Towards Reconciling SPARQL and Certain Answers (Extended Abstract)*

Shqiponja Ahmetaj, Wolfgang Fischl, Reinhard Pichler, Mantas Šimkus, and Sebastian Skritek

Institute of Information Systems, TU Vienna, Austria
{ahmetaj,wfischl,pichler,simkus,skritek}@dbai.tuwien.ac.at

Abstract. SPARQL entailment regimes are strongly influenced by the big body of works on ontology-based query answering, notably in the area of Description Logics (DLs). However, the semantics of query answering under SPARQL entailment regimes is defined in a more naive and much less expressive way than the certain answer semantics usually adopted in database and DL literature. In this paper we introduce an intuitive certain answer semantics also for SPARQL and show the feasibility of this approach. For OWL 2 QL entailment, we develop algorithms for the evaluation of an interesting fragment of SPARQL (the so-called well-designed SPARQL). Exploiting these algorithms, we can show that the complexity of neither query answering nor the most fundamental query analysis tasks (such as query containment and equivalence testing) is negatively affected by the presence of OWL 2 QL entailment under the proposed semantics.

1 Introduction

In the recently released recommendation [7], the W3C has defined various SPARQL entailment regimes to allow users to specify implicit knowledge about the vocabulary in an RDF graph. The theoretical underpinning to the systems for query answering under rich entailment regimes is provided by the big body of work on ontology-based query answering, notably in the area of Description Logics (DLs) [4]. However, the semantics of query answering under SPARQL entailment regimes is defined in a more naive and much less expressive way than the *certain answer semantics* usually adopted in the DL and database literature.

Example 1. Consider an RDF graph G containing a single triple $(b, \mathsf{a}, \mathsf{Prof})$ – stating that b is a professor – and an ontology $\mathcal{O}$ containing the triples

$$(\mathsf{Prof}, \texttt{rdfs:sc}, \texttt{\_:b}), (\texttt{\_:b}, \mathsf{a}, \texttt{owl:Restriction}),$$
$$(\texttt{\_:b}, \texttt{owl:onProperty}, \mathsf{teaches}), (\texttt{\_:b}, \texttt{owl:someValuesFrom}, \texttt{owl:Thing}).$$

[*] A longer version of this paper was presented at WWW 2015 [1].

– stating that every professor teaches somebody. Now consider the following simple SPARQL query: SELECT $?x$ WHERE $(?x, \mathsf{teaches}, ?y)$.[1] Following the SPARQL entailment regimes standard [7], this query yields an empty result. □

This result is rather unintuitive: by the inclusion we know for certain that b teaches somebody. However, the SPARQL entailment standard requires that all values assigned to any variable must come from the RDF graph – thus treating distinguished variables (which are ultimately output) and non-distinguished variables (which are eventually projected out) in the same way. In contrast, the certain answer semantics retrieves all mappings on the distinguished variables that allow to satisfy the query in every possible model of the database and the ontology (yielding the certain answer $\mu = \{?x \to b\}$ in the above example).

The **goal of this work** is to introduce an intuitive certain answer semantics also for SPARQL under OWL 2 QL entailment with similarly favorable results as for CQ answering under DL-Lite$_{\mathcal{R}}$ (which provides the theoretical underpinning of the OWL 2 QL entailment regime).

The reason why for this purpose we cannot simply take over all the results from CQ answering under DL-Lite is that SPARQL provides some crucial extensions over CQs. One of them is the OPTIONAL operator (henceforth referred to as OPT operator, for short). It allows the user to retrieve *partial solutions* in cases where no match for the complete query can be found, instead of failing to provide any solution. Observe that these queries are no longer monotone. Thus, the usual certain answer semantics (i.e., something is a certain answer if it is present in every model) turns out to be unsatisfactory:

Example 2. Consider the SPARQL query: SELECT $?x, ?z$ WHERE $(?x, \mathsf{teaches}, ?y)$ OPT $(?y, \mathsf{knows}, ?z)$ over the graph $G = \{(b, \mathsf{teaches}, c)\}$ and empty ontology $\mathcal{O}$. The query yields a unique solution $\mu = \{?x \to b\}$. Clearly, also the extended graph $G' = G \cup \{(c, \mathsf{knows}, d)\}$ is a model of $(G, \mathcal{O})$. But in G', μ is no longer a solution since μ can be extended to solution $\mu' = \{?x \to b, ?z \to d\}$. Hence, there exists no mapping which is a solution in every possible model of $(G, \mathcal{O})$. □

