Some remarks on the paper "semQA: SPARQL with Idempotent Disjunction"

Marcelo Arenas
PUC Chile

Claudio Gutierrez
Universidad de Chile

Jorge Pérez
PUC Chile

Abstract—In the paper "semQA: SPARQL with Idempotent Disjunction" [Shironoshita et al. 2009], the authors study the RDF query language SPARQL. In particular, they claim that some of the results presented in [Pérez et al. 2006] are not correct. In this note, we refute the claims made in [Shironoshita et al. 2009], and actually show that some of the formal results of [Shironoshita et al. 2009] are incorrect.

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

Proposition 4 of [Shironoshita et al. 2009]

In [Shironoshita et al. 2009], the authors correctly point out that Proposition 1 in [Pérez et al. 2006] 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 [Schmidt et al. 2008], and it is mentioned in the journal version of [Pérez et al. 2006] that was published recently [Pérez et al. 2009]. In [Shironoshita et al. 2009], the authors present this error in Proposition 4. However, this proposition as stated is not correct. In [Shironoshita et al. 2009], 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 [Shironoshita et al. 2009], equation (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 $P_1 = P_2 = P_3 = (?X, ?Y, ?Z)$.

Corollary 2 of [Shironoshita et al. 2009]

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 [Pérez et al. 2006] 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 [Pérez et al. 2006] is incorrect, but one can not conclude what is stated in Corollary 2 in [Shironoshita et al. 2009]:

Corollary 2. [Shironoshita et al. 2009] *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 [Shironoshita et al. 2009] (no proof of this corollary is provided in [Shironoshita et al. 2009]). Moreover, for the formalization of SPARQL introduced in [Pérez et al. 2006], 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 [Pérez et al. 2009].

Corollary 4 of [Shironoshita et al. 2009]

It is claimed in Corollary 4 of [Shironoshita et al. 2009] that the combined complexity of the evaluation problem for SPARQL is NP-complete. Furthermore, it is claimed in [Shironoshita et al. 2009] that this result does not contradict the PSPACE-completeness result for SPARQL proved in [Pérez et al. 2006], since the authors claim that [Pérez et al. 2006] 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 [Pérez et al. 2006] are both incorrect.

In [Shironoshita et al. 2009], 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 [Shironoshita et al. 2009], one can not conclude that the SPARQL-C problem is NP-complete.

In Definition 1 in [Shironoshita et al. 2009], 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 [Shironoshita et al. 2009] 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. [Shironoshita et al. 2009] *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:

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

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 [Shironoshita et al. 2009] only proves that the SPARQL-C problem is in NEXPTIME.

We now consider the comment in [Shironoshita et al. 2009] about contradicting the complexity results of [Pérez et al. 2006]. In [Pérez et al. 2006], 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 [Pérez et al. 2006], it is shown that the SPARQL-D problem is PSPACE-complete.

In [Pérez et al. 2006], 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 [Shironoshita et al. 2009], it is actually shown in [Pérez et al. 2006] that both the expression and the combined complexity of the SPARQL-D problem are PSPACE-complete. But not only that, the claim in [Shironoshita et al. 2009] 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 [Vardi 1982]. 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.

Corollary 5 of [Shironoshita et al. 2009]

In Corollary 5 of [Shironoshita et al. 2009], the authors claim that neither OPT nor UNION adds complexity to the SPARQL language. From the above discussion about Corollary 4 of [Shironoshita et al. 2009], it follows that Corollary 5 cannot be obtained from any of the results in the paper. Moreover, as we show in [Pérez et al. 2006], [Pérez et al. 2009], and other authors have also shown [Schmidt et al. 2008], the use of the OPT and UNION operators indeed add complexity to the evaluation problem for SPARQL.

An additional remark

We conclude this note by pointing out a misleading comment in [Shironoshita et al. 2009]. In [Pérez et al. 2006], 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 [Shironoshita et al. 2009] (Section 4) that:

> *"The analysis in [Pérez et al. 2006] 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 [Pérez et al. 2006] is incorrect. But this is not proved in [Shironoshita et al. 2009]. Indeed, the authors of [Shironoshita et al. 2009] study the SPARQL-C problem which is a variation of the evaluation problem considered in [Pérez et al. 2006]. Unfortunately, this is not clearly stated in [Shironoshita et al. 2009].

References

[Pérez et al. 2006] Jorge Pérez, Marcelo Arenas, Claudio Gutierrez. Semantics and Complexity of SPARQL. In Proceedings of the *5th International Semantic Web Conference*, pages 30-43, 2006.

[Pérez et al. 2009] Jorge Pérez, Marcelo Arenas, Claudio Gutierrez. Semantics and Complexity of SPARQL. *ACM Transactions on Database Systems*, 33(4), Article 16, 2009.

[Schmidt et al. 2008] M. Schmidt, M. Meier, G. Lausen. Foundations of SPARQL Query Optimization. *arXiv.org paper arXiv:0812.3788v1*, December 19, 2008.

[Shironoshita et al. 2009] E. P. Shironoshita, Y. R. Jean-Mary, R. M. Bradley, M. R. Kabuka. semQA: SPARQL with Idempotent Disjunction. In *IEEE Transactions on Knowledge and Data Engineering* 21(3):401-414, March 2009.

[Vardi 1982] M. Y. Vardi. The Complexity of Relational Query Languages (Extended Abstract) In Proceedings of the *14th Annual ACM Symposium on Theory of Computing*, pages 137-146, 1982.