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. A prominent approach to implementing ontology-mediated queries (OMQs) is to rewrite into a first-order query, which is then executed using a conventional SQL database system. We consider the case where the ontology is formulated in the description logic $\mathcal{EL}$ and the actual query is a conjunctive query and show that rewritings of such OMQs can be efficiently computed in practice, in a sound and complete way. Our approach combines a reduction with a decomposed backwards chaining algorithm for OMQs that are based on the simpler atomic queries, also illuminating the relationship between first-order rewritings of OMQs based on conjunctive and on atomic queries. Experiments with real-world ontologies show promising results.

1 Introduction

One of the most important 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 [3, 4, 5, 6, 7, 10, 12, 15]. In fact, the DL-Lite family of description logics (DLs) was invented specifically with the aim to guarantee that FO-rewritings of OMQs (whose ontology is formulated in DL-Lite) always exist [1, 7], 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, one may hope that FO-rewritings of practically relevant OMQs exist in the majority of cases. This hope was confirmed in an experimental evaluation carried out in the context of the $\mathcal{EL}$ family of description logics where less than 1% of the considered queries was found not to be FO-rewritable [12]; 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 $\mathcal{EL}$, which can be viewed as a logical core of the OWL EL profile of the OWL 2 ontology language [20]. We use $(\mathcal{L}, \mathcal{Q})$ to denote the OMQ language that consists of all OMQs where the ontology is formulated in the description logic $\mathcal{L}$ and the actual query is formulated in the query language $\mathcal{Q}$. Important choices for $\mathcal{Q}$ include *atomic queries (AQs)* and the much more expressive *conjunctive queries (CQs)*. It has been shown in [6] that for OMQs from $(\mathcal{EL}, \text{AQ})$, it is ExpTime-complete to decide FO-rewritability. Combining the techniques from [6] and the backwards chaining approach to query rewriting brought forward e.g. in [8, 15], a practical algorithm for computing FO-rewritings of OMQs from $(\mathcal{EL}, \text{AQ})$ was then developed in [12]. This algorithm is based on a *decomposed version* of backwards chaining that implements a form of structure sharing. It was implemented in the *Grind* system and shown to perform very well in practice [12]. 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.

The aim of this paper is to devise a way to efficiently compute FO-rewritings of OMQs from $(\mathcal{EL}, \text{CQ})$, and thus the challenge is to deal with conjunctive queries instead of only with atomic ones. Note that, as shown in [5], FO-rewritability in $(\mathcal{EL}, \text{CQ})$ is still ExpTime-complete. Our approach is to combine a reduction with the decomposed algorithm from [12], also illuminating the relationship between first-order rewritings of OMQs based on CQs and on AQs. It is worthwhile to point out that naive reductions of FO-rewritability in $(\mathcal{EL}, \text{CQ})$ to FO-rewritability in $(\mathcal{EL}, \text{AQ})$ 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. As a simple example, consider the OMQ that consists of the ontology and query

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

and which is FO-rewritable into $\exists y\, s(x, y)$, but the only AQ $A(x)$ that occurs in q is not FO-rewritable in the presence of $\mathcal{O}$.[1] In fact, it is not clear how to attain a reduction of FO-rewritability in $(\mathcal{EL}, \text{CQ})$ to FO-rewritability in $(\mathcal{EL}, \text{AQ})$, and even less so a polynomial time one. This leads us to considering mildly restricted forms of CQs and admitting reductions that make certain assumptions on the algorithm used to compute FO-rewritings in $(\mathcal{EL}, \text{AQ})$—all of them are 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 form a collection of directed trees. In this case, we indeed achieve a polynomial time reduction to FO-rewritability in $(\mathcal{EL}, \text{AQ})$. 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 (union of conjunctive queries) 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 an undirected sense, in the query

[1] OMQs also allow to fix the signature (set of concept and role names) that can occur in the ABox. In this example, we do not assume any restriction on the ABox signature.

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 $(\mathcal{EL}, \text{AQ})$ 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 $(\mathcal{EL}, \text{AQ})$, similar to those constructed 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 ontology.

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 very reasonable given that we are dealing with a complex static analysis problem.

Proofs are deferred to the appendix, which is made available at http://www.cs.uni-bremen.de/tdki/research/papers.html.

Related Work. We directly build on our prior work in [12] as discussed above, and to a lesser degree also on [5,6]. The latter line of work has recently been picked up in the context of existential rules [3]. The distinguishing features of our work are that (1) our algorithms are sound, complete, and terminating, that is, they find an FO-rewriting if there is one and report failure otherwise, and (2) we rely on the decomposed calculus from [12] that implements structure sharing for constructing small rewritings and achieving practical feasibility. We are not aware of other work that combines features (1) and (2) and is applicable to OMQs based on $\mathcal{EL}$. In the context of the description logic DL-Lite, though, the construction of small rewritings has received a lot of attention, see e.g. [11, 13, 22, 23]. Producing small rewritings of OMQs whose ontology is a set of existential rules has been studied in [14], but there are no termination guarantees. Constructing small *Datalog*-rewritings of OMQs based on $\mathcal{EL}$, which are guaranteed to always exist, was studied e.g. in [9, 21, 25, 26]. A different approach to answering $\mathcal{EL}$-based OMQs using SQL databases is the combined approach where the consequences of the ontology are materialized in the data [18, 24].

2 Preliminaries

Let $\mathsf{N_C}$, $\mathsf{N_R}$, and $\mathsf{N_I}$ be countably infinite sets of *concept names*, *role names*, and *individual names*. An *$\mathcal{EL}$-concept* is formed according to the syntax rule

$$C, D ::= \top \mid A \mid C \sqcap D \mid \exists r.C$$

where A ranges over $\mathsf{N_C}$ and r over $\mathsf{N_R}$. An *$\mathcal{EL}$-TBox $\mathcal{T}$* is a finite set of *concept inclusions* $C \sqsubseteq D$, with C and D $\mathcal{EL}$-concepts. Throught the paper, we use $\mathcal{EL}$-TBoxes as ontologies. An *ABox* is a finite set of *concept assertions* $A(a)$ and *role assertions* $r(a, b)$ where a and b range over $\mathsf{N_I}$. We use $\mathsf{Ind}(\mathcal{A})$ to denote the set of individual names in the ABox $\mathcal{A}$. A *signature* is a set of concept and role names. When an ABox uses only symbols from a signature Σ, then we call it a *Σ-ABox*. To emphasize that a signature Σ is used to constrain the symbols admitted in ABoxes, we sometimes call Σ an *ABox signature*.

The semantics of concepts, TBoxes, and ABoxes is defined in the usual way, see [2]. We write $\mathcal{T} \models C \sqsubseteq D$ if the concept inclusion $C \sqsubseteq D$ is satisfied in every model of $\mathcal{T}$; when $\mathcal{T}$ is empty, we write $\models C \sqsubseteq D$. As usual in ontology-mediated querying, we make the *standard names assumption*, that is, an interpretation $\mathcal{I}$ satisfies a concept assertion $A(a)$ if $a \in A^{\mathcal{I}}$ and a role assertion $r(a, b)$ if $(a, b) \in r^{\mathcal{I}}$.

A *conjunctive query (CQ)* takes the form $q(\mathbf{x}) = \exists \mathbf{y}\, \varphi(\mathbf{x}, \mathbf{y})$ with $\mathbf{x}, \mathbf{y}$ tuples of variables and φ a conjunction of atoms of the form $A(x)$ and $r(x, y)$ that uses only variables from $\mathsf{var}(q) = \mathbf{x} \cup \mathbf{y}$. The variables $\mathbf{x}$ are the *answer variables* of q, denoted $\mathsf{avar}(q)$, and the *arity* of q is the length of $\mathbf{x}$. 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. An *atomic query (AQ)* is a conjunctive query of the form $A(x)$. A *union of conjunctive queries (UCQ)* is a disjunction of CQs that share the same answer variables.

