On the Static Analysis for SPARQL Queries Using Modal Logic

Nicola Guido
Université Joseph Fourier, Grenoble
nicola.guido@inria.fr

Abstract

Static analysis is a core task in query optimization and knowledge base verification. We study static analysis techniques for SPARQL, the standard language for querying Semantic Web data. Specifically, we investigate the query containment problem and query-update independence analysis. We are interested in developing techniques through reductions to the validity problem in logic.

1 Introduction

The Resource Description Framework (RDF) is a graph data format for the representation of information in the Web. An RDF statement is a *subject-predicate-object* structure, called RDF triple, intended to describe resources and properties of those resources. SPARQL is a W3C Recommendation query language for RDF. It is based on the notion of triple patterns. During query evaluation, variables inside these patterns are matched against the RDF input graph. The solution of the evaluation process is a set of mappings, where each mapping associates a set of variables with graph components.

We study static analysis techniques for SPARQL, in particular for the query containment problem and the query-update independence analysis problem.

The query containment problem consists in determining whether the results of a query are included in the results of another query, for every dataset. Query containment is crucial in several contexts, such as query optimization, knowledge base verification, information integration, integrity checking and cooperative answering. Needless to say, query containment is undecidable if we do not limit the expressive power of the query language.

The query-update independence analysis problem has been introduced with the release of the update language for SPARQL in SPARQL 1.1 Update specification. The query-update independence problem consists in determining whether the query results are affected by an update execution, for any possible RDF input graph. Determining independence is especially useful in the context of huge RDF repositories, where it permits to avoid expensive yet useless re-evaluation of queries.

For these two problems, we are interested in studying static analysis techniques through a reduction to the validity problem in logic. We consider the modal logic K, that allows less complex decision procedure for satisfiability judgement than a first-order logic, yet it still has strong expressive power to support the query language features that we want to analyse. We first translate queries into formulas in K and then the overall problems are reduced into satisfiability tests.

One advantage of this approach is to open the way for implementations using off-the-shelf satisfiability solvers for K. This makes it possible to benefit from years of research in optimization of modal logic satisfiability solvers in the context of SPARQL static analysis.

2 Related Work

Static analysis and especially the containment problem for SPARQL queries is a topic that has attracted attention quite recently [Chekol *et al.*, 2012; 2013; Letelier *et al.*, 2012; Pichler and Skritek, 2014]. These studies concentrate on sound mathematical analyses, computational complexity results and some of them consider implementation issues.

The line of research found in [Letelier *et al.*, 2012; Pichler and Skritek, 2014] concentrates on complexity results, and provides complexity bounds for the containment problem with a variety of query language fragments. In particular, they obtain a Π_2^p complexity bound for the well-designed fragment, restricted to AND and OPTIONAL operators. However, the implementation reported in [Letelier *et al.*, 2012] is no longer available and no implementation is reported in [Pichler and Skritek, 2014].

The line of research followed in [Chekol *et al.*, 2012; 2013] also provides some complexity results. They mainly provide a 2-EXPTIME upper-bound for the containment problem for a fragment of SPARQL queries without the optional operator, but in the presence of RDFS/OWL constraints. In [Chekol *et al.*, 2013] the authors provide a benchmark for the static analysis of SPARQL queries (without the optional operator), and report experimental results that, overall, confirm that SPARQL containment solvers are still in early stage.

To the best of our knowledge, the static detection of the query-update independence for SPARQL is a new topic for which no research work has been reported yet.

3 Approach

We have investigated several directions for developing static analysis techniques using a logical approach.

We first studied a suitable target logic, expressive enough to support the restricted query language features and the static analysis problems taken into consideration.

We started to study the alternation-free fragment of the propositional modal μ-calculus (a.k.a. AF_μ), enriched with (1) nominals, (2) backward modalities, and (3) functional modalities. Unfortunately, this logic with all three features is undecidable. However, the logics with any two features out of the three are decidable [Tanabe *et al.*, 2008]. The choice of this logic seemed natural to deal with problems over graphs (SPARQL works over graphs) and an additional benefit of using a μ-encoding seemed to derive from the available of fix-points and modalities for encoding recursion.

