Computing FO-Rewritings in $\mathcal{EL}$ in Practice: from Atomic to Conjunctive Queries

Peter Hansen and Carsten Lutz

University of Bremen, Germany
{hansen, clu}@informatik.uni-bremen.de

Abstract. It has recently been demonstrated in [10] that FO-rewritings of ontology-mediated queries can be efficiently computed in practice, in a sound and complete way, when the ontology is formulated in $\mathcal{EL}$ and the actual query is an atomic query (AQ). In this paper, we show how to lift this approach, which is based on a decomposed version of backwards chaining, from AQs to (rooted) conjunctive queries (rCQs). While we achieve a polynomial time reduction when the quantified parts of the CQ are tree-shaped, a more subtle approach is required in the general case. We conduct experiments based on real-world ontologies which show promising results.

1 Introduction

One of the most important technical tools in ontology-mediated querying is *query rewriting*: reformulate a given ontology-mediated query (OMQ) in an equivalence-preserving way in a query language that is supported by a database system used to store the data. Since SQL is the dominating query language in conventional database systems, rewriting into SQL and into first-order logic (FO) as its logical core has attracted particularly much attention [2, 3, 4, 5, 6, 8, 10, 11]. In fact, the DL-Lite family of description logics (DLs) was invented specifically with the aim to guarantee that FO-rewritings of OMQs (whose TBox is formulated in DL-Lite) always exist [1, 6], but is rather restricted in expressive power. For essentially all other DLs, there are OMQs which cannot be equivalently rewritten into an FO query. However, ontologies used in real-world applications tend to have a very simple structure, and consequently, FO-rewritings of practically relevant OMQs might exist in the majority of cases. This hope was confirmed in an experimental evaluation carried out in the context of the description logic $\mathcal{EL}$, where less than 1% of the considered queries was found to be not FO-rewritable [10]; moreover, most of the negative cases seemed to be due to modeling mistakes in the ontology.

In this paper, we focus on the description logic $\mathcal{EL}$ and aim to push the frontier of efficiently computing FO-rewritings of OMQs from *atomic queries (AQs)* to *conjunctive queries (CQs)*. As usual, we use $(\mathcal{L}, \mathcal{Q})$ to denote the OMQ language that consists of all OMQs $(\mathcal{T}, \Sigma, q)$ where $\mathcal{T}$ is a TBox formulated in the description logic $\mathcal{L}$ and q is a query formulated in the query language $\mathcal{Q}$ (and Σ is an ABox signature). It has been shown in [5] that for OMQs from $(\mathcal{EL}, \text{AQ})$, it is ExpTime-complete to decide FO-rewritability. Combining the techniques from [5] and the backwards chaining approach to query rewriting brought forward e.g. in [7,11], a practical algorithm for computing FO-rewritings of OMQs from (an extension of) $(\mathcal{EL}, \mathrm{AQ})$ was then developed in [10]. It is based on a *decomposed version* of backwards chaining that implements a form of structure sharing. This algorithm was implemented in the *Grind* system and shown to perform very well in practice: on 10989 inputs and with a timeout of 30 seconds, the algorithm terminated on all but 127 inputs and needed only 1.5h execution time in total (an average of 0.5 seconds per input). It is important to remark that the algorithm is *complete*, that is, it computes an FO-rewriting whenever there is one and reports failure otherwise.

We intend to lift this approach from AQs to CQs. Note that it was shown in [4] that FO-rewritability in $(\mathcal{EL}, \mathrm{CQ})$ is still ExpTime-complete. Since the details of the decomposed algorithm from [10] are already rather complex, one would ideally hope to achieve a black box (and practically feasible) polynomial time reduction of FO-rewritability in $(\mathcal{EL}, \mathrm{CQ})$ to $(\mathcal{EL}, \mathrm{AQ})$. However, naive such reductions fail. In particular, FO-rewritability of all AQs that occur in a CQ q are neither a sufficient nor a necessary condition for q to be FO-rewritable. For example, when

$$\mathcal{T} = \{\exists r.A \sqsubseteq A,\ \exists s.\top \sqsubseteq A\} \quad \Sigma = \{A, r, s\} \quad q(x) = \exists y\,(A(x) \wedge s(x,y))$$

then $Q = (\mathcal{T}, \Sigma, q)$ is FO-rewritable into $\exists y\, s(x,y)$, but the only AQ $A(x)$ that occurs in q is not FO-rewritable. In fact, a black box reduction does not seem to be possible in general. Thus, we consider mildly restricted forms of CQs and exhibit reductions that are not completely black box, but make certain assumptions on the algorithm used to compute FO-rewritings in $(\mathcal{EL}, \mathrm{AQ})$—all of them satisfied by the decomposed backwards chaining algorithm implemented in Grind.

We first consider the class of *tree-quantified CQs (tqCQs)* in which the quantified parts of the CQ are tree-shaped. In this case, we indeed achieve a black box polynomial time reduction for FO-rewritability. To also transfer actual FO-rewritings from the OMQ constructed in the reduction to the original OMQ, we make the assumption that the rewriting of the former takes the form of a UCQ in which every CQ is tree-shaped and that, in a certain sense made precise in the paper, atoms are never introduced into the rewriting 'without a reason'. Both conditions are very natural in the context of backwards chaining and satisfied by the decomposed algorithm.

We then move to *rooted CQs (rCQs)* in which every quantified variable must be reachable from some answer variable (in the query graph). We consider this a mild restriction and expect that almost all queries in practical applications will be rCQs. In the rCQ case, we do not achieve a black box reduction. Instead, we assume that FO-rewritings of the constructed OMQs from $(\mathcal{EL}, \mathrm{AQ})$ are obtained from a certain straightforward backwards chaining algorithm or a refinement thereof as implemented in the Grind system. We then show how to combine the construction of (several) OMQs from $(\mathcal{EL}, \mathrm{AQ})$, similar to what we have done in the black box reduction in the tqCQ case, with a modification of the assumed algorithm to decide FO-rewritability in $(\mathcal{EL}, \text{rCQ})$ and to construct actual rewritings. The approach involves exponential blowups, but only in parameters that we expect to be very small in practical cases and that, in particular, only depend on the actual query contained in the OMQ but not on the TBox.