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. We use $(\mathcal{EL}, \mathrm{AQ})$ to denote the set of OMQs where $\mathcal{T}$ is an $\mathcal{EL}$-TBox and q is an AQ, and similarly for $(\mathcal{EL}, \mathrm{CQ})$ and so on. We do generally not allow equality in CQs that are part of an OMQ. Let $Q = (\mathcal{T}, \Sigma, q)$ be an OMQ, $\mathcal{A}$ a Σ-ABox and $\mathbf{a} \subseteq \mathsf{Ind}(\mathcal{A})$. We write $\mathcal{A} \models Q(\mathbf{a})$ if $\mathcal{I} \models q(\mathbf{a})$ for all models $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{A}$. In this case, $\mathbf{a}$ is a *certain answer* to Q on $\mathcal{A}$.

Example 1. Consider an example from the medical domain. The following ABox holds data about patients and diagnoses:

$$\mathcal{A} = \{\mathsf{Person}(a), \mathsf{hasDisease}(a, oca_1), \mathsf{Albinism}(oca_1)\}$$

A TBox $\mathcal{T}_1$ is used to make domain knowledge available:

$$\begin{aligned} \mathcal{T}_1 = \{&\mathsf{Albinism} \sqsubseteq \mathsf{HereditaryDisease}, \\ \mathsf{Person} \sqcap \exists \mathsf{hasDisease}.\mathsf{HereditaryDisease} &\sqsubseteq \mathsf{GeneticRiskPatient}\} \end{aligned}$$

Let Q_1 be the OMQ $(\mathcal{T}_1, \Sigma_{\mathsf{full}}, q_1(x))$, where $q_1(x) = \mathsf{GeneticRiskPatient}(x)$, and Σ_{full} contains all concept and role names. It can be verified that $\mathcal{A} \models Q_1(a)$. $\dashv$

We do not distinguish between a CQ and the set of atoms in it and associate with each CQ q a directed graph $G_q := (\mathsf{var}(q), \{(x, y) \mid r(x, y) \in q\})$ (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. Every $\mathcal{EL}$-concept can be viewed as a tree-shaped CQ and vice versa; for example, the $\mathcal{EL}$-concept $A \sqcap \exists r.(B \sqcap \exists s.A)$ corresponds to the CQ $q(x) = \exists y, z\, A(x) \wedge r(x,y) \wedge B(y) \wedge s(y,z) \wedge A(z)$. We will not always distinguish between the two representations and even mix them. We might thus write $\exists r.q$ to denote an $\mathcal{EL}$-concept when q is a tree-shaped CQ; if $q(x)$ is as in the example just given, then $\exists r.q$ is the $\mathcal{EL}$-concept $\exists r.(A \sqcap \exists r.(B \sqcap \exists s.A))$. 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. For example, $q(x_1, x_2) = \exists y_1, y_2\, r(x_1, x_2) \wedge r(x_2, x_1) \wedge r(x_1, y_1) \wedge s(x_2, y_2)$ is a tqCQ and the tCQs in q are $\exists y_1\, r(x_1, y_1)$ and $\exists y_2\, s(x_2, y_2)$; by adding to q the atom $r(y_1, y_2)$, we obtain an rCQ that is not a tqCQ.

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 $(\mathcal{EL}, \mathrm{CQ})$ [4, 6]; 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 $q \vee (B(x) \wedge x = y)$, but not into an UCQ that does not use equality.

Example 2. We extend the TBox $\mathcal{T}_1$ from Example 1 to additionally describe the hereditary nature of genetic defects:

$$\mathcal{T}_2 := \mathcal{T}_1 \cup \{\mathsf{Person} \sqcap \exists \mathsf{hasParent}.\mathsf{GeneticRiskPatient} \sqsubseteq \mathsf{GeneticRiskPatient}\}.$$

The OMQ $Q_1' = (\mathcal{T}_2, \Sigma_{\mathsf{full}}, q_1(x))$ with $q_1(x)$ as in Example 1, is not FO-rewritable, intuitively because it expresses unbounded reachability along the $\mathsf{hasParent}$ role. In contrast, consider the OMQ $Q_2 = (\mathcal{T}_2, \Sigma_{\mathsf{full}}, q_2(x))$ where $q_2(x) = \exists y\ \mathsf{GeneticRiskPatient}(x) \wedge \mathsf{hasDisease}(x,y) \wedge \mathsf{Albinism}(y)$. Even though q_2 is an extension of q_1 with additional atoms, Q_2 is FO-rewritable, with $\varphi(x) = q_2(x) \vee \big(\exists y\ \mathsf{Person}(x) \wedge \mathsf{hasDisease}(x,y) \wedge \mathsf{Albinism}(y)\big)$ a concrete rewriting. ⊣

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 $(\mathcal{EL}, \mathrm{rCQ})$ 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 technical notions that are central to the constructions in Section 4. Both notions have been used before in the context of ontology-mediated querying, see for example [16, 17]. They are illustrated in Example 3 below.

Definition 1 (Fork rewriting). *Let* q_0 *be a CQ.* Obtaining a CQ q from q_0 by fork elimination *means to select 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} in q with x_i, where $i \in \{0,1\}$ is chosen such that x_i is an answer variable 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} is an answer variable. 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, that is, $q|_V$ is the set of atoms in q that use only variables from 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, $\mu : \{1, \dots, \ell\} \rightarrow R$ assigns to each set S_i a variable from R, $\nu : R \rightarrow \mathsf{Ind}(\mathcal{A})$ assigns to each variable from R and individual name from $\mathcal{A}$, 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 [17]. 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 $\langle R, S_1, \dots, S_\ell, r_1, \dots, r_\ell, \mu, \nu \rangle$ 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 $1 \leq i \leq \ell$.*

Example 3. To illustrate the described notions, consider the following CQ.

$$\begin{aligned} q_3(x) = \exists y_1, y_2, z\ & \mathsf{Person}(x) \wedge \\ & \mathsf{hasDisease}(x, y_1) \wedge \mathsf{MelaminDeficiency}(y_1) \wedge \mathsf{causedBy}(y_1, z) \wedge \\ & \mathsf{hasDisease}(x, y_2) \wedge \mathsf{ImpairedVision}(y_2) \wedge \mathsf{causedBy}(y_2, z) \wedge \\ & \mathsf{GeneDefect}(z) \end{aligned}$$

It asks for persons suffering from two conditions connected with the same gene defect. Let the ABox $\mathcal{A}$ consist only of the assertion $\mathsf{OCA1aPatient}(a)$. We extend the TBox $\mathcal{T}_2$ from Example 2, as follows:

$$\begin{aligned} \mathcal{T}_3 := \mathcal{T}_2 \cup \{\ & \mathsf{OCA1aPatient} \sqsubseteq \mathsf{Person} \sqcap \mathsf{hasDisease.OCA1aAlbinism} \\ & \mathsf{OCA1aAlbinism} \sqsubseteq \mathsf{ImpairedVision} \sqcap \mathsf{MelaninDeficiency} \\ & \mathsf{OCA1aAlbinism} \sqsubseteq \exists \mathsf{causedBy.GeneDefect}\ \} \end{aligned}$$

Let $Q = (\mathcal{T}_3, \Sigma_{\mathsf{full}}, q_3(x))$. It can be verified that $\mathcal{A} \models Q(a)$. By Lemma 1, this is witnessed by a fork rewriting and a splitting Π. The fork rewriting is

$$\begin{aligned} q_3'(x) = \exists y_1, z\ & \mathsf{Person}(x) \wedge \\ & \mathsf{hasDisease}(x, y_1) \wedge \mathsf{MelaminDeficiency}(y_1) \wedge \mathsf{ImpairedVision}(y_1) \wedge \\ & \mathsf{causedBy}(y_1, z) \wedge \mathsf{GeneDefect}(z) \end{aligned}$$

The splitting $\Pi = \langle R, S_1, r_1, \mu, \nu \rangle$ of q_3' wrt. $\mathcal{A}$ and $\mathcal{T}_3$ is defined by setting

$$R = \{x\},\ S_1 = \{y_1, z\},\ r_1 = \mathsf{hasDisease},\ \mu(1) = x,\ \nu = (x \mapsto a)$$

It can be verified that the conditions given in Lemma 1 are satisfied. ⊣

3 Tree-quantified CQs

We reduce 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 OMQs 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.

