THE UNIVERSITY of EDINBURGH

Edinburgh Research Explorer

First-Order Rewritability of Frontier-Guarded Ontology-Mediated Queries

Citation for published version:
Barceló, P, Berger, G, Lutz, C & Pieris, A 2018, First-Order Rewritability of Frontier-Guarded Ontology-Mediated Queries. in *Proceedings of the 12th Alberto Mendelzon International Workshop on Foundations of Data Management: Cali, Colombia, May 21-25, 2018.* CEUR Workshop Proceedings (CEUR-WS.org), 12th Alberto Mendelzon International Workshop on Foundations of Data Management, Cali, Colombia, 21/05/18.

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Publisher's PDF, also known as Version of record

Published In:
Proceedings of the 12th Alberto Mendelzon International Workshop on Foundations of Data Management

OPEN ACCESS

First-Order Rewritability of Frontier-Guarded Ontology-Mediated Queries[⋆]

Pablo Barceló[1], Gerald Berger[2], Carsten Lutz[3], and Andreas Pieris[4]

[1] University of Chile `pbarcelo@dcc.uchile.cl`
[2] Vienna University of Technology `gberger@dbai.tuwien.ac.at`
[3] University of Bremen `clu@uni-bremen.de`
[4] University of Edinburgh `apieris@inf.ed.ac.uk`

1 Introduction

Ontology-based data access (OBDA) is a successful application of knowledge representation and reasoning technologies in information management systems. One premier goal is to facilitate access to data that is heterogeneous and incomplete. This is achieved via an ontology that enriches the user query, typically a union of conjunctive queries, with domain knowledge. It turned out that the ontology and the user query can be seen as two components of one composite query, called *ontology-mediated query* (OMQ). The problem of answering OMQs is thus central to OBDA.

There is a consensus that the required level of scalability in OMQ answering can be achieved by using standard database management systems. To this end, a standard approach used nowadays is *query rewriting*: the ontology $\mathcal{O}$ and the database query q are combined into a new query $q_{\mathcal{O}}$, the so-called *rewriting*, which gives the same answer as the OMQ consisting of $\mathcal{O}$ and q over all input databases. It is of course essential that the rewriting $q_{\mathcal{O}}$ is expressed in a language that can be handled by standard database systems. The typical language that is considered is the class of first-order (FO) queries.

In this work, we focus on two central OMQ languages based on *guarded* and *frontier-guarded tuple-generating dependencies* (TGDs), and we study the problem whether an OMQ is FO-rewritable, i.e, it can be equivalently expressed as a first-order query. Recall that a guarded (resp., frontier-guarded) TGD is a sentence of the form $\forall \bar{x}, \bar{y}\,(\varphi(\bar{x}, \bar{y}) \to \exists \bar{z}\,\psi(\bar{x}, \bar{z}))$, where φ and ψ are conjunctions of relational atoms, and φ has an atom that contains all the variables $(\bar{x} \cup \bar{y})$ (resp., $\bar{x}$) [1, 8]. Our goal is to develop specially tailored techniques that allow us to understand the above non-trivial problem, and also to pinpoint its computational complexity. To this end, as we discuss below, we follow two different approaches. Our results can be summarized as follows:

▶ We first focus on the simpler OMQ language based on guarded TGDs and atomic queries, and, in Section 2, we provide a characterization of FO-rewritability that forms the basis for applying tree automata techniques.

▶ We then exploit, in Section 3, standard two-way alternating parity tree automata. In particular, we reduce our problem to the problem of checking the finiteness of the language of an automaton. The reduction relies on a refined version of the characterization of FO-rewritability established in Section 2. This provides a transparent solution to our problem based on standard tools, but it does not lead to an optimal result.

[⋆] This is a short version of [3]

▶ Towards an optimal result, we use, in Section 4, a more sophisticated automata model, known as cost automata. In particular, we reduce our problem to the problem of checking the boundedness of a cost automaton. This allows us to show that FO-rewritability for OMQs based on guarded TGDs and atomic queries is in 2ExpTime, and in ExpTime for predicates of bounded arity. The complexity analysis relies on an intricate result on the boundedness problem for a certain class of cost automata [5, 9].

▶ Finally, in Section 5, by using the results of Section 4, we provide a complete picture for the complexity of our problem, i.e., deciding whether an OMQ based on (frontier-)guarded TGDs and arbitrary (unions of) conjunctive queries is FO-rewritable.