We have implemented our approach in the Grind system and carried out experiments on five real-world ontologies with 10 hand-crafted CQs for each. The average runtimes are between 0.5 and 19 seconds (depending on the ontology), which we consider reasonably short given that we are dealing with a complex static analysis problem. For the proofs, see the Appendix of [9], available at http://www.cs.uni-bremen.de/tdki/research/papers.html.

2 Preliminaries

We use standard notation for $\mathcal{EL}$-TBoxes (sets of concept inclusions), for ABoxes, for conjunctive queries (CQs), and for unions thereof (UCQs), see for example [4]. We do not assume any normal form for $\mathcal{EL}$-TBoxes $\mathcal{T}$. With $\mathsf{Ind}(\mathcal{A})$, we denote the set of individual names in the ABox $\mathcal{A}$. As usual in the context of ontology-mediated querying, ABoxes cannot contain compound concepts, but only concept names. Recall that an *atomic query (AQ)* takes the form $A(x)$, A a concept name, and that a *signature* is a set of concept and role names.

Unless noted otherwise, we allow equality in CQs, but we assume w.l.o.g. that equality atoms contain only answer variables, and that when $x = y$ is an equality atom in q, then y does not occur in any other atoms in q. Other occurences of equality can be eliminated by identifying variables. With $\mathsf{var}(q)$, we denote the set of all variables used in the CQ q and use $\mathsf{avar}(q)$ for the set of all answer variables. We do not distinguish between a CQ and the set of atoms in it and associate with each CQ q a directed graph $G_q := (\mathsf{var}(q), \{(x, y) \mid r(x, y) \in q\})$, defined in the expected way (equality atoms are not reflected). A CQ q is *tree-shaped* if G_q is a directed tree and $r(x, y), s(x, y) \in q$ implies $r = s$. A *tree CQ (tCQ)* is a tree-shaped CQ with the root the only answer variable, and a *tree UCQ (tUCQ)* is a disjunction of tree CQs. Clearly, every $\mathcal{EL}$ concept can be viewed as a tCQ and vice versa, and we will not always distinguish between the two representations. For example, we might write $\exists r.q$ to denote an $\mathcal{EL}$ concept when q is a tree-shaped CQ. If convenient, we also view a CQ q as an ABox $\mathcal{A}_q$ which is obtained from q by dropping equality atoms and then replacing each variable with an individual (not distinguishing answer variables from quantified variables). A *rooted CQ (rCQ)* is a CQ q such that in the undirected graph induced by G_q, every quantified variable is reachable from some answer variable. A *tree-quantified CQ (tqCQ)* is an rCQ q such that after removing all atoms $r(x, y)$ with $x, y \in \mathsf{avar}(q)$, we obtain a disjoint union of tCQs. We call these tCQs the *tCQs in q*.

An *ontology-mediated query (OMQ)* is a triple $Q = (\mathcal{T}, \Sigma, q)$ where $\mathcal{T}$ is a TBox, Σ an ABox signature, and q a CQ. The semantics is defined in the standard way via certain answers. In particular, we write $\mathcal{A} \models Q(\mathbf{a})$ if $\mathbf{a}$ is a certain answer to the OMQ Q on the ABox $\mathcal{A}$; we again refer to [4] for full details.

We use $(\mathcal{EL}, \mathrm{AQ})$ to denote the set of OMQs where $\mathcal{T}$ is formulated in $\mathcal{EL}$ and q is an AQ, and similarly for $(\mathcal{EL}, \mathrm{CQ})$, $(\mathcal{EL}, \mathrm{rCQ})$, and so on. We do generally not allow equality in CQs that are part of an OMQ.

An OMQ $Q = (\mathcal{T}, \Sigma, q)$ is *FO-rewritable* if there is a first-order (FO) formula φ such that $\mathcal{A} \models Q(\mathbf{a})$ iff $\mathcal{A} \models \varphi(\mathbf{a})$ for all Σ-ABoxes $\mathcal{A}$. In this case, φ is an *FO-rewriting of* Q. When φ happens to be a UCQ, we speak of a *UCQ-rewriting* and likewise for other classes of queries. It is known that FO-rewritability coincides with UCQ-rewritability for OMQs from $(\mathcal{EL}, \mathrm{CQ})$ [3, 5]; note that equality is important here as, for example, the OMQ $(\{B \sqsubseteq \exists r.A\}, \{B, r\}, q)$ with $q(x, y) = \exists z(r(x, z) \wedge r(y, z) \wedge A(z))$ rewrites into the UCQ $q \vee (B(x) \wedge x = y)$.

We shall sometimes refer to the problem of *(query) containment* between two OMQs $Q_1 = (\mathcal{T}_1, \Sigma, q_1)$ and $Q_2 = (\mathcal{T}_2, \Sigma, q_2)$; we say Q_1 *is contained in* Q_2 if $\mathcal{A} \models Q_1(\mathbf{a})$ implies $\mathcal{A} \models Q_2(\mathbf{a})$ for all Σ-ABoxes $\mathcal{A}$ and $\mathbf{a} \subseteq \mathsf{Ind}(\mathcal{A})$. If both OMQs are from ($\mathcal{EL}$, rCQ) and $\mathcal{T}_1 = \mathcal{T}_2 = \mathcal{T}$, then we denote this with $q_1 \subseteq_{\mathcal{T}} q_2$.

