Correspondence Papers

Some Remarks on the Paper "semQA: SPARQL with Idempotent Disjunction"

Marcelo Arenas, Claudio Gutierrez, and Jorge Pérez

Abstract—In the paper, "semQA: SPARQL with Idempotent Disjunction" [4], the authors study the RDF query language SPARQL. In particular, they claim that some of the results presented in [1] are not correct. In this note, we refute the claims made in [4], and actually show that some of the formal results of [4] are incorrect.

Index Terms—RDF, query language, SPARQL, evaluation problem, combined complexity.

1 Proposition 4 of [4]

In [4], the authors correctly point out that Proposition 1 in [1] is incorrect, as the operator OPT does not distribute over the operator UNION in SPARQL, that is, the following equivalence does not hold in general:

$$(P_1 \text{ OPT } (P_2 \text{ UNION } P_3)) \equiv ((P_1 \text{ OPT } P_2) \text{ UNION } (P_1 \text{ OPT } P_3)). \quad (1)$$

In fact, this error was kindly brought to our attention by Michael Schmidt [3], and it is mentioned in the journal version of [1] that was recently published [2]. In [4], the authors present this error in Proposition 4. However, this proposition as stated is not correct. In [4], the authors start by saying "Let P_1, P_2, and P_3 be graph patterns." In standard mathematical notation, this statement means that the proposition should hold for every choice of P_1, P_2, and P_3. Thus, as claimed in [4], (1) should not hold for any patterns P_1, P_2, and P_3. But this is not the case as, for example, (1) holds for $P_1 = P_2 = P_3 = (?X, ?Y, ?Z)$.

2 Corollary 2 of [4]

A graph pattern P is in UNION normal form if $P = (P_1 \text{ UNION } P_2 \text{ UNION} \cdots \text{UNION } P_n)$, where each pattern $P_i (1 \leq i \leq n)$ does not mention the UNION operator.

The main motivation for including equivalence (1) in [1] was to prove that every SPARQL pattern is equivalent to a pattern in UNION normal form. It is important to notice that from the fact that (1) does not hold, one can conclude that the proof in [1] is incorrect, but one cannot conclude what is stated in Corollary 2 in [4]:

Corollary 2 ([4]). *It is not always possible to transform a graph pattern into an equivalent one of the form* $(P_1 \text{ UNION } P_2 \text{ UNION} \cdots \text{UNION } P_n)$, *where each* P_i *is a graph pattern that does not contain UNION operators.*

To prove this corollary, one has to exhibit a SPARQL pattern that is not equivalent to any pattern in UNION normal form, but this is not done in [4] (no proof of this corollary is provided in [4]). Moreover, for the formalization of SPARQL introduced in [1], it is possible to prove that every SPARQL pattern is equivalent to a pattern in UNION normal form. The new proof of this fact does not use equivalence (1) and can be found in [2].

3 Corollary 4 of [4]

It is claimed in Corollary 4 of [4] that the combined complexity of the evaluation problem for SPARQL is NP-complete. Furthermore, it is claimed in [4] that this result does not contradict the PSPACE-completeness result for SPARQL proved in [1], since the authors claim that [1] studied the expression complexity of the evaluation problem for SPARQL. Below, we show that the proof of Corollary 4 as well as the claim about contradicting the complexity results of [1] are both incorrect.

In [4], the authors study the following decision problem, which is called SPARQL-C problem. Given a SPARQL pattern P and an RDF graph G, check whether there exists a solution in the evaluation of P over G. In Corollary 4, the authors claim that the SPARQL-C problem is NP-complete. Next, we show that from the arguments given in [4], one cannot conclude that the SPARQL-C problem is NP-complete.

In Definition 1 in [4], the authors introduce the OR operator for SPARQL. Later, in Proposition 7, they show the following equivalence:

$$(P_1 \text{ OPT } P_2) \equiv (P_1 \text{ OR } (P_1 \text{ AND } P_2)). \quad (2)$$

This equation is used in [4] to define the *i-d pattern* corresponding to a SPARQL pattern. More precisely, a pattern P' is the i-d pattern corresponding to a SPARQL pattern P, if P' is obtained from P by replacing every UNION operator by an OR operator, and every OPT operator by the right-hand side of (2).

To prove Corollary 4, the authors first show in Theorem 2 that the SPARQL-C problem for patterns that use only AND, FILTER, and OR is NP-complete, and then they use the following observation (mentioned in the proof of Corollary 3):

Claim A ([4]). *Let G be an RDF graph, P a pattern, and P' the i-d pattern corresponding to P. If there exists a solution for P' over G, then there exists a solution for P over G.*

In fact, the authors mention that the NP-completeness of the SPARQL-C problem follows directly from Theorem 2 and Claim A. The rationale behind this conclusion is the following: One knows that the SPARQL-C problem for patterns that use only AND, FILTER, and OR is NP-complete (Theorem 2). Thus, the SPARQL-C problem is NP-complete as one can translate an arbitrary pattern P into the i-d pattern corresponding to it, and then use Claim A. But this conclusion is incorrect, since **the i-d pattern corresponding to a pattern P can be of exponential size in the size of P**. For example, consider a pattern P:

$$(((\cdots((P_1 \text{ OPT } P_2) \text{ OPT } P_3) \cdots) \text{ OPT } P_{n-1}) \text{ OPT } P_n).$$

Then the i-d pattern corresponding to P, that is obtained by successively applying equivalence (2), has 2^{n-i} copies of P_i for every $i \in \{1, \ldots, n\}$, and thus, it is of exponential size in the size of P. This shows that the argument in [4] only proves that the SPARQL-C problem is in NEXPTIME.