2 Semantic Characterization

We first need to recall the basics on ontology-mediated querying. An *ontology-mediated query* (OMQ) is a triple $Q = (\mathbf{S}, \mathcal{O}, q)$, where $\mathbf{S}$ is a (non-empty) schema (the *data schema*), $\mathcal{O}$ is a set of TGDs (the *ontology*), and q is a UCQ over $\mathbf{S} \cup \mathsf{sig}(\mathcal{O})$, where $\mathsf{sig}(\mathcal{O})$ is the set of predicates occurring in $\mathcal{O}$. The semantics of Q is given in terms of *certain answers*. More precisely, the *evaluation* of Q over an $\mathbf{S}$-database $\mathfrak{D}$, that is, a finite set of atoms of the form $R(\bar{t})$, where $R \in \mathbf{S}$ and $\bar{t}$ is a tuple of constants, denoted $Q(\mathfrak{D})$, is defined as the certain answers to q w.r.t. $\mathfrak{D}$ and $\mathcal{O}$. We write (C, Q) for the class of OMQs $(\mathbf{S}, \mathcal{O}, q)$, where $\mathcal{O}$ falls in the class of TGDs C, and q in the query language Q. For example, $(\mathsf{G}, \mathsf{AQ}_0)$ is the class of OMQs where the ontology falls in the class of guarded TGDs (G), and the query falls in the class of propositional atomic queries (AQ_0). Analogously, we define the OMQ languages based on frontier-guarded TGDs (FG), conjunctive queries (CQ), and unions thereof (UCQ).

We proceed to give a characterization of FO-rewritability of OMQs from $(\mathsf{G}, \mathsf{AQ}_0)$ in terms of the existence of certain tree-like databases. Our characterization is related to, but different from characterizations used for OMQs based on DLs such as $\mathcal{EL}$ and $\mathcal{ELI}$ [6, 7]. In what follows, we write $\mathsf{wd}(\mathbf{S})$ for the *width* of $\mathbf{S}$, i.e., the maximum arity among all predicates of $\mathbf{S}$, and $T_{\mathbf{S}}$ for the integer $\max\{0, \mathsf{wd}(\mathbf{S}) - 1\}$. Given a database $\mathfrak{D}$ and an OMQ $Q \in (\mathsf{G}, \mathsf{AQ}_0)$, we write $\mathfrak{D} \models Q$ for the fact that $Q(\mathfrak{D})$ is non-empty.

Theorem 1. *Consider an OMQ $Q \in (\mathsf{G}, \mathsf{AQ}_0)$ with data schema $\mathbf{S}$. Q is FO-rewritable iff there is a $k \geq 0$ such that, for every $\mathbf{S}$-database $\mathfrak{D}$ of tree-width at most $T_{\mathbf{S}}$, if $\mathfrak{D} \models Q$, then there is a $\mathfrak{D}' \subseteq \mathfrak{D}$ with at most k facts such that $\mathfrak{D}' \models Q$.*

3 Alternating Tree Automata Approach

In this section, we exploit the well-known algorithmic tool of *two-way alternating parity tree automata* (2ATA) over finite trees of bounded degree (see, e.g., [10]) and prove that the problem of deciding whether a query from $(\mathsf{G}, \mathsf{AQ}_0)$ is FO-rewritable can be solved in triple exponential time. Although this result is not optimal, our construction provides a transparent solution based on standard tools.

The idea behind our solution is, given a query $Q \in (\mathsf{G}, \mathsf{AQ}_0)$, to devise a 2ATA $\mathcal{B}_Q$ such that Q is FO-rewritable iff the language accepted by $\mathcal{B}_Q$ is finite. This is a standard idea with roots in the study of the boundedness problem for *monadic Datalog*

(see e.g., [11]). In particular, we show that for a query $Q \in (\mathsf{G}, \mathsf{AQ}_0)$ with data schema $\mathbf{S}$, there is a 2ATA $\mathcal{B}_Q$ on trees of degree at most $2^{\mathsf{wd}(\mathbf{S})}$ such that Q is FO-rewritable iff the language of $\mathcal{B}_Q$ is finite. The state set of $\mathcal{B}_Q$ is of size double exponential in $\mathsf{wd}(\mathbf{S})$, and of size exponential in $|\mathbf{S} \cup \mathsf{sig}(\mathcal{O})|$. Moreover, $\mathcal{B}_Q$ can be constructed in time double exponential in the size of Q. This result relies on a refined version of the semantic characterization provided by Theorem 1. The key idea is to construct a 2ATA $\mathcal{B}_Q$ whose language corresponds to suitable encodings of databases $\mathfrak{D}$ of bounded tree-width that "minimally" entail Q, i.e., $\mathfrak{D} \models Q$, but if we remove any atom from $\mathfrak{D}$, then Q is no longer entailed. Having the above result in place, we can then exploit standard techniques on 2ATAs [11, 12] in order to establish the following result:

Theorem 2. *The problem of deciding whether a query $Q \in (\mathsf{G}, \mathsf{AQ}_0)$ is FO-rewritable is in* 3EXPTIME*, and in* 2EXPTIME *for predicates of bounded arity.*

4   Cost Automata Approach

We proceed to study FO-rewritability for $(\mathsf{G}, \mathsf{AQ}_0)$ using the more sophisticated model of cost automata. Cost automata extend traditional automata (on words, trees, etc.) by providing counters that can be manipulated at each transition. Instead of assigning a Boolean value to each input structure (indicating whether the input is accepted or not), these automata assign a value from $\mathbb{N} \cup \{\infty\}$ to each input. Here, we focus on cost automata that work on finite trees of unbounded degree, and allow for two-way movements. This allows us to improve the complexity obtained in Theorem 2:

Theorem 3. *The problem of deciding whether a query $Q \in (\mathsf{G}, \mathsf{AQ}_0)$ is FO-rewritable is in* 2EXPTIME*, and in* EXPTIME *for predicates of bounded arity.*

As above, we develop a semantic characterization, by refining the semantic characterization of Theorem 1, that relies on a minimality criterion for trees accepted by cost automata. The extra features provided by cost automata allow us to deal with such a minimality criterion in a more efficient way than standard 2ATAs. While our application of cost automata is transparent, the complexity analysis relies on an intricate result on the boundedness problem for a certain class of cost automata from [5, 9].

