{sk} \cup \Psi^x \models_{\mathsf{f}} tr(E).$$

Since, by Proposition 1, $tr(S)^{sk}$, $tr(S)^{sk} \cup \Psi^{rdf}$ and $tr(S)^{sk} \cup \Psi^{rdfs}$ are equivalent to sets of Horn formulas, this result implies that simple, RDF, and RDFS entailment can be computed using existing F-Logic rule reasoners such as FLORA-2, and Ontobroker, as well as other rule reasoners[5]. Notice that, in the corollary, $tr(E)$ can be seen as a boolean conjunctive query (i.e. a yes/no query) and the existentially quantified variables (blank nodes) in $tr(E)$ are the non-distinguished variables.

The final embedding in F-Logic we consider is a direct embedding of the extensional RDFS semantics which eliminates part of the RDFS vocabulary, yielding a set of Horn formulas. We first define the notion of *nonstandard use* of the RDFS vocabulary. Nonstandard use of the RDFS vocabulary intuitively corresponds to using the vocabulary in locations where it has not been intended, for example $\langle \mathtt{type}, \mathtt{subPropertyOf}, a\rangle$.

We say that a term t occurs in a property position if it occurs as the predicate of a triple, as the subject or object of a `subPropertyOf` triple, as the subject of a `domain` or `range` triple, or as the subject in a triple $\langle t, \mathtt{type}, \mathtt{Property}\rangle$ or $\langle t, \mathtt{type}, \mathtt{ContainerMembershipProperty}\rangle$. A term t occurs in a class position if it occurs as the subject or object of a `subClassOf` triple, as the object of a `domain`, `range`, or `type` triple, or as the subject of a triple $\langle t, \mathtt{type}, \mathtt{Class}\rangle$.

Definition 2. *Let S be an RDF graph. Then S has* nonstandard use *of the RDFS vocabulary if* `type`*,* `subClassOf`*,* `domain`*,* `range`*, or* `subPropertyOf` *occurs in a non-property position in S, or* `ContainerMembershipProperty`*,* `Resource`*,* `Class`*, or* `Property` *occurs in S.*

Definition 3. *Let $\langle s,p,o\rangle$ be an RDF triple, then*

$$tr^{erdfs}(\langle s, \mathtt{type}, o\rangle) = s : o,$$
$$tr^{erdfs}(\langle s, \mathtt{subClassOf}, o\rangle) = \forall x(x : s \supset x : o),$$
$$tr^{erdfs}(\langle s, \mathtt{subPropertyOf}, o\rangle) = \forall x,y(x[s \twoheadrightarrow y] \supset x[o \twoheadrightarrow y]),$$
$$tr^{erdfs}(\langle s, \mathtt{domain}, o\rangle) = \forall x,y(x[s \twoheadrightarrow y] \supset x : o),$$
$$tr^{erdfs}(\langle s, \mathtt{range}, o\rangle) = \forall x,y(x[s \twoheadrightarrow y] \supset y : o), \text{ and}$$
$$tr^{erdfs}(\langle s,p,o\rangle) = s[p \twoheadrightarrow o], \text{ otherwise.}$$

Let S be an RDF graph. Then,

$$tr^{erdfs}(S) = \exists\, bl(S)(\bigwedge\{tr^{erdfs}(\langle s,p,o\rangle) \mid \langle s,p,o\rangle \in S\} \cup \{tr^{erdfs}(\langle s,p,o\rangle) \mid \langle s,p,o\rangle \text{ is an RDFS axiomatic triple with no nonstandard use of the RDFS vocabulary}\})$$

Theorem 2. *Let S, E be RDF graphs with no nonstandard use of the RDFS vocabulary. Then,*

$$S \models_{erdfs} E \text{ iff } tr^{erdfs}(S) \models_{\mathsf{f}} tr^{erdfs}(E).$$

Furthermore, $(tr^{erdfs}(S))^{sk}$ is a conjunction of F-Logic Datalog formulas.
If, additionally, E does not contain `subClassOf`*,* `domain`*,* `range`*, or* `subPropertyOf`*, then $tr^{erdfs}(E)$ is a conjunction of atomic molecules with an existential prefix, and*

$$S \models_{erdfs} E \text{ iff } (tr^{erdfs}(S))^{sk} \models_{\mathsf{f}} tr^{erdfs}(E).$$

Since $(tr^{erdfs}(S))^{sk}$ is a set of Datalog formulas, we have that, if the RDF graphs fulfills certain (natural) conditions, query answering techniques from the area of deductive databases can be used for checking eRDFS entailment.

