Logical Foundations of (e)RDF(S): Complexity and Reasoning

Jos de Bruijn

Faculty of Computer Science
Free University of Bozen-Bolzano, Italy
`debruijn AT inf.unibz.it`

DERI – 23 May 2008

Joint work with Stijn Heymans

The problem

- Difference between RDFS and classical semantics (OWL DL) has caused a rift in the Semantic Web representation and reasoning community ⇒ Undesirable
- Mistake should not be repeated in a logic-based Semantic Web rules language (RIF)

- Objective 1: bring RDFS and OWL DL together ⇒ Hard
- Objective 2: make sure the Semantic Web rules language is compatible with RDFS ⇒ Doable

- Step 1 (this paper): explore relationship between RDFS and Logic through embeddings (⇒ using logic for reasoning with RDFS; several novel complexity results)

Overview

- 6 entailment regimes (simple, RDF, RDFS, extensional RDFS, D*, D)
- Embed in (first-order) F-Logic
  - Rule-based reasoning with simple, RDF, RDFS, and part of extensional RDFS entailment
  - Novel complexity results for simple, RDF, and part of extensional RDFS entailment
- Embed F-Logic in contextual $\mathit{DL\text{-}Lite}_{\mathcal{R}}$ (not in the presentation)
  - DL-based reasoning with part of extensional RDFS entailment
  - Novel complexity result for extensional RDFS

Contents

F-Logic Syntax

- Atomic formulas are $\top, \bot$, predicate formulas (e.g. $p(a), r(x, y)$), or *molecules*:
  - an *is-a* assertion $C : D$ (e.g. $john : Person, x : y$)
  - a data molecule $C[D \twoheadrightarrow E]$ (e.g. $john[hasChild \twoheadrightarrow mary], x[y \twoheadrightarrow z]$)
- Compound formulas are constructed in the usual way using logical connectives ($\neg, \wedge, \vee, \supset$), quantifiers ($\forall, \exists$), parentheses.

$$\forall x(x : s \supset x : o)$$
$$\forall x(x : Person \supset \exists y(x[hasFather \twoheadrightarrow y]))$$

F-Logic Semantics

Proposition

Given an F-Logic theory and formula Φ *and* ϕ *which do not contain isa and data,* Φ' *and* ϕ' *are obtained by*

- *replacing every is-a molecule* $a : b$ *with* $isa(a, b)$ *and*
- *every data molecule* $a[b \twoheadrightarrow c]$ *with* $data(a, b, c)$*.*

Then,

- *the models of* Φ *and* Φ' *are isomorphic and*
- $\Phi \models_{\mathsf{f}} \phi$ *iff* $\Phi' \models \phi'$*.*

$\Rightarrow$ all known results about reasoning with FOL immediately apply to F-Logic (e.g. complexity of Datalog)

Embedding Simple Entailment

Definition

Let $\langle s, p, o\rangle$ be a triple, S an RDF graph, and $b_1, \ldots, b_n$ be the blank nodes in S.

$$\begin{aligned} tr(\langle s,p,o\rangle) &= s[p \twoheadrightarrow o] \\ tr(S) &= \exists\, b_1,\ldots,b_n \big(\bigwedge \{ tr(\langle s,p,o\rangle) \mid \langle s,p,o\rangle \in S \} \big) \end{aligned}$$

$$S = \{\langle \_{:}x, \texttt{domain}, \mathit{Person}\rangle, \langle \mathit{mother}, \texttt{subPropertyOf}, \_{:}x\rangle\}$$

$$\begin{aligned} tr(S) = \exists \_{:}x(\_{:}x[\texttt{domain} \twoheadrightarrow \mathit{Person}] \wedge \\ \mathit{mother}[\texttt{subPropertyOf} \twoheadrightarrow \_{:}x) \end{aligned}$$

Theorem

S simple-entails E if and only if $tr(S) \models_{\mathsf{f}} tr(E)$

Embedding RDF, RDFS, extensional RDFS

RDF, RDFS, extensional RDFS semantics are axiomatized using the sets of formulas $\Psi^{rdf}, \Psi^{rdfs}, \Psi^{erdfs}$:

$$\boxed{\begin{array}{l} \Psi^{rdf} = \{tr(\langle s,p,o\rangle) \mid \langle s,p,o\rangle \text{ is an RDF axiomatic triple}\} \cup \\ \quad \{wellxml(t) \mid t \text{ is a well-typed XML literal in } S\} \cup \\ \quad \{\forall x(\exists y,z(y[x \twoheadrightarrow z]) \supset x[\mathtt{type} \twoheadrightarrow \mathtt{Property}]), \\ \quad \ldots \end{array}}$$

$$\boxed{\begin{array}{l} \Psi^{rdfs} = \Psi^{rdf} \cup \{tr(\langle s,p,o\rangle) \mid \langle s,p,o\rangle \text{ is an RDFS axiomatic triple}\} \cup \\ \quad \{pl(t) \mid t \text{ is a plain literal in } S\} \cup \\ \quad \forall u,v,x,y(x[\mathtt{domain} \twoheadrightarrow y] \wedge u[x \twoheadrightarrow v] \supset u[\mathtt{type} \twoheadrightarrow y]), \\ \quad \ldots \end{array}}$$

$$\boxed{\begin{array}{l} \Psi^{erdfs} = \Psi^{rdfs} \cup \{\forall x,y(\forall u,v(u[x \twoheadrightarrow v] \supset u[\mathtt{type} \twoheadrightarrow y]) \supset \\ \quad\quad x[\mathtt{domain} \twoheadrightarrow y]), \\ \ldots \end{array}}$$