We now introduce two more involved notions that are central to the technical constructions in Section 4, fork rewritings and splittings. Both notions have been used before in the context of ontology-mediated querying, see for example [12,13].

Definition 1 (Fork rewriting). *Let q_0 be a CQ.* Obtaining a CQ q from q_0 by fork elimination *means to choose two atoms $r(x_0, y)$ and $r(x_1, y)$ with y an existentially quantified variable, then to replace every occurrence of x_{1-i} with x_i where $i \in \{0, 1\}$ is chosen such that $x_i \in \mathsf{avar}(q_0)$ if any of x_0, x_1 is an answer variable, and to finally add the atom $x_i = x_{1-i}$ if $x_{1-i} \in \mathsf{avar}(q_0)$. When q can be obtained from q_0 by repeated (but not necessarily exhaustive) fork elimination, then q is a* fork rewriting *of q_0.*

For a CQ q and $V \subseteq \mathsf{var}(q)$, we use $q|_V$ to denote the restriction of q to the variables in V.

Definition 2 (Splitting). *Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox, q a CQ, and $\mathcal{A}$ an ABox. A* splitting *of q w.r.t. $\mathcal{A}$ and $\mathcal{T}$ is a tuple $\Pi = \langle R, S_1, \dots, S_\ell, r_1, \dots, r_\ell, \mu, \nu \rangle$, where $R, S_1, \dots, S_n$ is a partitioning of $\mathsf{var}(q)$, $r_1, \dots, r_\ell$ are role names, $\mu : \{1, \dots, \ell\} \to R$ assigns to each set S_i a variable from R, $\nu : R \to \mathsf{Ind}(\mathcal{A})$, and the following conditions are satisfied:*

1. *$\mathsf{avar}(q) \subseteq R$ and $x = y \in q$ implies $\nu(x) = \nu(y)$;*
2. *if $r(x, y) \in q$ with $x, y \in R$, then $r(\nu(x), \nu(y)) \in \mathcal{A}$;*
3. *$q|_{S_i}$ is tree-shaped and can thus be seen as an $\mathcal{EL}$ concept $C_{q|_{S_i}}$, for $1 \leq i \leq \ell$;*
4. *if $r(x, x') \in q$ then either (i) x, x' belong to the same set $R, S_1, \dots, S_\ell$, or (ii) $x \in R$ and, for some i, $r = r_i$ and x' root of $q|_{S_i}$.*

The following lemma illustrates the combined use and raison d'être of both fork rewritings and splittings. A proof is standard and omitted, see for example [13]. It does rely on the existence of *forest models* for ABoxes and $\mathcal{EL}$-TBoxes, that is, for every ABox $\mathcal{A}$ and TBox $\mathcal{T}$, there is a model $\mathcal{I}$ whose shape is that of $\mathcal{A}$ with a directed (potentially infinite) tree attached to each individual.

Lemma 1. *Let $Q = (\mathcal{T}, \Sigma, q_0)$ be an OMQ from $(\mathcal{EL}, CQ)$, $\mathcal{A}$ a Σ-ABox, and $\mathbf{a} \subseteq \mathsf{Ind}(\mathcal{A})$. Then $\mathcal{A} \models Q(\mathbf{a})$ iff there exists a fork rewriting q of q_0 and a splitting $\langle R, S_1, \ldots, S_\ell, r_1, \ldots, r_\ell, \mu, \nu\rangle$ of q w.r.t. $\mathcal{A}$ and $\mathcal{T}$ such that the following conditions are satisfied: (1) $\nu(\mathbf{x}) = \mathbf{a}$, $\mathbf{x}$ the answer variables of q_0; (2) if $A(x) \in q$ and $x \in R$, then $\mathcal{A}, \mathcal{T} \models A(\nu(x))$; (3) $\mathcal{A}, \mathcal{T} \models \exists r_i.C_{q|_{S_i}}(\nu(\mu(i)))$ for $1 \leq i \leq \ell$.*

3 Tree-quantified CQs

We provide a polynomial time reduction from FO-rewritability in $(\mathcal{EL}, \text{tqCQ})$ to FO-rewritability in $(\mathcal{EL}, \text{AQ})$ and, making only very mild assumptions on the algorithm used for solving the latter problem, show that rewritings of the OMQ produced in the reduction can be transformed in a straightforward way into rewritings of the original OMQ. The mild assumptions are that the algorithm produces a tUCQ-rewriting and that, informally, when constructing the tCQs of the tUCQ-rewriting it never introduces atoms 'without a reason'—this will be made precise later.

Let $Q = (\mathcal{T}, \Sigma, q_0)$ be from $(\mathcal{EL}, \text{tqCQ})$. We can assume w.l.o.g. that q_0 contains only answer variables: every tCQ in q with root x can be represented as an $\mathcal{EL}$ concept C and we can replace the tree with the atom $A_C(x)$ (unless it has only a single node) and extend $\mathcal{T}$ with $C \sqsubseteq A_C$ where A_C is a fresh concept name that is not included in Σ. Clearly, the resulting OMQ is equivalent to the original one.