In this paper, we discuss further problems with a literal adoption of a certain answer semantics in the presence of the OPT operator, and propose a suitable modified definition for the class of *well-designed* SPARQL queries [11]. This modified semantics also requires an adaptation and extension of the known query answering algorithms for DL-Lite. We present two such modified algorithms for query evaluation. Finally, we shall show that the additional expressive power due to the certain answers comes without an increase of the complexity.

Related Work. For our findings the following work is most relevant to us: the semantics of SPARQL was investigated in [3], which also introduces weakly-monotone queries, i.e. well-designed SPARQL. The semantics for SPARQL over

[1] Following [11], we use a more algebraic style notation, denoting triples in parentheses with comma-separated components, rather than the blank-separated turtle notation.

OWL ontologies is standardized by the World Wide Web consortium in [7]. Our two algorithms are based upon the standard rewriting algorithm for *DL-Lite* [5] and a more advanced algorithm for the DL Horn-$\mathcal{SHIQ}$ [6]. There is a huge body of results on CQ answering under different DLs (cf. [5, 6, 10, 12]). For SPARQL recent work [8] presents a *stronger* semantics, where entire mappings are discarded, whose possible extensions to optional subqueries would imply inconsistencies in the knowledge base. In [2], the authors describe a rewriting of SPARQL query answering under OWL 2 QL into Datalog$^{\pm}$. A slight modification allows them to remove the active domain semantics of variables, however this only applies to variables occuring in a single BGP. Libkin [9] also criticizes the standard notion of certain answers in case of non-monotone queries. Similar to his suggestion to use the greatest lower bounds in terms of informativeness, our approach chooses the most informative solutions as certain answers.

2 SPARQL and OWL 2 QL

OWL 2 QL is based on DL-Lite$_{\mathcal{R}}$, a lightweight description logic. Its fundamental building blocks are *constants* c, *atomic concepts* A and *atomic roles* R, which are countably infinite and mutually disjoint subsets of a set $\mathbf{U}$ of URIs. From these we can build *basic roles* R and R^{-}, and *basic concepts* B and $\exists Q$, where Q is a basic role. Using the above, DL-Lite$_{\mathcal{R}}$ allows one to express the following kind of statements: Membership assertions (c, a, B) or (c, Q, c'), concept inclusions $(B_1, \texttt{rdfs:sc}, B_2)$, role inclusions $(Q_1, \texttt{rdfs:sp}, Q_2)$ as well as concept and role disjointness (where c, c' are constants and B_i, Q_i are basic concepts resp. basic roles). In the following, an ontology $\mathcal{O}$ is any set of such expressions, excluding membership assertions, which we assume to be part of the RDF graph. A *knowledge base (KB)* $\mathcal{G} = (G, \mathcal{O})$ consists of an RDF graph G and an ontology $\mathcal{O}$.

The basic building block of SPARQL queries are *triple patterns* $(s, p, o) \in (\mathbf{U} \cup \mathbf{V})^3$, where $\mathbf{V}$ is a set of variables. In this work we only consider triple patterns of the form $(?x, \mathsf{a}, B)$ or $(?x, Q, ?y)$ where B (Q) is a basic concept (role). More complex *graph patterns* are built from triple patterns via operators like e.g. AND, OPT, or UNION. Here, we consider a SPARQL query to be a graph pattern, possibly extended by top-level projection. Given a graph pattern P, a set $\mathcal{X} \subseteq \mathbf{V}$ of variables occurring in P and an RDF graph G, the answer $[\![(P, \mathcal{X})]\!]_G$ to P, projected to $\mathcal{X}$, over G is a set of partial mappings from $\mathcal{X}$ to $\mathbf{U}$. We say a mapping μ_1 is subsumed by another mapping μ_2, denoted by $\mu_1 \sqsubseteq \mu_2$, if $\mathsf{dom}(\mu_1) \subseteq \mathsf{dom}(\mu_2)$ and $\mu_1(?x) = \mu_2(?x)$ for all $?x \in \mathsf{dom}(\mu_1)$, where $\mathsf{dom}(\mu_i)$ denotes the set of variables the mapping μ_i is defined on.