- *M. Arenas and J. Pérez are with the Department of Computer Science, Pontificia Universidad Católica de Chile, Vicuna Mackenna 4860, Edificio San Agustin - 4to piso, Santiago 7820436, Chile. E-mail: {marenas, jperez}@ing.puc.cl.*
- *C. Gutierrez is with the Department of Computer Science, Universidad de Chile, Blanco Encalada 2120, Santiago 8370459, Chile. E-mail: cgutierr@dcc.uchile.cl.*

Manuscript received 8 Mar. 2009; revised 30 Dec. 2009; accepted 18 June 2010; published online 16 Aug. 2010.
Recommended for acceptance by M. Garofalakis.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2009-03-0123.
Digital Object Identifier no. 10.1109/TKDE.2010.139.

We now consider the comment in [4] about contradicting the complexity results of [1]. In [1], a variation of the SPARQL-C problem is considered: given a SPARQL pattern P, an RDF graph G, and a mapping μ, check whether μ is a solution in the evaluation of P over G. Let us call this problem the SPARQL-D problem. In [1], it is shown that the SPARQL-D problem is PSPACE-complete.

In [1], the lower bound for the SPARQL-D problem is proved by considering a fixed database, and the upper bound is proved without imposing any restrictions. Thus, contrary to what is claimed in [4], it is actually shown in [1] that both, the expression and the combined complexity of the SPARQL-D problem, are PSPACE-complete. But not only that, the claim in [4] that the combined complexity is lower than the expression complexity for the case of SPARQL is incorrect, as the expression complexity of a query language is lower than or equal to the combined complexity of the language [5]. It is straightforward to prove this general fact, as in the case of the expression complexity one assumes the database to be fixed, while in the combined complexity the database is part of the input. In particular, any reduction showing a lower bound for the expression complexity of a query language can be used to prove the same lower bound for the combined complexity of this language.

4 Corollary 5 of [4]

In Corollary 5 of [4], the authors claim that neither OPT nor UNION adds complexity to the SPARQL language. From the above discussion about Corollary 4 of [4], it follows that Corollary 5 cannot be obtained from any of the results in the paper. Moreover, as we show in [1], [2], and other authors have also shown [3], the use of the OPT and UNION operators indeed adds complexity to the evaluation problem for SPARQL.

5 An Additional Remark

We conclude this note by pointing out a misleading comment in [4]. In [1], it is shown that the combined complexity of the SPARQL-D problem for the fragment composed by the AND and FILTER operators is polynomial, while it is NP-complete if one also includes UNION and PSPACE-complete, if all four operators are included. However, it is claimed in Section 4 of [4] that:

> *"The analysis in [1] states that these three subsets have an increasing complexity. However, a close verification of the proofs offered by the authors in the long version of their paper led us to reexamine these conclusions; in particular, we show here that the combined complexity of conjunctive queries is NP-complete and, thus, intractable."*

From this comment, one infers that the polynomial result in [1] is incorrect. But this is not proved in [4]. Indeed, the authors of [4] study the SPARQL-C problem, which is a variation of the evaluation problem considered in [1]. Unfortunately, this is not clearly stated in [4].

Acknowledgments

The authors are grateful to the anonymous referees for their helpful comments. Arenas and Gutierrez were supported by FONDECYT grant 1090565.

References

[1] J. Pérez, M. Arenas, and C. Gutierrez, "Semantics and Complexity of SPARQL," *Proc. Fifth Int'l Semantic Web Conf.*, pp. 30-43, 2006.
[2] J. Pérez, M. Arenas, and C. Gutierrez, "Semantics and Complexity of SPARQL," *ACM Trans. Database Systems*, vol. 34, no. 3, pp. 16:1-16:45, 2009.
[3] M. Schmidt, M. Meier, and G. Lausen, "Foundations of SPARQL Query Optimization," *Proc. 13th Int'l Conf. Database Theory*, pp. 4-33, 2010.
[4] E.P. Shironoshita, Y.R. Jean-Mary, R.M. Bradley, and M.R. Kabuka, "semQA: SPARQL with Idempotent Disjunction," *IEEE Trans. Knowledge and Data Eng.* vol. 21, no. 3, pp. 401-414, Mar. 2009.
[5] M.Y. Vardi, "The Complexity of Relational Query Languages (Extended Abstract)," *Proc. 14th Ann. ACM Symp. Theory of Computing*, pp. 137-146, 1982.