We show how to construct an OMQ $Q' = (\mathcal{T}', \Sigma', q_0')$ from $(\mathcal{EL}, \text{AQ})$ with the announced properties; in particular, Q is FO-rewritable if and only if Q' is. Let $\mathsf{CN}(\mathcal{T})$ and $\mathsf{RN}(\mathcal{T})$ denote the set of concept names and role names that occur in $\mathcal{T}$, and let sub_L denote the of concepts that occur on the left-hand side of a concept inclusion in $\mathcal{T}$, closed under subconcepts. Reserve a fresh concept name A^x for every $A \in \mathsf{CN}(\mathcal{T})$ and $x \in \mathsf{avar}(q_0)$, and a fresh role name r^x for every $r \in \mathsf{RN}(\mathcal{T})$ and $x \in \mathsf{avar}(q_0)$. Set

$$\begin{aligned}\Sigma' = \Sigma \cup \{A^x \mid A \in \mathsf{CN}(\mathcal{T}) \cap \Sigma \text{ and } x \in \mathsf{avar}(q_0)\} \cup \\ \{r^x \mid r \in \mathsf{RN}(\mathcal{T}) \cap \Sigma \text{ and } x \in \mathsf{avar}(q_0)\}.\end{aligned}$$

Additionally reserve a concept name $A^x_{\exists r.E}$ for every concept $\exists r.E \in \mathsf{sub}_L(\mathcal{T})$ and every $x \in \mathsf{avar}(q_0)$. Define

$$\begin{aligned}\mathcal{T}' := \mathcal{T} &\cup \{C^x_L \sqsubseteq D^x_R \mid x \in \mathsf{var}(q_0) \text{ and } C \sqsubseteq D \in \mathcal{T}\} \\ &\cup \{\exists r^x.C \sqsubseteq A^x_{\exists r.C} \mid x \in \mathsf{var}(q_0) \text{ and } \exists r.C \in \mathsf{sub}_L(\mathcal{T})\} \\ &\cup \{C^y_L \sqsubseteq A^x_{\exists r.C} \mid r(x,y) \in q_0 \text{ and } \exists r.C \in \mathsf{sub}_L(\mathcal{T})\} \\ &\cup \{\bigsqcap_{A(x) \in q_0} A^x \sqsubseteq N\}\end{aligned}$$

where for a concept $C = A_1 \sqcap \cdots \sqcap A_n \sqcap \exists r_1.E_1 \sqcap \cdots \sqcap \exists r_m.E_m$, the concepts C^x_L and C^x_R are given by

$$\begin{aligned}C^x_L &= A^x_1 \sqcap \cdots \sqcap A^x_n \sqcap A^x_{\exists r_1.E_1} \sqcap \cdots \sqcap A^x_{\exists r_m.E_m} \\ C^x_R &= A^x_1 \sqcap \cdots \sqcap A^x_n \sqcap \exists r^x_1.E_1 \sqcap \cdots \sqcap \exists r^x_m.E_m\end{aligned}$$

Moreover, set $q'_0 := N(x)$.

Before proving that the constructed OMQ Q' behaves in the desired way, we give some preliminaries. It is known that, if an OMQ from $(\mathcal{EL}, \text{AQ})$ has an FO-rewriting, then it has a tUCQ-rewriting, see for example [5, 10]. A tCQ q is *conformant* if it satisfies the following properties:

1. if $A(x)$ is a concept atom, then either A is of the form B^y and x is the answer variable or A is not of this form and x is a quantified variable;
2. if $r(x, y)$ is a role atom, then either r is of the form s^z and x is the answer variable or r is not of this form and x is a quantified variable.

A *conformant tUCQ* is then defined in the expected way. The notion of conformance captures what we informally described as never introducing atoms into the rewriting 'without a reason'. By the following lemma, FO-rewritability of the OMQs constructed in our reduction implies conformant tUCQ-rewritability, that is, there is indeed no reason to introduce any of the atoms that are forbidden in conformant rewritings.

Lemma 2. *Let Q be from $(\mathcal{EL}, tqCQ)$ and Q' the OMQ constructed from Q as above. If Q' is FO-rewritable, then it is rewritable into a conformant tUCQ.*

When started on an OMQ produced by our reduction, the algorithms presented in [10] and implemented in the Grind system produce a conformant tUCQ-rewriting. Indeed, this can be expected of any reasonable algorithm based on backwards chaining. Let q' be a conformant tUCQ-rewriting of Q'. The *corresponding UCQ for Q* is the UCQ q obtained by taking each CQ from q', replacing every atom $A^x(x_0)$ with $A(x)$ and every atom $r^x(x_0, y)$ with $r(x, y)$, and adding all atoms $r(x, y)$ from q_0 such that both x and y are answer variables. The answer variables in q are those of q_0. Observe that q is a union of tqCQs.

Proposition 1. *Q is FO-rewritable iff Q' is FO-rewritable. Moreover, if q' is a conformant tUCQ-rewriting of Q' and q the corresponding UCQ for Q, then q is a rewriting of Q.*

The proof strategy is to establish the 'moreover' part and to additionally show how certain UCQ-rewritings of Q can be converted into UCQ-rewritings of Q'. More precisely, a CQ q is a *derivative* of q_0 if it results from q_0 by exchanging atoms $A(x)$ for $\mathcal{EL}$ concepts C, seen as tree-shaped CQs rooted in x. We are going to prove the following lemma in Section 4.

Lemma 3. *If an OMQ $(\mathcal{T}, \Sigma, q_0)$ from $(\mathcal{EL}, tqCQ)$ is FO-rewritable, then it has a UCQ-rewriting in which each CQ is a derivative of q_0.*

Let q be a UCQ in which every CQ is a derivative of q_0. Then the *corresponding UCQ for Q'* is the UCQ q' obtained by taking each CQ from q, replacing every atom $A(x)$, x answer variable, with $A^x(x_0)$, every atom $r(x, y)$, x answer variable and y quantified variable, with $r^x(x_0, y)$, and deleting all atoms $r(x_1, x_2)$, x_1, x_2 answer variables. The answer variable in q' is x_0. Note that q' is a tUCQ. To establish the "only if" direction of Proposition 1, we show that when q is a UCQ-rewriting of Q in which every CQ is a derivative of the query q_0, then the corresponding UCQ for Q' is a rewriting of Q'.