By imposing certain restrictions on the occurrence of variables, the fragment of *well-designed SPARQL (wdSPARQL)* was introduced in [11]. It possesses several desirable properties, like coNP-completeness of query evaluation. Of importance for our work is that these queries are *weakly-monotone* [3]: If $\mu \in [\![(P, \mathcal{X})]\!]_G$, then for every RDF graph G' with $G \subseteq G'$, there exists $\mu' \in [\![(P, \mathcal{X})]\!]_{G'}$ s.t. $\mu \sqsubseteq \mu'$ (i.e., while μ need not be a solution over G', it can be extended to one).

3 Certain Answers of well-designed SPARQL

Before providing our definition of certain answers, we need to introduce two additional notions. Let P be a well-designed graph pattern. Following [11], we say that P' is a reduction of P (denoted as $P' \trianglelefteq P$) if P' can be constructed from P by replacing in P sub-patterns of the form (P_1 OPT P_2) by P_1. Second, for a mapping μ and some property A, we shall say that μ is $\sqsubseteq$-*maximal w.r.t.* A if μ satisfies A, and there is no μ' such that $\mu \sqsubseteq \mu'$, $\mu' \not\sqsubseteq \mu$, and μ' satisfies A.

Definition 1. *Let $\mathcal{G} = (G, \mathcal{O})$ be a KB and $Q = (P, \mathcal{X})$ a well-designed query. A mapping μ is a* certain answer *to Q over $\mathcal{G}$ if it is a $\sqsubseteq$-maximal mapping s.t. (1) $\mu \sqsubseteq [\![Q]\!]_{G'}$ for every model G' of $\mathcal{G}$, and (2) $\mathsf{vars}(P') \cap \mathcal{X} = \mathsf{dom}(\mu)$ for some $P' \trianglelefteq P$. We denote by $\mathsf{cert}(P, \mathcal{X}, \mathcal{G})$ the set of all certain answers to Q over $\mathcal{G}$.*

The reason for restricting the set of certain answers to $\sqsubseteq$-maximal mappings is that queries with projection and/or UNION may have "subsumed" solutions, i.e. solutions s.t. also some proper extension is a solution. But then – with set semantics – we cannot recognize the reason why some subsumed solution is possibly not a solution in some possible world, as illustrated in Example 3. Since in our first step towards reconciling SPARQL and certain answers we decide to stick to set semantics, we allow only "maximal" solutions as certain answers.

Example 3. Consider the following query SELECT $?x, ?z$ WHERE $(?x, \mathsf{teaches}, ?y)$ OPT $(?y, \mathsf{knows}, ?z)$ over the graph $G = \{(a, \mathsf{teaches}, b), (b, \mathsf{knows}, c), (a, \mathsf{teaches}, d)\}$ and empty ontology $\mathcal{O}$. As possible models of $(G, \mathcal{O})$ we have all graphs containing G. Hence, $\mu = \{?x \to a, ?z \to c\}$ and $\mu' = \{?x \to a\}$ ($?y$ is bound to d) are both answers to G and can be extended to solutions in every possible model.

Next consider $G' = \{(a, \mathsf{teaches}, b), (b, \mathsf{knows}, c)\}$. If we take as certain answers all mappings that can be extended to some solution in every possible model, then μ' from above is still a certain answer. □

Property (2) in the definition of certain answers ensures that the domain of such an answer adheres to the structure of nested OPTs in the query. However, we can show that this property need not be considered during the computation of the certain answers, but can be enforced in a simple post-processing step. We call such answers that satisfy Definition 1 except property (2) *certain pre-answers*, and use $\mathsf{certp}(P, \mathcal{X}, \mathcal{G})$ to denote the set of all certain pre-answers. The same is also true for projection, which can also be performed in a simple post-processing step. Thus, it suffices to compute $\mathsf{certp}(P, \mathcal{G})$, which can be done via universal solutions (referred to as *canonical model* in the area of DLs) as follows.