3.2. Embedding Extensional RDFS in First-Order Logic

An F-Logic theory Φ is *translatable* to contextual FOL if it has no :: molecules and for molecules of the forms $t_1[t_2 \twoheadrightarrow t_3]$ and $t_1 : t_2$ holds that t_2 is a constant symbol.

[5]Note that the attribute value construct $a[b \twoheadrightarrow c]$ is the only construct specific to F-Logic which is used in the embeddings. Since it does not carry any specific semantics, it may be straightforwardly embedded using a ternary predicate $attval(a,b,c)$. Notice also that all rules are safe, and thus Datalog engines may be used.

Let Φ be an F-Logic theory which is translatable to contextual FOL, then $(\Phi)^{FO}$ is the contextual FOL theory obtained from Φ by:
– replacing every $t_1[t_2 \twoheadrightarrow t_3]$ with $t_2(t_1, t_3)$, and
– replacing every $t_1 : t_2$ with $t_2(t_1)$.

The following is a straightforward generalization of a result in [5].

Proposition 2. *Let Φ (resp., ϕ) be an equality-free F-Logic theory (resp., formula) which is translatable to contextual FOL. Then,* $\Phi \models_{\mathsf{f}} \phi$ *iff* $(\Phi)^{FO} \models_c (\phi)^{FO}$.

An RDF graph S is a *non-higher-order* RDF graph if S does not contain blank nodes in class or property positions and does not contain nonstandard use of the RDFS vocabulary. A non-higher order RDF graph S is a *classical* RDF graph if the sets of URIs occurring in class and property positions in S (and its context, e.g. entailing or entailed graph) are mutually disjoint, and disjoint with the sets of all URIs not occurring in class or property positions in S (and its context).

Theorem 3. *Let S, E be non-higher-order (resp., classical) RDF graphs. Then, $(tr^{erdfs}(S))^{FO}$, $(tr^{erdfs}(E))^{FO}$ are theories of contextual (resp., classical) FOL and $S \models_{erdfs} E$ iff $(tr^{erdfs}(S))^{FO} \models_c (tr^{erdfs}(E))^{FO}$ (resp., $(tr^{erdfs}(S))^{FO} \models (tr^{erdfs}(E))^{FO}$).*

4. Complexity of RDF

The complexity of simple and RDFS entailment is well known, and the complexity of RDF and extensional RDFS entailment follow immediately. Note that, although the set of axiomatic triples is infinite, only a finite subset needs to be taken into account when checking the entailment.

Proposition 3 ([7, 10, 4])**.** *Let S, E be RDF graphs, then the problems $S \models_s E$, $S \models_{rdf} E$, and $S \models_{rdfs} E$ are* NP-complete *in the combined size of S and E, and polynomial in the size of S. If E is ground, then the respective problems are in* P*. Additionally, the problem $S \models_{erdfs} E$ is* NP*-hard.*

From the embedding in F-Logic, together with the complexity of nonrecursive Datalog, we obtained the following novel characterization of the complexity of simple and RDF entailment.

Proposition 4. *Let S, E be RDF graphs. Then, the problems $S \models_s E$ and $S \models_{rdf} E$ are in* LogSpace *with respect to the size of S, and with respect to the combined size of the graphs if E is ground.*

By the correspondence between FOL and Description Logics and earlier complexity results for the Description Logic $DL\text{-}Lite_{\mathcal{R}}$ [2] we obtain the following results.

Theorem 4. *Let S, E be RDF graphs with no nonstandard use of the RDFS vocabulary. Then, the problem of deciding $S \models_{erdfs} E$ is* NP-complete *in the size of the graphs, and polynomial if E is ground.*

Regime	Restrictions on S	Restrictions on E	Complexity
$x \in \{s, rdf, rdfs\}$	none	none	NP-complete
$x \in \{s, rdf\}$	none	ground	LogSpace
$x \in \{rdfs\}$	none	ground	P
$x \in \{erdfs\}$	none	none	NP-hard
$x \in \{erdfs\}$	no nonst. RDFS	no nonst. RDFS	NP-complete
$x \in \{erdfs\}$	no nonst. RDFS	ground, no nonst. RDFS	P

Table 2. Complexity of Entailment $S \models_x E$, measured in the combined size of S and E

Table 2 summarizes the complexity of the different entailment regimes; "No nonst. RDFS" stands for "no nonstandard use of the RDFS vocabulary". The results in the first and third line of the Table were obtained in [7, 4, 10]. To the best of our knowledge, the other results in the table are novel.