4 Rooted CQs

We replace tqCQs with the more general rCQs. In this case, we are not going to achieve a black box reduction, but rely on a concrete algorithm for solving FO-rewritability in $(\mathcal{EL}, \text{AQ})$, namely a straightforward (and not necessarily terminating) backwards chaining algorithm or a (potentially terminating) refinement thereof. We show how to combine the construction of (several) OMQs from $(\mathcal{EL}, \text{AQ})$ with a modification of the assumed algorithm to decide FO-rewritability in $(\mathcal{EL}, \text{rCQ})$ and to construct actual rewritings.

We start with introducing the straightforward backwards chaining algorithm mentioned above which we refer to as bc_{AQ}. Central to bc_{AQ} is a backwards chaining step based on concept inclusions in a TBox. Let C and D be $\mathcal{EL}$ concepts, $E \sqsubseteq F$ a concept inclusion, and $x \in \mathsf{var}(C)$ (where C is viewed as a tree-shaped CQ). Then D is *obtained from C by applying $E \sqsubseteq F$ at x* if D can be obtained from C by

- removing $A(x)$ for all concept names A with $\models F \sqsubseteq A$;
- removing $r(x, y)$ and the tree-shaped CQ G rooted at y when $\models F \sqsubseteq \exists r.G$;
- adding $A(x)$ for all concept names A that occur in E as a top-level conjunct (that is, that are not nested inside existential restrictions);
- adding $\exists r.G$ as a CQ with root x, for each $\exists r.G$ that is a top-level conjunct of E.

Let C and D be $\mathcal{EL}$ concepts. We write $D \prec C$ if D can be obtained from C by removing an existential restriction (not necessarily on top level, and potentially resulting in $D = \top$ when C is of the form $\exists r.E$). We use $\prec^*$ to denote the reflexive and transitive closure of $\prec$ and say that *D is $\prec$-minimal with $\mathcal{T} \models D \sqsubseteq A_0$* if $\mathcal{T} \models D \sqsubseteq A_0$ and there is no $D' \prec D$ with $\mathcal{T} \models D' \sqsubseteq A_0$.

Now we are in the position to describe algorithm bc_{AQ}. It maintains a set M of $\mathcal{EL}$ concepts that represent tCQs. Let $Q = (\mathcal{T}, \Sigma, A_0)$ be from $(\mathcal{EL}, \text{AQ})$. Starting from the set $M = \{A_0\}$, it exhaustively performs the following steps:

1. find $C \in M$, $x \in \mathsf{var}(C)$, a concept inclusion $E \sqsubseteq F \in \mathcal{T}$, and D, such that D is obtained from C by applying $E \sqsubseteq F$ at x;
2. find a $D' \prec^* D$ that is $\prec$-minimal with $\mathcal{T} \models D' \sqsubseteq A_0$, and add D' to M.

Application of these steps might not terminate. We use $\mathsf{bc}_{\text{AQ}}(Q)$ to denote the potentially infinitary UCQ $\bigvee M|_\Sigma$ where M is the set obtained in the limit and $q|_\Sigma$ denotes the restriction of the UCQ q to those disjuncts that only use symbols from Σ. The following is standard to prove, see [10, 11] and Lemma 5 below for similar results.

Lemma 4. *Let Q be an OMQ from $(\mathcal{EL}, AQ)$. If $\mathsf{bc}_{AQ}(Q)$ is finite, then it is a UCQ-rewriting of Q. Otherwise, Q is not FO-rewritable.*

The algorithm for deciding FO-rewritability in $(\mathcal{EL}, \text{AQ})$ presented in [10] and underlying the Grind system can be seen as a refinement of bc_{AQ}. Indeed, that algorithm always terminates and returns $\bigvee M|_\Sigma$ if that UCQ is finite and reports non-FO-rewritability otherwise. Moreover, the UCQ rewriting is represented in a decomposed way and output as a non-recursive Datalog program for efficiency and succinctness. For our purposes, the only important aspect is that, when started on an FO-rewritable OMQ, it computes exactly the UCQ-rewriting $\bigvee M|_\Sigma$.

We next introduce a generalized version $\mathsf{bc}^+_{\mathrm{AQ}}$ of $\mathsf{bc}_{\mathrm{AQ}}$ that takes as input an OMQ $Q = (\mathcal{T}, \Sigma, A_0)$ from $(\mathcal{EL}, \mathrm{AQ})$ and an additional $\mathcal{EL}$-TBox $\mathcal{T}^{\mathsf{min}}$, such that termination and output of $\mathsf{bc}^+_{\mathrm{AQ}}$ agrees with that of $\mathsf{bc}_{\mathrm{AQ}}$ when the input satisfies $\mathcal{T}^{\mathsf{min}} = \mathcal{T}$. Starting from $M = \{A_0\}$, algorithm $\mathsf{bc}^+_{\mathrm{AQ}}$ exhaustively performs the following steps:

1. find $C \in M$, $x \in \mathsf{var}(C)$, a concept inclusion $E \sqsubseteq F \in \mathcal{T}$, and D, such that D is obtained from C by applying $E \sqsubseteq F$ at x;
2. find $D' \prec^* D$ that is $\prec$-minimal with $\mathcal{T}^{\mathsf{min}} \models D' \sqsubseteq A_0$, and add D' to M.

We use $\mathsf{bc}^+_{\mathrm{AQ}}(Q, \mathcal{T}^{\mathsf{min}})$ to denote the potentially infinitary UCQ $\bigvee M|_\Sigma$, M obtained in the limit. Note that $\mathsf{bc}^+_{\mathrm{AQ}}$ uses the TBox $\mathcal{T}$ for backwards chaining and $\mathcal{T}^{\mathsf{min}}$ for minimization while $\mathsf{bc}_{\mathrm{AQ}}$ uses $\mathcal{T}$ for both purposes. The refined version of $\mathsf{bc}_{\mathrm{AQ}}$ implemented in the Grind system can easily be adapted to behave like a terminating version of $\mathsf{bc}^+_{\mathrm{AQ}}$.