5   The Complete Picture

We show the following result:

Theorem 4. *The problem of deciding whether a query $Q \in (\mathsf{C}, \mathsf{Q})$ is FO-rewritable is*

– 2EXPTIME-*complete, for $\mathsf{C} = \mathsf{FG}$ and $\mathsf{Q} \in \{\mathsf{UCQ}, \mathsf{CQ}, \mathsf{AQ}_0\}$, even for predicates of arity at most two.*
– 2EXPTIME-*complete, for $\mathsf{C} = \mathsf{G}$ and $\mathsf{Q} \in \{\mathsf{UCQ}, \mathsf{CQ}\}$, even for predicates of arity at most two.*
– 2EXPTIME-*complete, and* EXPTIME-*complete for predicates of bounded arity (even if the predicates have arity at most two), for $\mathsf{C} = \mathsf{G}$ and $\mathsf{Q} = \mathsf{AQ}_0$.*

Lower bounds. The 2ExpTime-hardness in the first and the second items is inherited from [6], which focuses on OMQs based on $\mathcal{ELI}$ and CQs. For the 2ExpTime-hardness in the third item, we exploit the fact that containment for OMQs from $(\mathsf{G}, \mathsf{AQ}_0)$ is 2ExpTime-hard, even if the right-hand side query is FO-rewritable; this is implicit in [4]. The ExpTime-hardness is inherited from [7], where it is shown that deciding FO-rewritability for OMQs based on $\mathcal{EL}$ and atomic queries is ExpTime-hard.

Upper bounds. The fact that for predicates of bounded arity FO-rewritability for OMQs from $(\mathsf{G}, \mathsf{AQ}_0)$ is in ExpTime is obtained from Theorem 3. It remains to show that the problem for $(\mathsf{FG}, \mathsf{UCQ})$ is in 2ExpTime. We first reduce FO-rewritability for $(\mathsf{FG}, \mathsf{UCQ})$ to FO-rewritability for $(\mathsf{FG}, \mathsf{AQ}_0)$ via an easy polynomial time reduction, and then show that the latter is in 2ExpTime. This is achieved by reducing the problem to FO-rewritability for $(\mathsf{G}, \mathsf{AQ}_0)$, and then apply Theorem 3. This reduction relies on a technique known as *treeification*, and is inspired by a construction from [2].

References

1. Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, and Eric Salvat. On rules with existential variables: Walking the decidability line. *Artif. Intell.*, 175(9-10):1620–1654, 2011.
2. Vince Bárány, Balder ten Cate, and Luc Segoufin. Guarded negation. *J. ACM*, 62(3):22:1–22:26, 2015.
3. Pablo Barceló, Gerald Berger, Carsten Lutz, and Andreas Pieris. First-order rewritability of frontier-guarded ontology-mediated queries. In *IJCAI*, 2018. To appear.
4. Pablo Barceló, Miguel Romero, and Moshe Y. Vardi. Does query evaluation tractability help query containment? In *PODS*, pages 188–199, 2014.
5. Michael Benedikt, Balder ten Cate, Thomas Colcombet, and Michael Vanden Boom. The complexity of boundedness for guarded logics. In *LICS*, pages 293–304, 2015.
6. Meghyn Bienvenu, Peter Hansen, Carsten Lutz, and Frank Wolter. First order-rewritability and containment of conjunctive queries in horn description logics. In *IJCAI*, pages 965–971, 2016.
7. Meghyn Bienvenu, Carsten Lutz, and Frank Wolter. First-order rewritability of atomic queries in horn description logics. In *IJCAI*, 2013.
8. Andrea Calì, Georg Gottlob, and Michael Kifer. Taming the infinite chase: Query answering under expressive relational constraints. *J. Artif. Intell. Res.*, 48:115–174, 2013.
9. Thomas Colcombet and Nathanaël Fijalkow. The bridge between regular cost functions and omega-regular languages. In *ICALP*, pages 126:1–126:13, 2016.
10. Stavros S. Cosmadakis, Haim Gaifman, Paris C. Kanellakis, and Moshe Y. Vardi. Decidable optimization problems for database logic programs (preliminary report). In *STOC*, pages 477–490, 1988.
11. Moshe Y. Vardi. Automata theory for database theoreticans. In *Theoretical Studies in Computer Science*, pages 153–180, 1992.
12. Moshe Y. Vardi. Reasoning about the past with two-way automata. In *ICALP*, pages 628–641, 1998.