Let Q be an OMQ from $(\mathcal{EL}, \text{tqCQ})$. 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 set 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

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

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

Example 4. Consider the OMQ $Q = (\mathcal{T}_1, \Sigma_{\mathsf{full}}, q(x,y))$ with $\mathcal{T}_1$ as in Example 1 and let $q(x,y)$ the following tqCQ:[2]

$$q(x,y) = \exists z \ \mathsf{GeneticRiskPatient}(x) \wedge \mathsf{hasDisease}(x,y) \wedge \mathsf{Disease}(y) \wedge \mathsf{hasDisease}(x,z) \wedge \mathsf{Albinism}(z)$$

We first remove quantified variables: all atoms that contain the variable z are replaced by $A_{\exists \mathsf{hasDisease.Albinism}}(y)$, and the TBox is extended with the inclusion $\exists \mathsf{hasDisease.Albinism} \sqsubseteq A_{\exists \mathsf{hasDisease.Albinism}}$. We then construct $\mathcal{T}'_1$, which we give here only partially. The final concept inclusion in $\mathcal{T}_1$ is

$$\mathsf{GeneticRiskPatient}^x \sqcap \mathsf{Disease}^y \sqcap A^x_{\exists \mathsf{hasDisease.Albinism}} \sqsubseteq N,$$

representing the updated query without role atoms; for example, the concept name $\mathsf{Disease}^y$ stands for the atom $\mathsf{Disease}(y)$. Among others, $\mathcal{T}'_1$ contains the further concept inclusions

$$\exists \mathsf{hasDisease}^x.\mathsf{HereditaryDisease} \sqsubseteq A^x_{\exists \mathsf{hasDisease.HereditaryDisease}}$$
$$\mathsf{HereditaryDisease}^y \sqsubseteq A^x_{\exists \mathsf{hasDisease.HereditaryDisease}}$$

where, intuitively, the lower concept inclusion captures that case that the truth of the concept $\exists \mathsf{hasDisease.HereditaryDisease}$ is witnessed at y (the role atom $\mathsf{hasDisease}(x,y)$ from q is only implicit here) while the upper concept inclusion deals with other witnesses. ⊣

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 [6,12]. 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.

[2] We only use here that $\mathcal{T}_1$ contains the concept $\exists \mathsf{hasDisease.HereditaryDisease}$ on the left-hand side of a concept inclusion.

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 [12] 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 consider OMQs based on rCQs, a strict generalization of tqCQs. In this case, we are not going to achieve a 'black box' reduction, but rely on a concrete algorithm for solving FO-rewritability in $(\mathcal{EL}, \text{AQ})$. This algorithm is a straightforward and not necessarily terminating backwards chaining algorithm or a (potentially terminating) refinement thereof, as implemented in the Grind system. 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 the TBox used in the OMQ. 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 $\mathsf{bc}_{\mathrm{AQ}}$. It maintains a set M of $\mathcal{EL}$-concepts that represent tCQs. Let $Q = (\mathcal{T}, \Sigma, A_0)$ be from $(\mathcal{EL}, \mathrm{AQ})$. 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 $D' \prec^* D$ 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 $\mathsf{bc}_{\mathrm{AQ}}(Q)$ 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. Note that, in Point 2, it is possible to find the desired D' in polynomial time since the subsumption '$\mathcal{T} \models D' \sqsubseteq A_0$' can be decided in polynomial time. The following is standard to prove, see [12, 15] 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.*

Example 5. Consider the TBox

$$\mathcal{T} = \{\mathsf{Person} \sqcap \exists \mathsf{hasParent}.\mathsf{GeneticRiskPatient} \sqsubseteq \mathsf{GeneticRiskPatient}\}$$

and let $Q = (\mathcal{T}, \Sigma, \mathsf{GeneticRiskPatient}(x))$ with $\Sigma = \{\mathsf{Person}, \mathsf{GeneticRiskPatient}\}$. Note that the role name $\mathsf{hasParent}$ does not occur in Σ. Even though the set M generated by $\mathsf{bc}_{\mathrm{AQ}}$ (in the limit of its non-terminating run) is infinite, $\mathsf{bc}_{\mathrm{AQ}}(Q) = \mathsf{GeneticRiskPatient}(x)$ is finite and a UCQ-rewriting of Q. ⊣

The algorithm for deciding FO-rewritability in $(\mathcal{EL}, \mathrm{AQ})$ presented in [12] and underlying the Grind system can be seen as a refinement of $\mathsf{bc}_{\mathrm{AQ}}$. 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 (a non-recursive Datalog program that is equivalent to) 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 [17]). 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 $\mathcal{T}^{\mathsf{min}}$.

Example 6. Consider query q_3 from Example 3 and TBox $\mathcal{T}_1$ from Example 1. Constructing $\mathcal{T}_{q_3}$ (thus considering q_3 as a fork rewriting of itself) would add concept inclusions like

$$\mathsf{Person}^x \sqcap \exists \mathsf{hasDisease}^x.\mathsf{HereditaryDisease} \sqsubseteq \mathsf{GeneticRiskPatient}^x$$

The final concept inclusion added is the following, listing concepts needed at x, y_1, y_2, and z that result in a match of q_3:

$$\mathsf{Person}^x \sqcap \mathsf{MelaminDeficiency}^{y_1} \sqcap \mathsf{ImpairedVision}^{y_2} \sqcap \mathsf{GeneDefect}^z \sqsubseteq N$$

When building the TBox $\mathcal{T}^{\mathsf{min}}_{q_3}$, it is necessary to look for matches of q_3 by a splitting Π of a fork rewriting of q_3 w.r.t. $\mathcal{A}_{q_3}$ and $\mathcal{T}_1$. We consider here the splitting $\Pi = \langle R, S_1, r_1, \mu, \nu \rangle$ of the fork rewriting q_3' of q_3 given in Example 3, defined by setting

$$R = \{x\},\ S_1 = \{y_1, z\},\ r_1 = \mathsf{hasDisease},\ \mu(1) = x,\ \nu = (x \mapsto x)$$

For Π, the following concept inclusion is added to $\mathcal{T}^{\mathsf{min}}_{q_3}$:

$$\mathsf{Person}^x \sqcap \exists \mathsf{hasDisease}^x.\big(\mathsf{MelaminDeficiency} \sqcap \mathsf{ImpairedVision} \sqcap \mathsf{causedBy}.\mathsf{GeneDefect}\big) \sqsubseteq N \qquad \dashv$$

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.

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.*

To prove Proposition 2, we introduce a backwards chaining algorithm $\mathsf{bc}_{\mathrm{rCQ}}$ 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 then show a correspondence between the run of $\mathsf{bc}_{\mathrm{rCQ}}$ on the input OMQ Q from $(\mathcal{EL}, \mathrm{rCQ})$ and the runs of $\mathsf{bc}^{+}_{\mathrm{AQ}}$ on the constructed inputs of the form $(Q_{q_r}, \mathcal{T}^{\mathsf{min}}_{q_r})$.

On the way, we also provide the missing proof for Lemma 3, which in fact is a consequence of the correctness of $\mathsf{bc}_{\mathrm{rCQ}}$ (states as Lemma 5 in the appendix) and the observation 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. The latter is due to the definition of the $\mathsf{bc}_{\mathrm{rCQ}}$ algorithm, which starts with a set of minimized fork rewritings of q_0, and the fact that the only fork rewriting of a tqCQ is the query itself.

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.