Embedding RDF, RDFS, extensional RDFS (II)

Theorem (Satisfiability)

- *S is rdf-satisfiable if and only if* $tr(S) \cup \Psi^{rdf}$ *has a model*
- *S is rdfs-satisfiable if and only if* $tr(S) \cup \Psi^{rdfs}$ *has a model*
- *S is erdfs-satisfiable if and only if* $tr(S) \cup \Psi^{erdfs}$ *has a model*

Theorem (Entailment)

- *S rdf-entails E if and only if* $tr(S) \cup \Psi^{rdf} \models_{\mathsf{f}} tr(E)$
- *S rdfs-entails E if and only if* $tr(S) \cup \Psi^{rdfs} \models_{\mathsf{f}} tr(E)$
- *S erdfs-entails E if and only if* $tr(S) \cup \Psi^{erdfs} \models_{\mathsf{f}} tr(E)$

Rule-Based Reasoning

Φ^{sk} denotes *skolemization* of Φ: replace every existential variable with new constant

Corollary

- *S simple-entails E if and only if* $tr(S)^{sk} \models_{\mathrm{f}} tr(E)$
- *S rdf-entails E if and only if* $tr(S)^{sk} \cup \Psi^{rdf} \models_{\mathrm{f}} tr(E)$
- *S rdfs-entails E if and only if* $tr(S)^{sk} \cup \Psi^{rdfs} \models_{\mathrm{f}} tr(E)$

Proposition

$tr(S)^{sk}, \Psi^{rdf}, \Psi^{rdfs}$ *can be equivalently rewritten to sets of F-Logic Datalog formulas.*

⇒ simple, RDF, and RDFS entailment can be computed using existing Datalog reasoners

Direct Embedding of eRDFS

Definition

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

Restrictions on the RDF Graphs

Definition

S has *nonstandard use* of the RDFS vocabulary if

- `type`, `subClassOf`, `domain`, `range` or `subPropertyOf` occurs in the subject or object position of a triple in S or
- `ContainerMembershipProperty`, `Resource`, `Class`, `Datatype` or `Property` occurs in any position other than the object position of a `type`-triple in S.

e.g. $\langle \texttt{type}, \texttt{subPropertyOf}, a\rangle$

Faithfulness of Direct Embedding

Theorem

Let S, E *be RDF graphs with no nonstandard use of the RDFS vocabulary such that* `Resource`, `Class`, `Property`, `ContainerMembershipProperty` *and* `Datatype` *do not occur in* E. *Then,*

- *whenever the class and property vocabularies in* E *are subsets of those in* S,

$$S \models_{erdfs} E \text{ iff } tr^{erdfs}(S) \cup \Psi^{erdfs-V} \models_{\mathsf{f}} tr^{erdfs}(E);$$

- $(tr^{erdfs}(S))^{sk}$ *is a set of F-Logic Datalog formulas.*

⇒ under certain conditions, rule-based reasoning can be done with eRDFS entailment

Complexity of RDF Reasoning

Complexity of checking *S entails E*

Entailment	Restrictions on *S*	Restrictions on *E*	Complexity
simple,RDF,RDFS	none	none	NP-complete *
simple, RDF	none	ground	LogSpace †
RDFS	none	ground	P *
RDFS	none	ground	P-hard †,‡
eRDFS	none	none	NP-hard *
eRDFS	no nonst. RDFS	no nonst. RDFS	NP-complete †
eRDFS	no nonst. RDFS	ground, no nonst. RDFS	P †
D*	none	none	NP-complete *
D	none	none	coNP-hard †,§

* Established in [Gutierrez et al., 2004; de Bruijn et al., 2005; ter Horst, 2005]

† Novel

‡ By reduction from PATH SYSTEM ACCESSIBILITY

§ By reduction from NONEXISTENCE OF K-COLORING

Outlook

- Upper bound extensional RDFS entailment
- Precise relationships between extensional RDFS and OWL DL
- Combination of RDFS and logical rules
  - Ongoing work in the RIF working group
  - Second draft:
    RIF RDF and OWL Compatibility, Jos de Bruijn, Editor, W3C Working Draft, 15 April 2008. Latest version available at `http://www.w3.org/TR/rif-rdf-owl/`.