Our aim is to convert an OMQ $Q = (\mathcal{T}, \Sigma, q_0)$ from $(\mathcal{EL}, \mathrm{rCQ})$ into a set of pairs $(Q', \mathcal{T}^{\mathsf{min}})$ with Q' an OMQ from $(\mathcal{EL}, \mathrm{AQ})$ and $\mathcal{T}^{\mathsf{min}}$ an $\mathcal{EL}$-TBox such that Q is FO-rewritable iff $\mathsf{bc}^+_{\mathrm{AQ}}(Q', \mathcal{T}^{\mathsf{min}})$ terminates for all pairs $(Q', \mathcal{T}^{\mathsf{min}})$ and, moreover, if this is the case, then the resulting UCQ-rewritings can straightforwardly be converted into a rewriting of Q.

Let $Q = (\mathcal{T}, \Sigma, q_0)$. We construct one pair $(Q_{q_r}, \mathcal{T}^{\mathsf{min}}_{q_r})$ for each fork rewriting q_r of q_0. We use $\mathsf{core}(q_r)$ to denote the minimal set V of variables that contains all answer variables in q_r and such that after removing all atoms $r(x, y)$ with $x, y \in V$, we obtain a disjoint union of tree-shaped CQs. We call these CQs the *trees in* q_r. Intuitively, we separate the tree-shaped parts of q_r from the cyclic part, the latter identified by $\mathsf{core}(q_r)$. This is similar to the definition of tqCQs where, however, cycles cannot involve any quantified variables. In a forest model of an ABox and a TBox as mentioned before Lemma 1, the variables in $\mathsf{core}(q_r)$ must be mapped to the ABox part of the model (rather than to the trees attached to it). Now $(Q_{q_r}, \mathcal{T}^{\mathsf{min}}_{q_r})$ is defined by setting $Q_{q_r} = (\mathcal{T}_{q_r}, \Sigma_{q_r}, N(x))$ and

$$\begin{aligned}\mathcal{T}_{q_r} = \mathcal{T} &\cup \{C^x_R \sqsubseteq D^x_R \mid x \in \mathrm{core}(q_r), C \sqsubseteq D \in \mathcal{T}\}\\ &\cup \{\bigsqcap_{C(x) \text{ a tree in } q_r} C^x_R \sqsubseteq N\}\end{aligned}$$

where C^x_R is defined as in Section 3, and Σ_{q_r} is the extension of Σ with all concept names A^x and role names r^x used in $\mathcal{T}_{q_r}$ such that $A, r \in \Sigma$.

It remains to define $\mathcal{T}^{\mathsf{min}}_{q_r}$, which is $\mathcal{T}_{q_r}$ extended with one concept inclusion for each fork rewriting q of q_0 and each splitting $\Pi = \langle R, S_1, \ldots, S_\ell, r_1, \ldots, r_\ell, \mu, \nu\rangle$ of q w.r.t. $\mathcal{A}_{q_r}$, as follows. For each $x \in \mathsf{avar}(q_r)$, the equality atoms in q_r give rise to an equivalence class $[x]_{q_r}$ of answer variables, defined in the expected way. We only consider the splitting Π of q if it preserves answer variables modulo equality, that is, if $x \in \mathsf{avar}(q)$, then there is a $y \in [x]_{q_r}$ such that $\nu(x) = y$. We then add the inclusion

$$\Big(\bigsqcap_{\substack{A(x)\in q \\ \text{with } x\in R}} A^{\nu(x)} \Big) \sqcap \Big(\bigsqcap_{1\leq i\leq \ell} \exists r_i^{\nu(\mu(i))}.C_{q|_{S_i}} \Big) \sqsubseteq N$$

It can be shown that, summing up over all fork rewritings and splittings, only polynomially many concepts $\exists r_i^{\nu(\mu(i))}.C_{q|_{S_i}}$ are introduced (this is similar to the proof of Lemma 6 in [13]). Note that we do not introduce fresh concept names of the form $A^x_{\exists r.C}$ as in Section 3. This is not necessary here because of the use of fork rewritings and splittings in $\mathcal{T}_{\mathsf{min}}$.

It can be seen that when $\mathsf{bc}^+_{\mathrm{AQ}}(Q_{q_r}, \mathcal{T}^{\mathsf{min}}_{q_r})$ is finite, then it is a conformant tUCQ in the sense of Section 3. Thus, we can also define a *corresponding UCQ q for Q* as in that section, that is, q is obtained by taking each CQ from q', replacing every atom $A^x(x_0)$ with $A(x)$ and every atom $r^x(x_0, y)$ with $r(x, y)$, and adding all atoms $r(x, y)$ from q_r such that $x, y \in \mathsf{core}(q_r)$. The answer variables in q are those of q_0. We aim to prove the following.

Proposition 2. *Let $Q = (\mathcal{T}, \Sigma, q_0)$ be an OMQ from $(\mathcal{EL}, rCQ)$. If $\mathsf{bc}^+_{AQ}(Q_{q_r}, \mathcal{T}^{\mathsf{min}}_{q_r})$ is finite for all fork rewritings q_r of q_0, then $\bigvee_{q_r} \widehat{q}_{q_r}$ is a UCQ-rewriting of Q, where $\widehat{q}_{q_r}$ is the UCQ for Q that corresponds to $\mathsf{bc}^+_{AQ}(Q_{q_r}, \mathcal{T}^{\mathsf{min}}_{q_r})$. Otherwise, Q is not FO-rewritable.*

There are two exponential blowups in the presented approach. First, the number of fork rewritings of q_0 might be exponential in the size of q_0. We expect this not to be a problem in practice since the number of fork rewritings of realistic queries should be fairly small. And second, the number of splittings can be exponential and thus the same is true for the size of each $\mathcal{T}^{\mathsf{min}}_{q_r}$. We expect that also this blowup will be moderate in practice. Moreover, in an optimized implementation one would not represent $\mathcal{T}^{\mathsf{min}}_{q_r}$ as a TBox, but rather check the existence of fork rewritings and splittings that give rise to concept inclusions in $\mathcal{T}^{\mathsf{min}}_{q_r}$ in a more direct way. This involves checking whether concepts of the form $\exists r_i^{\nu(\mu(i))}.C_{q'|_{S_i}}$ are derived, and the fact that there are only polynomially many different such concepts should thus be very relevant regarding performance.