5 Experiments

We have extended the *Grind* system [12] 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 ontologies and hand-crafted conjunctive queries. The system can be downloaded from http://www.cs.uni-bremen.de/~hansen/grind, together with the ontologies and queries, and is released under GPL. It outputs rewritings in the form of non-recursive Datalog queries. We have implemented the following optimization: given $Q = (\mathcal{T}, \Sigma, q_0)$, first compute all fork rewritings of q_0, rewrite away all variables outside of 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. If this check fails, we go through the full construction described in the paper.

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. The first three ontologies are from the biology domain, and are available through Bioportal[3].

[3] https://bioportal.bioontology.org

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

$$
\begin{aligned}
q_1(x,y) &= \mathsf{Patient}(x) \wedge \mathsf{shows}(x,y) \wedge \mathsf{Endocarditis}(y)\\
q_2(w,x,y,z) &= \mathsf{Doctor}(w) \wedge \mathsf{hasPersonPerforming}(x,w) \wedge \mathsf{Surgery}(x) \wedge\\
&\quad \mathsf{actsOn}(x,y) \wedge \mathsf{Tissue}(y) \wedge \mathsf{actsOn}(x,z) \wedge\\
&\quad \mathsf{InternalOrgan}(z) \wedge \mathsf{hasAlphaConnection}(y,z)\\
q_7(x) &= \exists y,z\ \mathsf{Protein}(x) \wedge \mathsf{contains}(x,y) \wedge \mathsf{Tetracycline}(y) \wedge\\
&\quad \mathsf{InternalOrgan}(z) \wedge \mathsf{isActedOnSpecificallyBy}(z,y)\\
q_8(x) &= \exists v,w,y,z\ \mathsf{Sulphonamide}(v) \wedge \mathsf{serves}(v,w) \wedge \mathsf{TumorMarkerRole}(w) \wedge\\
&\quad \mathsf{NamedEnzyme}(x) \wedge \mathsf{serves}(x,w) \wedge \mathsf{actsOn}(x,z) \wedge \mathsf{Liver}(z) \wedge\\
&\quad \mathsf{TeichoicAcid}(y) \wedge \mathsf{actsOn}(y,z)\\
q_{10}(x) &= \exists y,z\ \mathsf{BodyStructure}(x) \wedge \mathsf{isBetaConnectionOf}(x,y) \wedge \mathsf{Brain}(y) \wedge\\
&\quad \mathsf{IntrinsicallyNormalBodyStructure}(z) \wedge \mathsf{isBetaConnectionOf}(z,y)
\end{aligned}
$$

Fig. 1. Examplary queries used for experiments with TBox not-galen.

MOHSE and not-galen are different versions of the GALEN ontology[4], which describes medical terms. Some statistics is given in Table 1, namely the number of concept inclusions (CI), concept names (CN), and role names (RN) in each ontology. For each ontology, we hand-crafted 10 conjunctive queries (three tqCQs and seven rCQs), varying in size from 2 to 5 variables and showing several different topologies (see Fig. 1 for a sample).

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. A detailed picture of individual runtimes is given in Figure 2.

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

[4] http://www.opengalen.org/

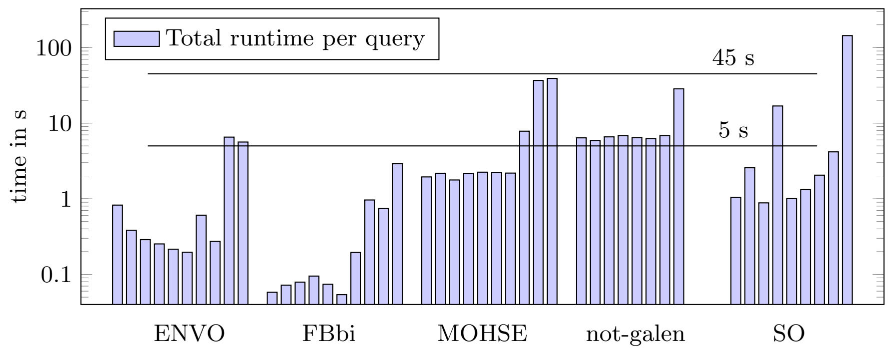

Fig. 2. Runtimes for individual OMQs, showing only non-aborting runs.

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) and an algorithm for query containment in $(\mathcal{EL}, \text{CQ})$ is also available. This follows from (a minor variation of) an observation from [5]: FO-rewritability of non-Boolean OMQs from $(\mathcal{EL}, \text{CQ})$ can be polynomially reduced to a combination of containment in $(\mathcal{EL}, \text{CQ})$ and FO-rewritability in $(\mathcal{EL}, \text{rCQ})$. As future work, 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 the extension $\mathcal{ELI}$ of $\mathcal{EL}$ with inverse roles.

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. Baader, F., Horrocks, I., Lutz, C., Sattler, U.: An introduction to description logics. Cambride University Press (2017)
3. Barceló, P., Berger, G., Pieris, A.: Containment for rule-based ontology-mediated queries (19 April 2017), available at https://arxiv.org/abs/1703.07994 [cs.DB]
4. 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)
5. 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)

6. Bienvenu, M., Lutz, C., Wolter, F.: First order-rewritability of atomic queries in Horn description logics. In: Proc. of IJCAI, pp. 754–760 (2013)
7. 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)
8. Deutsch, A., Popa, L., Tannen, V.: Physical data independence, constraints, and optimization with universal plans. In: Proc. of VLDB, pp. 459–470 (1999)
9. Eiter, T., Ortiz, M., Simkus, M., Tran, T., Xiao, G.: Query rewriting for Horn-SHIQ plus rules. In: Proc. of AAAI (2012)
10. Feier, C., Lutz, C., Kuusisto, A.: Rewritability in monadic disjunctive datalog, MMSNP, and expressive description logics. In: Proc. of ICDT (2017)
11. Gottlob, G., Kikot, S., Kontchakov, R., Podolskii, V.V., Schwentick, T., Zakharyaschev, M.: The price of query rewriting in ontology-based data access. J. Artif. Intell. 213, pp. 42–59 (2014)
12. 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)
13. Kikot, S., Kontchakov, R., Zakharyaschev, M.: Conjunctive query answering with OWL 2 QL. In: Proc. of KR (2012)
14. König, M., Leclère, M., Mugnier, M.: Query rewriting for existential rules with compiled preorder. In: Proc. of IJCAI, pp. 3106–3112 (2015)
15. 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)
16. Lutz, C.: The complexity of conjunctive query answering in expressive description logics. In: Proc. of IJCAR, pp. 179–193 (2008)
17. Lutz, C.: Two upper bounds for conjunctive query answering in SHIQ. In: Proc. of DL (2008)
18. Lutz, C., Toman, D., Wolter, F.: Conjunctive query answering in the description logic EL using a relational database system. In: Proc. of IJCAI, pp. 2070–2075 (2009)
19. Lutz, C., Wolter, F.: Non-uniform data complexity of query answering in description logics. In: Proc. of KR (2012)
20. Motik, B., Cuenca Grau, B., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C.: OWL 2 Web Ontology Language: Profiles. W3C recommendation (11 December 2012), available at http://www.w3.org/TR/owl2-profiles/
21. Pérez-Urbina, H., Motik, B., Horrocks, I.: Tractable query answering and rewriting under description logic constraints. J. Applied Logic 8(2), pp. 186–209 (2010)
22. Rodriguez-Muro, M., Calvanese, D.: High performance query answering over DL-Lite ontologies. In: Proc. of KR (2012)
23. Rosati, R., Almatelli, A.: Improving query answering over DL-Lite ontologies. In: Proc. of KR (2010)
24. Stefanoni, G., Motik, B.: Answering conjunctive queries over EL knowledge bases with transitive and reflexive roles. In: Proc. of AAAI, pp. 1611–1617 (2015)
25. Stefanoni, G., Motik, B., Horrocks, I.: Small datalog query rewritings for EL. In: Proc. of DL (2012)
26. Trivela, D., Stoilos, G., Chortaras, A., Stamou, G.B.: Optimising resolution-based rewriting algorithms for OWL ontologies. J. Web Sem. 33, pp. 30–49 (2015)