5. RDF Extensions

In this section we consider extensions of RDF graphs with logical rules and general theories.

Definition 4. *An F-Logic extended RDF graph is a tuple $eS = \langle S, \Phi, x\rangle$, with S an RDF graph, Φ an F-Logic theory, and $x \in \{s, rdf, rdfs, erdfs\}$ an entailment regime.*

eS is satisfiable (resp., valid) if $tr(S) \cup \Psi^x \cup \Phi$ is satisfiable (resp., valid), and eS entails an F-Logic formula ϕ (resp., RDF graph E), denoted $eS \models \phi$ (resp., $eS \models E$), if $tr(S) \cup \Psi^x \cup \Phi \models_{\mathsf{f}} \phi$ (resp., $tr(S) \cup \Psi^x \cup \Phi \models_{\mathsf{f}} tr(E)$).

The following proposition follows immediately from Theorem 1.

Proposition 5. *Let S, E be RDF graphs, and $x \in \{s, rdf, rdfs, erdfs\}$ an entailment regime. Then,*

$$\langle S, \emptyset, x\rangle \models E \text{ iff } S \models_x E.$$

Considering such F-Logic extended RDF graphs, there is a discrepancy between the RDF and F-Logic constructs used for asserting class membership ($a[\texttt{type} \twoheadrightarrow C]$ vs. $a : C$) and asserting the subclass relation ($A[\texttt{subClassOf} \twoheadrightarrow B]$ vs. $A :: B$). Therefore, the interaction between the RDF graph and the F-Logic theory might not be as expected.

Consider, for example, the RDF graph $S = \{\langle A, \texttt{subClassOf}, B\rangle\}$ and the F-Logic theory $\Phi =$

$\{a:A\}$. Consider now the F-Logic extended RDF graph $T = \langle S, \Phi, rdfs\rangle$. One might intuitively expect $T \models a:B$. This is, however, not the case, because of the lack of interaction between the RDFS vocabulary and the F-Logic language constructs.

We overcome this limitation by using the so-called *RDF interaction axioms*:

$$\Psi^{RIA} = \{ \quad \forall x, y(x[\mathtt{type} \twoheadrightarrow y] \supset x:y), \\ \forall x, y(x[\mathtt{subClassOf} \twoheadrightarrow y] \supset x::y)\}.$$

Definition 5. *An F-Logic extended RDF graph eS is RIA-satisfiable (resp., valid) if $tr(S) \cup \Psi^x \cup \Phi \cup \Psi^{RIA}$ is satisfiable (resp., valid), and eS RIA-entails an F-Logic formula ϕ (resp., RDF graph E), denoted $eS \models_{RIA} \phi$ (resp., $eS \models_{RIA} E$), if $tr(S) \cup \Psi^x \cup \Phi \cup \Psi^{RIA} \models_{\mathsf{f}} \phi$ (resp., $tr(S) \cup \Psi^x \cup \Phi \cup \Psi^{RIA} \models_{\mathsf{f}} tr(E)$).*

The following proposition follows from Proposition 5 and the structure of the RIA axioms.

Proposition 6. *Let S, E be RDF graphs, and $x \in \{s, rdf, rdfs, erdfs\}$ an entailment regime. Then,*

$$\langle S, \emptyset, x\rangle \models_{RIA} E \text{ iff } S \models_x E.$$

Theorem 3 sanctions the extension of a subset of eRDFS with DL or FOL axioms:

Definition 6. *A contextual (resp., classical) FOL-extended RDF graph is a tuple $\langle S, \Phi\rangle$ where S is a non-higher-order (resp., classical) RDF graph, and Φ is a contextual (resp., classical) FOL theory.*

$\langle S, \Phi\rangle$ is satisfiable (resp., valid) if $(tr^{erdfs}(S))^{FO} \cup \Phi$ is satisfiable (resp., valid).