To prove Proposition 2, we introduce a backwards chaining algorithm for computing UCQ-rewritings of OMQs from $(\mathcal{EL}, \mathrm{rCQ})$ that we refer to as $\mathsf{bc}_{\mathrm{rCQ}}$. In a sense, $\mathsf{bc}_{\mathrm{rCQ}}$ is the natural generalization of $\mathsf{bc}_{\mathrm{AQ}}$ to rCQs. We first need to generalize some relevant notions underlying $\mathsf{bc}_{\mathrm{AQ}}$.

Let q be a CQ, $q' \subseteq q$, and $r(x, y) \in q$. Then q' is a *tree subquery in q with link $r(x, y)$* if q' is tree-shaped and the restriction of q to the variables reachable from y in the directed graph G_q, $\mathsf{var}(q') \cap \mathsf{avar}(q) = \emptyset$, and $s(u, z) \in q$ with $u \notin \mathsf{var}(q')$ and $z \in \mathsf{var}(q')$ implies $s(u, z) = r(x, y)$. Note that, taken together, $r(x, y)$ and q' can be viewed as an $\mathcal{EL}$-concept $\exists r.q'$. Let q and q' be CQs, $C \sqsubseteq D$ a concept inclusion, and $x \in \mathsf{var}(q)$. Then q' is *obtained from q by applying $C \sqsubseteq D$ at x* if q' can be obtained from q by

- removing $A(x)$ for all concept names A with $\models D \sqsubseteq A$;
- for each tree subquery q' of q with link $r(x,y)$ such that $\models D \sqsubseteq \exists r.q'$, removing $r(x,y)$ and q';
- adding $A(x)$ for all concept names A that occur in C as a top-level conjunct;
- adding $\exists r.E$ as a CQ with root x, for each $\exists r.E$ that is a top-level conjunct of C.

Let q, q' be CQs. We write $q' \prec q$ if q' can be obtained from q by selecting a tree subquery q'' in q with link $r(x,y)$ and removing both $r(x,y)$ and q''. We use $\prec^*$ to denote the reflexive and transitive closure of $\prec$ and say that q' *is $\prec$-minimal with* $q' \subseteq_{\mathcal{T}} q_0$ if $q' \subseteq_{\mathcal{T}} q_0$ and there is no $p \prec q'$ with $\mathcal{T} \models p \sqsubseteq A_0$.

Started on OMQ $Q = (\mathcal{T}, \Sigma, q_0)$, algorithm $\mathsf{bc}_{\mathrm{rCQ}}$ starts with a set R that contains for each fork rewriting q_r of q_0 a CQ $p \prec^* q_r$ that is $\prec$-minimal with $p \subseteq_{\mathcal{T}} q_0$ and then exhaustively performs the same steps as $\mathsf{bc}_{\mathrm{AQ}}$:

1. find $q \in R$, $x \in \mathsf{var}(q)$, $\alpha \in \mathcal{T}$, and q' such that q' is obtained from q by applying α at x;
2. find a $q'' \prec^* q'$ that is $\prec$-minimal with $q'' \subseteq_{\mathcal{T}} q_0$, and add q'' to R.

We use $\mathsf{bc}_{\mathrm{rCQ}}(Q)$ to denote the potentially infinitary UCQ $\bigvee R|_{\Sigma}$, R obtained in the limit.

The following establishes the central properties of the $\mathsf{bc}_{\mathrm{rCQ}}$ algorithm.

Lemma 5. *Let $Q = (\mathcal{T}, \Sigma, q_0)$ be an OMQ from $(\mathcal{EL}, rCQ)$. If $\mathsf{bc}_{rCQ}(Q)$ is finite, then it is a UCQ-rewriting of Q. Otherwise, Q is not FO-rewritable.*

We use Lemmas 4 and 5 and the construction of the queries Q_{q_r} and TBoxes $\mathcal{T}^{\mathsf{min}}_{q_r}$ to prove Proposition 2. Essentially, one shows that the run of $\mathsf{bc}_{\mathsf{rCQ}}(Q)$ is isomorphic to the union of the runs $\mathsf{bc}^{+}_{\mathsf{AQ}}(Q_{q_r}, \mathcal{T}^{\mathsf{min}}_{q_r})$. Note that Lemma 3 is a consequence of Lemma 5 and the fact that, when $Q = (\mathcal{T}, \Sigma, q_0)$ is from $(\mathcal{EL}, \mathrm{tqCQ})$, then $\mathsf{bc}_{\mathrm{rCQ}}(Q)$ contains only derivatives of q_0 (since the only fork rewriting of a tqCQ is the query itself).