Appendix

A Proofs for Section 3

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.*

Proof. (sketch) Assume that $Q' = (\mathcal{T}', \Sigma', q_0')$ is FO-rewritable. Then there is a tUCQ-rewriting φ [6, 12]. For all tCQs q in φ, it holds that $q \subseteq_{\mathcal{T}'} q_0'$. We can assume w.l.o.g. that every CQ q in φ is $\subseteq$-minimal with this property. We argue that φ must be conformant. Let q be a CQ in φ. First assume that q contains a concept atom $A(x_0)$ (or role atom $r(x_0, y)$) where x_0 is the answer variable and A not of the form B^y (or r not of the form s^z). Let q^- be q without this atom. The fact that $q \subseteq_{\mathcal{T}'} q_0'$ is equivalent to a_{x_0} being an answer to query Q' on the ABox $\mathcal{A}_q$, which is q seen as an ABox with root a_{x_0}. Entailment of AQs under a TBox can be characterized by derivation trees, see e.g. [5], which are similar to Datalog proof trees. Here, it is a consequence of the syntactic shape of $\mathcal{T}'$ that such a proof tree for $N(a)$, with a some individual, will not contain facts $A(a)$ (or $(r(a, b))$ in the derivation, where A is not of the form B^x (or r not of the form s^y). It follows that $q^- \subseteq_{\mathcal{T}'} q_0'$, a contradiction to minimality of q. Second, assume that q contains a concept atom $A^x(y)$ (or role atom $r^x(y, z)$), with y not the answer variable. Again regarding a proof tree for $N(a)$, the syntactic shape of $\mathcal{T}'$ prevents atoms of the described shape to occur at any other individual than a. It follows that $q^- \subseteq_{\mathcal{T}'} q_0'$, again contradicting minimality of q. □

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.*

Proof. "⇒". Assume that Q is FO-rewritable. By Lemma 3, there is a UCQ-rewriting q of Q in which every CQ is a derivative of q_0. Let q' be the corresponding UCQ for Q'. We argue that q' is a rewriting of Q'. Thus let $\mathcal{A}$ be a Σ'-ABox and $a_0 \in \mathsf{Ind}(\mathcal{A})$. We have to show that $\mathcal{A} \models Q'(a_0)$ iff $\mathcal{A} \models q'(a_0)$. Let $\mathbf{x} = x_1 \cdots x_n$ be the answer variables in q_0 and let $\mathbf{a} = a_1 \cdots a_n$ be a tuple of individual names that do not occur in $\mathcal{A}$.

In the first step, we unravel $\mathcal{A}$ into an infinite tree-shaped Σ'-ABox $\mathcal{A}'$ such that

1. $\mathcal{A} \models Q'(a_0)$ iff $\mathcal{A}' \models Q'(a_0)$ and
2. $\mathcal{A} \models q'(a_0)$ iff $\mathcal{A}' \models q'(a_0)$.

A *path in* $\mathcal{A}$ is a sequence $b_1, r_1, b_2, r_2, \ldots, b_k$ such that $b_1 = a_0$, $b_2, \ldots, b_k \in \mathsf{Ind}(\mathcal{A})$, $r_1, \ldots, r_{k-1}$ are role names that occur in $\mathcal{A}$, and $r_i(b_i, b_{i+1}) \in \mathcal{A}$ for $1 \leq i < k$. $\mathcal{A}'$ consists of all assertions

- $A(p)$ whenever p is a path in $\mathcal{A}$ that ends with b and $A(b) \in \mathcal{A}$; and

– $r(p, p')$ whenever p is a path in $\mathcal{A}$ that ends with b, $p' = rb'$ is a path in $\mathcal{A}$, and $r(b, b') \in \mathcal{A}$.

It follows from standard results about OMQs from $(\mathcal{EL}, \text{AQ})$ that Condition 1 is satisfied, see for example [19]. It can be verified that Condition 2 is also satisfied since q' is a tUCQ.

Using compactness and monotonicity, it is easy to show that since q is a rewriting of Q on (finite) ABoxes, it is also a rewriting of Q on infinite ABoxes. It thus remains to show that

3. $\mathcal{A}' \models Q'(a_0)$ iff $\mathcal{B}' \models Q(\mathbf{a})$ and
4. $\mathcal{A}' \models q'(a_0)$ iff $\mathcal{B}' \models q(\mathbf{a})$.

where $\mathcal{B}'$ is the (infinite) Σ-ABox that corresponds to $\mathcal{A}'$, that is, $\mathcal{B}'$ is obtained from $\mathcal{A}'$ by replacing every assertion $A^{x_i}(a_0)$ with $A(a_i)$ and every assertion $r^{x_i}(a_0, b)$ with $r(a_i, b)$, adding $r(a_i, a_j)$ whenever $r(x_i, x_j) \in q_0$, and then removing all remaining assertions that contain a symbol from $\Sigma' \setminus \Sigma$ or the individual name a_0.

In fact, Condition 3 can be shown using the construction of Q' and by translating counter models, and Condition 4 can be shown using the construction of q'.

"$\Leftarrow$". Assume that Q' is FO-rewritable. Then there is a conformant tUCQ-rewriting q' of Q'. Let q be the corresponding UCQ for Q. We have to show that q is a rewriting of Q (this also establishes the "moreover" part of the lemma). Thus let $\mathcal{A}$ be a Σ-ABox and $\mathbf{a} \subseteq \mathsf{Ind}(\mathcal{A})$. We aim to show that $\mathcal{A} \models Q(\mathbf{a})$ iff $\mathcal{A} \models q(\mathbf{a})$. Let $\mathbf{x} = x_1 \cdots x_n$ be the answer variables in q_0 and $\mathbf{a} = a_1 \cdots a_n$. It suffices to consider ABoxes $\mathcal{A}$ such that

$(*)$ $r(x_i, x_j) \in q_0$ implies $r(a_i, a_j) \in \mathcal{A}$

since, otherwise, $\mathcal{A} \not\models Q(\mathbf{a})$ and $\mathcal{A} \not\models q(\mathbf{a})$.

In the first step, we unravel $\mathcal{A}$ into an infinite Σ-ABox $\mathcal{A}'$ of more regular shape and with $\mathbf{a} \subseteq \mathsf{Ind}(\mathcal{A}')$ such that

1. $\mathcal{A} \models Q(\mathbf{a})$ iff $\mathcal{A}' \models Q(\mathbf{a})$ and
2. $\mathcal{A} \models q(\mathbf{a})$ iff $\mathcal{A}' \models q(\mathbf{a})$.

$\mathcal{A}'$ is constructed as follows. Start with the minimal ABox $\mathcal{A}'$ that satisfies $(*)$. Then extend $\mathcal{A}'$ as follows. A *path in* $\mathcal{A}$ is a sequence $b_1, r_1, b_2, r_2, \ldots, b_k$ such that $b_1 \in \mathbf{a}$, $b_2, \ldots, b_k \in \mathsf{Ind}(\mathcal{A})$, $r_1, \ldots, r_{k-1}$ are role names that occur in $\mathcal{A}$, and $r_i(b_i, b_{i+1}) \in \mathcal{A}$ for $1 \leq i < k$. Include in $\mathcal{A}'$ all assertions

– $A(p)$ whenever p is a path in $\mathcal{A}$ that ends with b and $A(b) \in \mathcal{A}$;
– $r(p, p')$ whenever p is a path in $\mathcal{A}$ that ends with b, $p' = rb'$ is a path in $\mathcal{A}$, and $r(b, b') \in \mathcal{A}$.

