Taming Existence in RDF Querying

François Bry[1], Tim Furche[1], Clemens Ley[2], Benedikt Linse[1], and Bruno Marnette[2]

[1] Institute for Informatics, University of Munich, Oettingenstraße 67, D-80538 München, Germany
[2] Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, OX1 3QD, England

Abstract. We introduce the recursive, rule-based RDF query language RDFLog. RDFLog extends previous RDF query languages by arbitrary quantifier alternation: blank nodes may occur in the scope of all, some, or none of the universal variables of a rule. In addition RDFLog is aware of important RDF features such as the distinction between blank nodes, literals and URIs or the RDFS vocabulary. The semantics of RDFLog is closed (every answer is an RDF graph), but lifts RDF's restrictions on literal and blank node occurrences for intermediary data. We show how to define a sound and complete operational semantics that can be implemented using existing logic programming techniques. Using RDFLog we classify previous approaches to RDF querying along their support for blank node construction and show equivalence between languages with full quantifier alternation and languages with only $\forall\exists$ rules.

1 Introduction

With the staggering amount of data available in RDF form on the Web, the second indispensable ingredient becomes the easy selection and processing of RDF data. For that purpose, a large number of RDF query languages has been proposed. In this paper, we add a further exemplar: RDFLog extends datalog to support the distinguishing features of RDF such as blank nodes and the logical core [1] of the RDFS vocabulary. In RDFLog, Blank nodes can be constructed by existentially quantified variables in rule heads. RDFLog allows *full alternation* between existential and universal quantifiers in a rule. This sharply contrasts with previous approaches to rule-based query languages that either do not support blank nodes (in rule heads) at all [2], or only a limited form of quantifier alternation [3].

To illustrate the benefits of full quantifier alternation, imagine an information system about university courses. We distinguish three types of rules with existential quantifiers (and thus blank nodes) based on the alternation of universal and existential quantifiers:

(1) "Someone knows each professor" can be represented in RDFLog as

$$\exists stu \forall prof\ ((prof, \mathsf{rdf{:}type}, \mathsf{uni{:}professor}) \rightarrow (stu, \mathsf{foaf{:}knows}, prof)) \qquad (1)$$

D. Calvanese and G. Lausen (Eds.): RR 2008, LNCS 5341, pp. 236–237, 2008.

We call such rules $\exists\forall$ rules. Some approaches such as [**?**] are limited to rules of this form. We show that a recursive rule language that is limited to these kind of rules is strictly less expressive than a language that allows rules also of the form discussed under (2) and (3). The gain is that languages with only $\exists\forall$ rules are still decidable. However, we also show that there are larger fragments of RDFLog that are still decidable.

(2) Imagine, that we would like to state that each lecture must be "practiced" by another course (such as a tutorial or practice lab) without knowing more about that course. This statement can not be expressed by $\exists\forall$ rules. In RDFLog it can be represented as

$$\forall lec \exists crs \big((lec, \mathsf{rdf{:}type}, \mathsf{uni{:}lecture}) \rightarrow (crs, \mathsf{uni{:}practices}, lec) \big) \tag{2}$$

Such rules are referred to as $\forall\exists$ rules. Recent proposals for rule extensions to SPARQL are limited to this form, if they consider blank nodes in rule heads at all. The reason is that in SPARQL `CONSTRUCT` patterns a fresh blank node is constructed for every binding of the universal variables.

(3) To the best of our knowledge, RDFLog is the first RDF query language that supports the third kind of rules, where quantifiers are allowed to alternate freely: This allows to express statements such as, for each lecture there is a course that "practices" that lecture and is attended by all students attending the lecture. This is represented in RDFLog as

$$\forall lec \exists crs \forall stu \big((lec, \mathsf{rdf{:}type}, \mathsf{uni{:}lecture}) \wedge (stu, \mathsf{uni{:}attends}, lec) \rightarrow (crs, \mathsf{uni{:}practices}, lec) \wedge (stu, \mathsf{uni{:}attends}, crs) \big) \tag{3}$$

We show (for the first time) that rules with full quantifier alternation can be normalized to $\forall\exists$ form. Thus full quantifier alternation does not add to the expressiveness of RDFLog. Rather, for all languages with $\forall\exists$ rules it comes by virtue of the rewriting presented here for free, despite being considerably more convenient for the programmer.

References

1. Muñoz, S., Pérez, J., Gutierrez, C.: Minimal deductive systems for RDF. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519. Springer, Heidelberg (2007)
2. Polleres, A.: From SPARQL to rules (and back). In: Proc. Int'l. World Wide Web Conf. (WWW). ACM, New York (2007)
3. Schenk, S., Staab, S.: Networked graphs: A declarative mechanism for SPARQL rules, SPARQL views and RDF data integration on the Web. In: Proc. Int'l. World Wide Web Conf (WWW). ACM Press, New York (2008)
4. Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L.: Data exchange: Semantics and query answering. In: Calvanese, D., Lenzerini, M., Motwani, R. (eds.) ICDT 2003. LNCS, vol. 2572. Springer, Heidelberg (2002)
5. Yang, G., Kifer, M.: Reasoning about anonymous resources and meta statements on the Semantic Web. Journal of Data Semantics 1 (2003)