Theorem 1. *Let $\mathcal{G} = (G, \mathcal{O})$ be a KB and P a well-designed graph pattern. Then, $\mathsf{certp}(P, \mathcal{G}) = \mathrm{MAX}([\![P]\!]_{\mathsf{univ}(G)}\downarrow)$, where $\mathrm{MAX}(M)$ is the set of $\sqsubseteq$-maximal mappings in M, $M\downarrow := \{\mu\downarrow \mid \mu \in M\}$ ($\mu\downarrow$ is the restriction of μ to those variables mapped to the active domain of G), and $\mathsf{univ}(G)$ is a universal solution of $\mathcal{G}$.*

However, computing the certain answers via a universal solution is not always practical, e.g. the universal solution can be infinite. As a result, query rewriting algorithms have been developed: These algorithms take the input query and the ontology, and rewrite them into a single query that can be evaluated over the input database without considering the ontology. By introducing several adaptations and extensions of the rewriting-based CQ evaluation for DL-Lite from [5], we develop two different approaches to answer well-designed SPARQL queries under OWL 2 QL entailment.

The first one proceeds in a modular way by rewriting basic building blocks of a SPARQL query (so-called BGPs) individually. It thus follows the general philosophy of SPARQL entailment regimes. One possible disadvantage of this modular approach is that it requires to maintain additional data structures to ensure consistency when combining the partial solutions for different BGPs. As a consequence, the complete algorithm has to be implemented from scratch because the standard tools cannot handle these additional data structures.

The goal of the second approach is thus to make use of the standard technology as much as possible. The idea is to transform the OWL 2 QL entailment under our new semantics into SPARQL query evaluation under RDFS entailment, for which strong tools are available. Unlike the first – modular – approach, this rewriting proceeds in a holistic way, i.e. it always operates on the whole query.

Based on these rewriting algorithms, we analyze the complexity of query answering and of several static query analyzing tasks such as query containment and equivalence. We are able to show that the additional power of our new semantics comes without additional costs in terms of complexity.

Acknowledgements

This work was supported by the Vienna Science and Technology Fund (WWTF), project ICT12-15 and by the Austrian Science Fund (FWF): P25207-N23.

References

1. S. Ahmetaj, W. Fischl, R. Pichler, M. Šimkus, and S. Skritek. Towards reconciling SPARQL and certain answers. In *Proc. of WWW 2015*, 2014.
2. M. Arenas, G. Gottlob, and A. Pieris. Expressive languages for querying the semantic web. In *Proc. of PODS 2014*, pages 14–26. ACM, 2014.
3. M. Arenas and J. Pérez. Querying semantic web data with SPARQL. In *Proc. of PODS 2011*, pages 305–316. ACM, 2011.
4. F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
5. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *J. Autom. Reasoning*, 39(3):385–429, 2007.
6. T. Eiter, M. Ortiz, M. Šimkus, T. Tran, and G. Xiao. Query rewriting for Horn-$\mathcal{SHIQ}$ plus rules. In *Proc. of AAAI 2012*. AAAI Press, 2012.
7. B. Glimm and C. Ogbuji. SPARQL 1.1 Entailment Regimes. W3C Recommendation, W3C, Mar. 2013. `http://www.w3.org/TR/sparql11-entailment`.

8. E. V. Kostylev and B. Cuenca Grau. On the semantics of SPARQL queries with optional matching under entailment regimes. In *Proc. of ISWC 2014*, 2014.
9. L. Libkin. Incomplete data: what went wrong, and how to fix it. In *Proc. of PODS 2014*, pages 1–13. ACM, 2014.
10. M. Ortiz, D. Calvanese, and T. Eiter. Data complexity of query answering in expressive description logics via tableaux. *Journal of Automated Reasoning*, 41(1):61–98, 2008.
11. J. Pérez, M. Arenas, and C. Gutierrez. Semantics and complexity of SPARQL. *ACM Trans. Database Syst.*, 34(3), 2009.
12. R. Rosati. On conjunctive query answering in EL. In *Proc. of DL 2007*, 2007.