This finishes the construction of $\mathcal{A}'$. It can be proved that Conditions 1 and 2 are satisfied, translating counter models to prove Condition 1 and exploiting the construction of q (which ensures that q is a union of tqCQs that contains only role atoms from q_0 among the answer variables) for Condition 2.

By compactness and monotonicity, q' is a rewriting of Q' also on infinite ABoxes. It thus remains to show that

3. $\mathcal{A}' \models Q(\mathbf{a})$ iff $\mathcal{B}' \models Q'(a_0)$ and
4. $\mathcal{A}' \models q(\mathbf{a})$ iff $\mathcal{B}' \models q'(a_0)$.

where $\mathcal{B}'$ is the (infinite) Σ'-ABox that corresponds to $\mathcal{A}'$, that is, $\mathcal{B}'$ is obtained from $\mathcal{A}'$ by replacing all assertions $A(a_i)$ with $A^{x_i}(a_0)$ and all assertions $r(a_i, b)$ with $r^{x_i}(a_0, b)$, and then removing all role assertions that involve only answer variables.

In fact, Condition 3 can be shown using the construction of Q' and by translating counter models, and Condition 4 can be shown using the construction of q. In both cases, one exploits that $(*)$ holds for $\mathcal{A}'$, which is a consequence of the fact that it holds for $\mathcal{A}$. □

B Proofs for Section 4

We introduce a backwards chaining algorithm for computing UCQ-rewritings of OMQs from $(\mathcal{EL}, \text{rCQ})$ that we refer to as bc_{rCQ}. In a sense, bc_{rCQ} is the natural generalization of bc_{AQ} to rCQs. We first need to generalize some relevant notions underlying bc_{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 $\mathcal{T} \models p \sqsubseteq q_0$.

Started on OMQ $Q = (\mathcal{T}, \Sigma, q_0)$, algorithm bc_{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 bc_{AQ}:

1. find $q \in R$, $x \in \mathsf{var}(q)$, a concept inclusion $E \sqsubseteq F \in \mathcal{T}$, and q' such that q' is obtained from q by applying $E \sqsubseteq F$ 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. It is proved by showing that there is a correspondence between the backwards chaining implemented in $\mathsf{bc}_{\mathrm{rCQ}}$ and the chase, a forward chaining procedure that can be applied to an ABox $\mathcal{A}$ and a TBox $\mathcal{T}$ to construct a universal model of $\mathcal{A}$ and $\mathcal{T}$, that is a model that gives exactly the certain answers on $\mathcal{A}$ to any OMQ from $(\mathcal{EL}, \mathrm{AQ})$ based on $\mathcal{T}$.

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.*

In preparation for the proof of Lemma 5, we remind the reader of the standard chase procedure. The chase is a forward chaining procedure that exhaustively applies the concept inclusions of a TBox to an ABox in a rule-like fashion. Its final result is a (potentially infinite) ABox in which all consequences of $\mathcal{T}$ are materialized. To describe the procedure in detail, it is helpful to regard $\mathcal{EL}$-concepts C as tree-shaped ABoxes $\mathcal{A}_C$. $\mathcal{A}_C$ can be obtained from the concept query corresponding to C by identifying its individual variables with individual names. Now let $\mathcal{T}$ be an $\mathcal{EL}$-TBox and $\mathcal{A}$ an ABox. A *chase step* consists in choosing a concept inclusion $C \sqsubseteq D \in \mathcal{T}$ and an individual $a \in \mathsf{Ind}(\mathcal{A})$ such that $\mathcal{A} \models C(a)$, and then extending $\mathcal{A}$ by taking a copy $\mathcal{A}_D$ of D viewed as an ABox with root a and such that all non-roots are fresh individuals, and then setting $\mathcal{A} := \mathcal{A} \cup \mathcal{A}_D$. The *result of chasing $\mathcal{A}$ with $\mathcal{T}$*, denoted with $\mathsf{ch}_{\mathcal{T}}(\mathcal{A})$, is the ABox obtained by exhaustively applying chase steps to $\mathcal{A}$ in a fair way. It is standard to show that the chase procudes a universal model, i.e. for all CQs q and tuples $\mathbf{a} = (a_1, \dots, a_n)$ over $\mathsf{Ind}(\mathcal{A})$, it holds that $\mathcal{A}, \mathcal{T} \models q(\mathbf{a})$ iff $\mathsf{ch}_{\mathcal{T}}(\mathcal{A}) \models q(\mathbf{a})$. We now prove Lemma 5.

Proof. For the first part, let $\mathsf{bc}_{\mathrm{rCQ}}(Q)$ be finite, and $\mathcal{A}$ be a Σ-ABox and $\mathbf{a} \subseteq \mathsf{Ind}(\mathcal{A})$. We have to show $\mathcal{A} \models \bigvee R|_{\Sigma}(\mathbf{a})$ iff $\mathcal{A} \models Q(\mathbf{a})$. For direction "$\Rightarrow$", assume that $\mathcal{A} \models \bigvee R|_{\Sigma}(\mathbf{a})$. Then there is a $q \in R|_{\Sigma}$ with $\mathcal{A} \models q(\mathbf{a})$. Consequently $\mathcal{A}, \mathcal{T} \models q(\mathbf{a})$. By construction of R, all its elements q satisfy $q \subseteq_{\mathcal{T}} q_0$, thus $\mathcal{A} \models Q(\mathbf{a})$, as required.

For direction "$\Leftarrow$", we examine the chase sequence that witnesses $\mathcal{A} \models Q(\mathbf{a})$. W.l.o.g. we can assume that $\mathcal{T}$ contains no conjunctions on the right-hand side of concept inclusions, i.e. $\mathcal{T}$ consists only of concept inclusions of the form $C \sqsubseteq A$ and $C \sqsubseteq \exists r.D$. If $\mathcal{A} \models Q(\mathbf{a})$, then $\mathsf{ch}_{\mathcal{T}}(\mathcal{A}) \models q_0(\mathbf{a})$ and consequently, there is a sequence of (not necessarily Σ-) ABoxes $\mathcal{A} = \mathcal{A}_0, \mathcal{A}_1, \dots, \mathcal{A}_k$ that *demonstrates* $\mathsf{ch}_{\mathcal{T}}(\mathcal{A}) \models q_0(\mathbf{a})$, that is, each $\mathcal{A}_{i+1}$ is obtained from $\mathcal{A}_i$ by a single chase step and $\mathcal{A}_k \models q_0(\mathbf{a})$.

It thus suffices to prove by induction on k that if $\mathcal{A} = \mathcal{A}_0, \dots, \mathcal{A}_k$ is a chase sequence that demonstrates $\mathsf{ch}_{\mathcal{T}}(\mathcal{A}) \models q_0(\mathbf{a})$, then $\mathcal{A} \models \bigvee R|_{\Sigma}(\mathbf{a})$. The induction start is trivial: For $k = 0$, $\mathcal{A}_k \models q_0(\mathbf{a})$ implies $\mathcal{A} \models q_0(\mathbf{a})$. Since q_0 is a fork rewriting of itself, and by definition of R_0, there is a query $p \prec^* q_0$ in R. Restrict the homomorphism witnessing $\mathcal{A} \models q_0(\mathbf{a})$ to the variables still present in p, and the result is a homomorphism from p to $\mathcal{A}$, mapping the answer variables to $\mathbf{a}$. Thus, we have $\mathcal{A} \models p(\mathbf{a})$, and $\mathcal{A} \models \bigvee R|_\Sigma(\mathbf{a})$. For the induction step, assume that $\mathcal{A} = \mathcal{A}_0, \ldots, \mathcal{A}_k$ is a chase sequence that demonstrates $\mathsf{ch}_\mathcal{T}(\mathcal{A}) \models q_0(\mathbf{a})$, with $k > 0$. Applying IH to the subsequence $\mathcal{A}_1, \ldots, \mathcal{A}_k$, we obtain that $\mathcal{A}_1 \models \bigvee R|_\Sigma(\mathbf{a})$. Thus there is a $q \in R|_\Sigma$ with $\mathcal{A}_1 \models q(\mathbf{a})$, witnessed by a homomorphism h from q to $\mathcal{A}_1$ with $h(\mathbf{x}) = \mathbf{a}$. If $\mathcal{A}_0 \models q(\mathbf{a})$, then we are done. Otherwise, look at the chase step that led from $\mathcal{A}_0$ to $\mathcal{A}_1$. Assume that $\mathcal{A}_1$ is obtained from $\mathcal{A}_0$ by choosing a concept inclusion $E \sqsubseteq F \in \mathcal{T}$ and $b \in \mathsf{Ind}(\mathcal{A}_0)$ with $\mathcal{A}_0 \models E(b)$, and adding a copy of the ABox $\mathcal{A}_F$ to $\mathcal{A}_0$ at b. There are two possibilities:

- $F = A$
  An atom $A(b)$ is added to $\mathcal{A}_0$. Further, let $z_1, \ldots, z_n$ be all variables of q such that $h(z_i) = b$ and $A(z_i) \in q$. There must be at least one such z_i, since otherwise h would not depend on any assertions added in the construction of $\mathcal{A}_1$ from $\mathcal{A}_0$ and thus witnessed $\mathcal{A}_0 \models q(\mathbf{a})$, a contradiction.
- $F = \exists r.G$
  An atom $r(b, d)$ with d fresh, and an ABox $\mathcal{A}_G$ of fresh individuals rooted in d, are added to $\mathcal{A}_0$. Let $\mathbf{y} = y_1, \ldots, y_m$ be all variables of q such that $h(y_i) = d$, and $\mathbf{z} = z_1, \ldots, z_n$ be all variables of q such that $h(z_i) = b$ and there is at least one r-successor of z_i in $\mathbf{y}$. As above, there must be at least one such z_i. Note that there are no answer variables among $\mathbf{y}$, as d is anonymous.

Ideally, in the second case we would have q conform to the following property:

($*$) For every $y_i \in \mathbf{y}$, it holds that y_i is the root of a tree subquery q'_i of q with link $r(z_j, y_i)$, and $\mathcal{A}_G \models q'_i$, where $z_j \in \mathbf{z}$.

Note that this is not guaranteed, as there might be forks $r(z_i, y), r(z_j, y)$ occurring in q at $\mathbf{z}$, or $r(y_i, x), r(y_j, x)$ with y_i, y_j from $\mathbf{y}$ or below. These variables could still be mapped to the tree-shaped part $\mathcal{A}_F(b)$ of $\mathcal{A}_1$ by h. Nonetheless, we can find a query $q' \in R$ such that q' fulfills ($*$): Assume there is a fork $r(z_i, y), r(z_j, y)$ with $y \in \mathbf{y}$ (forks below are handled in the same way), and q is a derivative of some $\prec$-minimized fork rewriting p of q_0. Then variables y, z_i, z_j are part of the core of p, as backward application of concept inclusions does not generate forks. Let p' be p with the fork $r(z_i, y), r(z_j, y)$ eliminated; we are guaranteed to have a query $p'' = \mathsf{min}(p') \in R$. Note first that in p'', the subtree rooted in z (the identification of z_i and z_j) might have been deleted by minimization. There has to be another $r(z_k, y_k)$, as otherwise, $\mathcal{A}_0 \models q(\mathbf{a})$, in which case we again would be done. From p'', we can obtain a derivative q' of p'' of the desired form by backwards application of concept inclusions in $\mathcal{T}$. This can be shown by induction on the length of the backwards chaining sequence that led from p to q: Either we can apply a concept inclusion α to both p and p'', or it is applied to the deleted tree in p, in which case we omit this application in derivatives of p'' when generating q'. In either case, the original fork will not be present in q'. Continue the proof using query q' for q.

Let the CQs $q^0, \ldots, q^n$ be such that $q = q^0$, and q^{i+1} can be obtained from q^i by doing the following if $z_i \in \text{var}(q^i)$ (otherwise, just set $q^{i+1} := q^i$):

1. remove $A(z_i)$ if $F = A$;
2. remove $r(z_i, y)$ and the subquery q' of q with link $r(z_i, y)$ if $\models F \sqsubseteq \exists r.q'$;
3. add $A(z_i)$ for all concept names A that are top-level conjuncts of E;
4. add $\exists r.H$ as a CQ with root z_i, for each $\exists r.H$ that is a top-level conjunct of E;
5. minimize the resulting $q^{i\prime}$, that is, choose $q^{i+1} \prec^* q^{i\prime}$ such that q^{i+1} is $\prec$-minimal with $q^{i+1} \subseteq_{\mathcal{T}} q_0$.

It is easy to prove by induction on i that $q^i \in R$ for all $i \leq n$. It thus remains to argue that $\mathcal{A}_0 \models q^n(\mathbf{a})$. To do this, we produce maps $h_0, \ldots, h_n$ such that h_i is a homomorphism from q^i to $\mathcal{A}_1$ with $h_i(\mathbf{x}) = \mathbf{a}$ and such that $h_i(z_j) = b$ if $z_j \in \text{var}(q^i)$, for all $i \leq n$. Start with $h_0 = h$. To produce h_{i+1} from h_i, first restrict h_i to the remainder of q^i after the removals in Step 2 were carried out. Then extend h_i to cover all fresh elements introduced via the subtrees $\exists r.H$ in Step 4. Note that, since $\mathcal{A}_0 \models E(b)$ and $h_i(z_i) = b$, this is possible. For the same reason, the resulting homomorphism h_i' respects all the concept assertions added in Step 3. Finally, to deal with the minimization in Step 5, restrict h_i' to $\mathsf{var}(q^{i+1})$.

By construction of the queries $q^0, \ldots, q^n$ and the homomorphisms $h_0, \ldots, h_n$, there is no atom in q^n such that the image of the atom under h_n is in $\mathcal{A}_1 \setminus \mathcal{A}_0$. To show this, assume to the contrary that there is such an atom $A(x)$ in q^n. There are two cases:

1. $h(x) = b$.
   Then $x = z_i$ for some i. Since $A(h(x)) = A(b)$ was added by the application of $E \sqsubseteq A$, the atom $A(x)$ was removed in Step 1 when constructing q^{i+1} from q^i, in contradiction to $A(x)$ being in q^n.
2. $h(x) \neq b$.
   Then $h(x)$ is a non-root node of the sub-ABox $\mathcal{A}_{\exists r.G}(b)$ of $\mathcal{A}_1$, and $F = \exists r.G$. There is an answer variable $x_j \in \mathbf{x}$ such that there is a path from x_j to x, i.e. a sequence of individuals $x_j = y_0, \ldots, y_\ell = x$ such that $r_i(y_i, y_{i+1}) \in q^n$ for some r_i, for all $i < \ell$. We find a corresponding path $h(y_0), \ldots, h(y_\ell)$ in $\mathcal{A}_1$, and since $\mathcal{A}_G$ has been linked to $\mathcal{A}_0$ only by $r(b, d)$, the individual b must be on that second path. Let y_p be such that $h(y_p) = b$. We must have $y_p = z_i$ for some i, and $r_p = r$. Note that by $(*)$, y_{p+1} is the root of a tree subquery q' of q^n with link $r(z_i, y_{p+1})$ (recall that links are unique). Homomorphism h_n maps y_{p+1} to d, so we have $\models F \sqsubseteq \exists r.q'$. Consequently, the subtree of q^n rooted at y_{p+1} was removed in Step 2 when constructing q^{i+1}, in contradiction to $A(x)$ being in q^n.

The case of role atoms is similar to subcase 2 above, but simpler (we know that $F = \exists r.G$). We have thus shown that there is no atom in q^n such that the image of this atom under h_n is in $\mathcal{A}_1 \setminus \mathcal{A}_0$. Consequently $\mathcal{A}_0 \models q^n(\mathbf{a})$ via h_n. As $\mathcal{A}_0$ is a Σ-ABox, it holds that $q^n \in R|_\Sigma$, and we are done with part 1.