We developed a first prototype solver for the μ-calculus with backward modalities and functional modalities. This prototype has been built in an incremental manner. The core of this prototype is a solver for the K-logic, that does not support fix-points, backward and functional modalities.

We show that the K-logic is powerful enough to deal with query containment for well-designed SPARQL queries, restricted to AND and OPTIONAL operators. As we mentioned, the containment complexity for this fragment is Π_2^P and the K-logic admits PSPACE-COMPLETE satisfiability solvers that are implemented in practice. Hence, our approach opens a way to take advantage of these implementations. In [Letelier *et al.*, 2012], the authors provide a natural tree representation for the well-designed SPARQL queries. We introduce a translation of this tree representation into K-formulas and we show how query containment can be reduced to K unsatisfiability.

After these works, we have started to investigate the query-update independence problem for SPARQL. In order to handle the independence problem, we needed to define a common semantics for the updates and the queries, that take into account information about their pattern structures. To achieve this requirement, we extended the relational algebra in [Cyganiak, 2005] for SPARQL 1.1 Update. Starting from this common semantics, we investigated how a notion of independence can be defined in the SPARQL context. We then discussed several possible formal grounds on which static analysis tools can be built to effectively check for independence.

4 Expected Contributions

To summarize, our contributions aim at providing:

- a neat linear translation of well-designed queries in terms of formulas expressed in the modal logic K and a formulation of the containment problem as unsatisfiability in logic;
- a first study towards the query-update independence problem for SPARQL, a necessary and sufficient condition definition for the independence, a discussion about the difficulties introduced by the SPARQL's open world assumption, and finally a formulation of the query-update independence problem as a validity problem in K;
- a prototype of the whole approach. This prototype consists in the assembly of a compiler and a solver. The solver provides several features like functional modalities and backward modalities;
- a relevant benchmark for SPARQL queries, in order to test containment and to detect query-update independence. The benchmark is based on the Berlin Benchmark for SPARQL, enriched with ad-hoc queries, containing the optional operator, to test the containment problem, and with update queries to test the query-update independence problem.
- a comparative study of the behaviour of different third-part solvers (K-solver, tree solver, μ-solver,...) over our benchmark. The logics decided by these solvers have different expressive powers. We are interested in studying a strategy to select the most efficient solver in relation to the peculiarities of the input queries (cyclic/acyclic, well-designed, conjunctive, etc.).
- a study of the static analysis problems under RDFS/OWL schema constraints. In particular, we plan to investigate the query-update independence problem under OWL schemas.

5 Perspectives

The prototypes that we have developed can be further improved and turned into a general purpose framework. This framework should accept several query language fragments as input and, for each of these, it should provide several functions for the static and dynamic analysis of SPARQL queries. It should integrate several solvers and switch to the best one according to the input query peculiarities.

References

[Chekol *et al.*, 2012] Melisachew Wudage Chekol, Jérôme Euzenat, Pierre Genevès, and Nabil Layaïda. SPARQL query containment under RDFS entailment regime. In *IJCAR'12*, pages 134–148, 2012.

[Chekol *et al.*, 2013] Melisachew Wudage Chekol, Jérôme Euzenat, Pierre Genevès, and Nabil Layaïda. Evaluating and benchmarking SPARQL query containment solvers. In *ISWC'13: Proceedings of the 12th International Semantic Web Conference*, pages 408–423, 2013.

[Cyganiak, 2005] Richard Cyganiak. A relational algebra for SPARQL. 2005.

[Letelier *et al.*, 2012] Andrs Letelier, Jorge Prez, Reinhard Pichler, and Sebastian Skritek. Static analysis and optimization of semantic web queries. In *PODS'12*, pages 89–100, 2012.

[Pichler and Skritek, 2014] Reinhard Pichler and Sebastian Skritek. Containment and equivalence of well-designed sparql. In *PODS '14*, pages 39–50, 2014.

[Tanabe *et al.*, 2008] Yoshinori Tanabe, Koichi Takahashi, and Masami Hagiya. A decision procedure for alternation-free modal μ-calculi. In *Advances in Modal Logic*, pages 341–362, 2008.