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

Pablo Barceló¹, Gerald Berger², Carsten Lutz³, and Andreas Pieris⁴

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 \( O \) and the database query \( q \) are combined into a new query \( q_O \), the so-called rewriting, which gives the same answer as the OMQ consisting of \( O \) and \( q \) over all input databases. It is of course essential that the rewriting \( q_O \) 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 \( \forall x, y \left( \varphi(x, y) \rightarrow \exists z \psi(x, z) \right) \), where \( \varphi \) and \( \psi \) are conjunctions of relational atoms, and \( \varphi \) has an atom that contains all the variables \( (x, y) \) (resp., \( x \) \( ) \)   [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 $2\text{Exptime}$, and in $\text{Exptime}$ 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 $Q = (S, O, q)$, where $S$ is a (non-empty) schema (the data schema), $O$ is a set of TGDs (the ontology), and $q$ is a UCQ over $S \cup \text{sig}(O)$, where $\text{sig}(O)$ is the set of predicates occurring in $O$. The semantics of $Q$ is given in terms of certain answers. More precisely, the evaluation of $Q$ over an $S$-database $D$, that is, a finite set of atoms of the form $R(t)$, where $R \in S$ and $t$ is a tuple of constants, denoted $Q(D)$, is defined as the certain answers to $q$ w.r.t. $D$ and $O$. We write $(C, Q)$ for the class of OMQs $(S, O, q)$, where $O$ falls in the class of TGDs $C$, and $q$ in the query language $Q$. For example, $(G, AQ_0)$ 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 $(G, AQ_0)$ 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 $EL$ and $ELI$ [6, 7]. In what follows, we write $wd(S)$ for the width of $S$, i.e., the maximum arity among all predicates of $S$, and $T_S$ for the integer $\max\{0, wd(S) - 1\}$. Given a database $D$ and an OMQ $Q \in (G, AQ_0)$, we write $D \models Q$ for the fact that $Q(D)$ is non-empty.

Theorem 1. Consider an OMQ $Q \in (G, AQ_0)$ with data schema $S$. $Q$ is FO-rewritable iff there is a $k \geq 0$ such that, for every $S$-database $D$ of tree-width at most $T_S$, if $D \models Q$, then there is a $D' \subseteq D$ with at most $k$ facts such that $D' \models Q$.

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 $(G, AQ_0)$ 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 $Q \in (G, AQ_0)$, to devise a 2ATA $B_Q$ such that $Q$ is FO-rewritable iff the language accepted by $B_Q$ is finite. This is a standard idea with roots in the study of the boundedness problem for monadic Datalog.
In particular, we show that for a query \( Q \in (G, AQ_0) \) with data schema \( S \), there is a 2ATA \( B_Q \) on trees of degree at most \( 2^{\text{wd}(S)} \) such that \( Q \) is FO-rewritable iff the language of \( B_Q \) is finite. The state set of \( B_Q \) is of size double exponential in \( \text{wd}(S) \), and of size exponential in \( |S \cup \text{sig}(O)| \). Moreover, \( B_Q \) 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 \( B_Q \) whose language corresponds to suitable encodings of databases \( D \) of bounded tree-width that “minimally” entail \( Q \), i.e., \( D \models Q \), but if we remove any atom from \( D \), 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 \( Q \in (G, AQ_0) \) is FO-rewritable is in \( 3\text{EXPTIME} \), and in \( 2\text{EXPTIME} \) for predicates of bounded arity.

### 4 Cost Automata Approach

We proceed to study FO-rewritability for \( (G, AQ_0) \) 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 \( Q \in (G, AQ_0) \) is FO-rewritable is in \( 2\text{EXPTIME} \), and in \( \text{EXPTIME} \) 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 \( Q \in (C, Q) \) is FO-rewritable is

- \( 2\text{EXPTIME}\)-complete, for \( C = FG \) and \( Q \in \{ \text{UCQ}, \text{CQ}, AQ_0 \} \), even for predicates of arity at most two.
- \( 2\text{EXPTIME}\)-complete, for \( C = G \) and \( Q \in \{ \text{UCQ}, \text{CQ} \} \), even for predicates of arity at most two.
- \( 2\text{EXPTIME}\)-complete, and \( \text{EXPTIME}\)-complete for predicates of bounded arity (even if the predicates have arity at most two), for \( C = G \) and \( Q = AQ_0 \).
Lower bounds. The $2\text{ExpTIME}$-hardness in the first and the second items is inherited from [6], which focuses on OMQs based on $\mathcal{ELI}$ and CQs. For the $2\text{ExpTIME}$-hardness in the third item, we exploit the fact that containment for OMQs from $(G, AQ_0)$ is $2\text{ExpTIME}$-hard, even if the right-hand side query is FO-rewritable; this is implicit in [4]. The $\text{ExpTIME}$-hardness is inherited from [7], where it is shown that deciding FO-rewritability for OMQs based on $\mathcal{EL}$ and atomic queries is $\text{ExpTIME}$-hard.

Upper bounds. The fact that for predicates of bounded arity FO-rewritability for OMQs from $(G, AQ_0)$ is in $\text{ExpTIME}$ is obtained from Theorem 3. It remains to show that the problem for $(FG, UCQ)$ is in $2\text{ExpTIME}$. We first reduce FO-rewritability for $(FG, UCQ)$ to FO-rewritability for $(FG, AQ_0)$ via an easy polynomial time reduction, and then show that the latter is in $2\text{ExpTIME}$. This is achieved by reducing the problem to FO-rewritability for $(G, AQ_0)$, and then apply Theorem 3. This reduction relies on a technique known as treeification, and is inspired by a construction from [2].

Acknowledgements. Barceló is funded by the Millennium Institute for Foundational Research on Data and Fondecyt grant 1170109. Berger is funded by the Austrian Science Fund (FWF), project number W1255-N23 and DOC fellowship of the Austrian Academy of Sciences. Lutz is funded by the ERC grant 647289 “CODA”. Pieris is funded by the EPSRC programme grant EP/M025268/1 “VADA”.

References