A contextual FOL extended RDF graph $\langle S, \Phi\rangle$ entails a contextual FOL formula ϕ (resp., non-higher-order RDF graph E) if $(tr^{erdfs}(S))^{FO} \cup \Phi \models_c \phi$ (resp., $(tr^{erdfs}(S))^{FO} \cup \Phi \models_c (tr^{erdfs}(E))^{FO}$).

A classical FOL extended RDF graph $\langle S, \Phi\rangle$ entails a classical FOL formula ϕ (resp., classical RDF graph E) if $(tr^{erdfs}(S))^{FO} \cup \Phi \models \phi$ (resp., $(tr^{erdfs}(S))^{FO} \cup \Phi \models (tr^{erdfs}(E))^{FO}$).

Proposition 7. *Let S, E be non-higher-order (resp., classical) RDF graphs. Then, $\langle S, \emptyset\rangle \models_c E$ (resp., $\langle S, \emptyset\rangle \models E$) iff $S \models_{erdfs} E$.*

The following results about the complexity of reasoning with extended RDF graphs follow immediately from the complexity results obtained in the previous section, and the complexity of the considered extensions.

We first consider RDF graphs extended with F-Logic Datalog rules.

Theorem 5. *Given an F-Logic extended RDF graph $eS = \langle S, \Phi, x\rangle$, with Φ a set of F-Logic Datalog rules, and a ground atom or molecule α, then*

- *if $x \in \{s, rdf, rdfs\}$, the problem of deciding $eS \models_{RIA} \alpha$ is polynomial, and*
- *if $x = erdfs$ and S has no nonstandard use of the RDFS vocabulary, the problem of deciding $eS \models \alpha$ is polynomial.*

We now consider RDF graphs, under the eRDFS entailment regime, extended with $DL\text{-}Lite_{\mathcal{R}}$ axioms. $DL\text{-}Lite_{\mathcal{R}}$ [2] is a Description Logic which largely subsumes the expressiveness of extensional RDFS, and for which most of the reasoning tasks are tractable (i.e. polynomial). We consider $DL\text{-}Lite_{\mathcal{R}}$ as defined in [2], and refer to this as *classical* $DL\text{-}Lite_{\mathcal{R}}$. We also consider a variant of $DL\text{-}Lite_{\mathcal{R}}$ in which the sets of class, rule, and individual identifiers are not disjoint, and refer to this has *contextual* $DL\text{-}Lite_{\mathcal{R}}$ (cf. contextual FOL).

Theorem 6. *Let $eS = \langle S, \Phi\rangle$ be a contextual (resp., classical) FOL extended RDF graph, such that S is ground, and Φ is the FOL equivalent of a contextual $DL\text{-}Lite_{\mathcal{R}}$ knowledge base $\mathcal{K}$, and let Φ' be the FOL equivalent of a contextual (resp., classical) $DL\text{-}Lite_{\mathcal{R}}$ knowledge base $\mathcal{K}'$. Then, the problem $eS \models_c \Phi'$ (resp., $eS \models \Phi'$) is polynomial.*

References

[1] F. Baader et al., editors. *The Description Logic Handbook*. Cambridge University Press, 2003.

[2] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. In *KR 2006*.

[3] E. Dantsin, T. Eiter, G. Gottlob, and A. Voronkov. Complexity and expressive power of logic programming. *ACM Comp. Surv.*, 33(3):374–425, September 2001.

[4] J. de Bruijn, E. Franconi, and S. Tessaris. Logical reconstruction of normative RDF. In *OWLED-2005*.

[5] J. de Bruijn and S. Heymans. Translating ontologies from predicate-based to frame-based languages. In *RuleML-2006*.

[6] J. de Bruijn, H. Lausen, A. Polleres, and D. Fensel. The web service modeling language: An overview. In *ESWC2006*.

[7] C. Gutierrez, C. Hurtado, and A. O. Mendelzon. Foundations of semantic web databases. In *PODS2004*.

[8] P. Hayes. RDF semantics. W3C Recommendation, 10 February 2004.

[9] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26, 2003.

[10] H. J. ter Horst. Completeness, decidability and complexity of entailment for rdf schema and a semantic extension involving the owl vocabulary. *J. Web Sem.*, 3(2–3):79–115, 2005.

[11] M. Kifer, G. Lausen, and J. Wu. Logical foundations of object-oriented and frame-based languages. *JACM*, 42(4):741–843, 1995.

[12] J. W. Lloyd. *Foundations of Logic Programming (2nd edition)*. Springer-Verlag, 1987.