5 Experiments

We have extended the *Grind* system [10] to support OMQs from $(\mathcal{EL}, \mathrm{tqCQ})$ and $(\mathcal{EL}, \mathrm{rCQ})$ instead of only from $(\mathcal{EL}, \mathrm{AQ})$, and conducted experiments with real-world TBoxes and hand-crafted conjunctive queries. The system is released under GPL, and can be downloaded from http://www.cs.uni-bremen.de/~hansen/grind, together with the TBoxes and queries. The system outputs rewritings in the form of non-recursive Datalog queries. It implements the following optimization: given $Q = (\mathcal{T}, \Sigma, q_0)$, first compute all fork rewritings of q_0, rewrite away all variables outside the core (in the same way in which tree parts of the query are removed in Section 3) to obtain a new OMQ $(\mathcal{T}', \Sigma, q_0')$, and then test for each atom $A(x) \in q_0'$ whether $(\mathcal{T}', \Sigma, A(x))$ is FO-rewritable. It can be shown that, if this is the case, then Q is FO-rewritable, and it is also possible to transfer the actual rewritings.

TBox	CI	CN	RN	Min CQ	Avg CQ	Max CQ	Avg AQ	Aborts
ENVO	1942	1558	7	0.2s	1.5s	7s	1s	0
FBbi	567	517	1	0.05s	0.5s	3s	0.3s	0
MOHSE	3665	2203	71	2s	10s	40s	6s	0
not-galen	4636	2748	159	6s	9s	28s	25s	2
SO	3160	2095	12	1s	19s	2m23s	4s	1

Table 1. TBox information and results of experiments

Experiments were carried out on a Linux (3.2.0) machine with a 3.5 GHz quad-core processor and 8 GB of RAM. For the experiments, we use (the $\mathcal{EL}$ part of) the ontologies ENVO, FBbi, SO, MOHSE, and not-galen. They are listed in Table 1, along with information about the number of concept inclusions (CI), concept names (CN), and role names (RN) they contain. For each TBox, we hand-crafted 10 conjunctive queries (three tqCQs and seven rCQs), varying in size from 2 to 5 variables and showing several different topologies.

The runtimes are reported in Table 1. Only three queries did not terminate in 30 minutes or exhausted the memory. For the successful ones, we list fastest (Min CQ), slowest (Max CQ), and average runtime (Avg CQ). For comparison, the Avg AQ column lists the time needed to compute FO-rewritings for all queries $(\mathcal{T}, \Sigma, A(x))$ with $A(x)$ an atom in q_0. This check is of course incomplete for FO-rewritability of Q, but can be viewed as a lower bound.

In summary, we believe that the outcome of our experiments is promising. While runtimes are higher than in the AQ case, they are still rather small given that we are dealing with an intricate static analysis task and that many parts of our system have not been seriously optimized. The queries with long runtimes or timeouts contain AQs that are not FO-rewritable, which forces the decomposed algorithm implemented in Grind to enter a more expensive processing phase.

6 Conclusion

We remark that our approach can also be used to compute FO-rewritings of OMQs from $(\mathcal{EL}, \text{CQ})$ even if the CQs are not rooted, as long as they are not Boolean (that is, as long as they contain at least one answer variable). This follows from (a minor variation of) an observation from [4]: FO-rewritability of non-Boolean OMQs from $(\mathcal{EL}, \text{CQ})$ can be reduced to a combination of containment in $(\mathcal{EL}, \text{CQ})$ and FO-rewritability in $(\mathcal{EL}, \text{rCQ})$. It would be interesting to extend our approach to UCQs, to the extension of $\mathcal{EL}$ with role hierarchies and domain and range restrictions, or even to $\mathcal{ELI}$.

Acknowledgements. We acknowledge support by ERC grant 647289 'CODA'.

References

1. Artale, A., Calvanese, D., Kontchakov, R., Zakharyaschev, M.: The DL-Lite family and relations. J. Artif. Intell. Res. 36, pp. 1–69 (2009)
2. Barceló, P., Berger, G., Pieris, A.: Containment for rule-based ontology-mediated queries. CoRR abs/1703.07994 (2017)
3. Bienvenu, M., ten Cate, B., Lutz, C., Wolter, F.: Ontology-based data access: A study through disjunctive datalog, CSP, and MMSNP. J. ACM Trans. Database Syst. 39(4), pp. 33:1–33:44 (2014)
4. Bienvenu, M., Hansen, P., Lutz, C., Wolter, F.: First order-rewritability and containment of conjunctive queries in Horn description logics. In: Proc. of IJCAI, pp. 965–971 (2016)
5. Bienvenu, M., Lutz, C., Wolter, F.: First order-rewritability of atomic queries in Horn description logics. In: Proc. of IJCAI, pp. 754–760 (2013)
6. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The DL-Lite family. J. Autom. Reasoning 39(3), pp. 385–429 (2007)
7. Deutsch, A., Popa, L., Tannen, V.: Physical data independence, constraints, and optimization with universal plans. In: Proc. of VLDB, pp. 459–470 (1999)
8. Feier, C., Lutz, C., Kuusisto, A.: Rewritability in monadic disjunctive datalog, MMSNP, and expressive description logics. In: Proc. of ICDT, pp. 1:1–1:17 (2017)
9. Hansen, P., Lutz, C.: Computing FO-rewritings in $\mathcal{EL}$ in practice: from atomic to conjunctive queries. In: Proc. of ISWC (2017)
10. Hansen, P., Lutz, C., Seylan, I., Wolter, F.: Efficient query rewriting in the description logic EL and beyond. In: Proc. of IJCAI. pp. 3034–3040 (2015)
11. König, M., Leclère, M., Mugnier, M., Thomazo, M.: Sound, complete and minimal UCQ-rewriting for existential rules. Semantic Web 6(5), pp. 451–475 (2015)
12. Lutz, C.: The complexity of conjunctive query answering in expressive description logics. In: Proc. of IJCAR, pp. 179–193 (2008)
13. Lutz, C.: Two upper bounds for conjunctive query answering in SHIQ. In: Proc. of DL (2008)
14. Lutz, C., Wolter, F.: Non-uniform data complexity of query answering in description logics. In: Proc. of KR (2012)