Now for the second part of Lemma 5. We prove the contrapositive, using a result from [5]:

Fact. Let $Q = (\mathcal{T}, \Sigma, q_0)$ be an OMQ from $(\mathcal{EL}, \text{rCQ})$. Q is FO-rewritable iff there is a $k \geq 0$ such that for all pseudo ditree Σ-ABoxes $\mathcal{A}$ of outdegree at most $|\mathcal{T}|$ and width at most $|q|$: if $\mathcal{A} \models Q(\mathbf{a})$ with $\mathbf{a}$ from the core of $\mathcal{A}$, then $\mathcal{A}|_{\leq k} \models Q(\mathbf{a})$.

We refrain from giving a detailed definition of the notions used in the above statement and only mention that, informally, a pseudo ditree ABox $\mathcal{A}$ of width i is a tree-shaped ABox (with all edges pointing downwards and without multi-edges) whose root has been replaced by an ABox with at most i individuals, called the core of $\mathcal{A}$. The outdegree refers to the non-core part of $\mathcal{A}$, and $\mathcal{A}|_{\leq k}$ means the result of removing all nodes from the tree part of $\mathcal{A}$ that are of depth exceeding k, that is, that are more than k steps away from the core.

It is straightforward to verify that every query q ever added to $\mathsf{bc}_{\text{rCQ}}(Q)$, viewed as an ABox $\mathcal{A}_q$, is a pseudo ditree Σ-ABox of width at most $|q|$ such that $\mathcal{A}_q \models Q(\mathbf{a})$ where $\mathbf{a}$ are the individuals in $\mathcal{A}_q$ that correspond to the answer variables in q. Using that q is $\prec$-minimal with $q \subseteq_\mathcal{T} q_0$, it can be shown that removing any subtree from $\mathcal{A}_q$ results in an ABox $\mathcal{A}'$ with $\mathcal{A}' \not\models Q(\mathbf{a})$. We say that $\mathcal{A}_q$ is *$\prec$-minimal with* $\mathcal{A}_q \models Q(\mathbf{a})$. This, in turn, can be used to prove in a standard way that $\mathcal{A}_q$ has outdegree at most $|\mathcal{T}|$. By the above fact and the $\prec$-minimality of $\mathcal{A}_q$, the depth of $\mathcal{A}_q$ is thus at most k. Clearly, there are only finitely many pseudo ditree Σ-ABoxes of bounded width, outdegree and depth. $\square$

We next prove Proposition 2, starting with some preliminaries. Let $Q = (\mathcal{T}, \Sigma, q_0)$ be an OMQ from $(\mathcal{EL}, \text{rCQ})$ and q_r a fork rewriting of q_0.

A conformant tCQ q' can be converted into a corresponding CQ q for Q, as detailed before Proposition 2. For easier reference, we use $\pi(q')$ to denote q. Conversely, let q be a derivative of q_r in the sense that q can be obtained from the restriction of q_r to the variables in $\mathsf{core}(q_r)$ by adding tree-shaped CQs rooted at variables in $\mathsf{core}(q_r)$. We can translate q into a *corresponding CQ q' for Q_{q_r}* as follows: replace every atom $A(x)$, $x \in \mathsf{core}(p)$, with $A^x(x_0)$; every atom $r(x, y)$, $x \in \mathsf{core}(p)$ and $y \notin \mathsf{core}(p)$, with $r^x(x_0, y)$; delete all atoms $r(x_1, x_2)$, $x_1, x_2 \in \mathsf{core}(p)$. The answer variable in q' is x_0. We use $\tau(q)$ to denote the query q'.

It can be verified that τ produces conformant tUCQs, and that $\pi(\tau(q)) = q$. Moreover, both π and π^- are injective, π translates Σ'-queries into Σ-queries, and τ translates Σ-queries into Σ'-queries.

When q is a derivative of q_r, then $p \prec^* q$ is a *$\prec$-minimization* of q if p is minimal with $p \subseteq_\mathcal{T} q_0$. Note that this is exactly the minimization carried out in Step 2 of the bc_{rCQ} algorithm started on Q. When q' is a conformant tCQ, then $p' \prec^* q'$ is a *$\prec$-minimization* of q' if p' is minimal with $\mathcal{T}^{\mathsf{min}}_{q_r} \models p' \sqsubseteq N$. Note that this is exactly the minimization carried out in Step 2 of the $\mathsf{bc}^+_{\text{AQ}}$ algorithm started on $(Q_{q_r}, \mathcal{T}^{\mathsf{min}}_{q_r})$.

Lemma 6. *Let $Q = (\mathcal{T}, \Sigma, q_0)$ be an OMQ from $(\mathcal{EL}, rCQ)$ and q_r a fork rewriting of q_0. Then*

1. *if q is a derivative of q_r and p a $\prec$-minimization of q, then $\tau(p)$ is a $\prec$-minimization of $\tau(q)$;*
2. *if q' is a conformant tCQ and p a $\prec$-minimization of $\pi(q')$, then there is a $\prec$-minimization p' of q' with $\pi(p') = p$;*
3. *if q' is a conformant tCQ and p' a $\prec$-minimization of q', then $\pi(p')$ is a $\prec$-minimization of $\pi(q')$.*

Proof. We only sketch a proof of Point 1, Points 2 and 3 are established very similarly.

Recall that $\prec$-minimization of $\tau(q)$ is based on the TBox $\mathcal{T}_{q_r}^{\mathsf{min}}$ and that, due to Lemma 1, for any query $q' \prec^* \tau(q)$ it holds that $\mathcal{T}_{q_r}^{\mathsf{min}} \models \bigsqcap_{C(x) \text{ a tree in } q'} C_R^x \sqsubseteq N$ iff $\mathcal{A}_{q'}, \mathcal{T} \models q_0(\mathbf{a})$ for some tuple $\mathbf{a}$ that can be obtained by starting with the tuple $\mathbf{x}$ of answer variables in q_0, then potentially replacing each variable x from $\mathbf{x}$ with a variables from $[x]_{q'}$, and finally replacing each variable with the corresponding individual name in $\mathcal{A}_{q'}$. It is standard to prove that this, in turn, is the case iff $q' \subseteq_\mathcal{T} q_0$, which is what minimization of q is based on. □

Proposition 2. *Let $Q = (\mathcal{T}, \Sigma, q_0)$ be an OMQ from $(\mathcal{EL}, rCQ)$. If $\mathsf{bc}^+_{AQ}(Q_{q_r}, \mathcal{T}_{q_r}^{\mathsf{min}})$ 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 $\mathsf{bc}^+_{AQ}(Q_{q_r}, \mathcal{T}_{q_r}^{\mathsf{min}})$. Otherwise, Q is not FO-rewritable.*

Proof. Let $Q = (\mathcal{T}, \Sigma, q_0)$ be an OMQ from $(\mathcal{EL}, \text{rCQ})$. We prove Proposition 2 by relating the runs of $\mathsf{bc}^+_{\text{AQ}}(q_r)$, q_r a fork rewriting of q_0, with the run of $\mathsf{bc}_{\text{rCQ}}(Q)$.

Recall that $\mathsf{bc}_{\text{rCQ}}(Q)$ initializes R with a set that contains for each fork rewriting q_r of q_0, a CQ $p \prec^* q_r$ that is $\prec$-minimal. For easier reference, we denote this initial set R with R_0. We use $\mathsf{min}(q_r)$ to denote p, thus $R_0 = \{\mathsf{min}(q_r) \mid q_r \text{ fork rewriting of } q_0\}$. Note that the different queries in R_0 do not interact during the run of $\mathsf{bc}_{\text{rCQ}}(Q)$, that is, the final set R can be written as $\bigcup_{p \in R_0} R_p$ where R_p denotes the result of starting with the set $R = \{p\}$ and then exhaustively applying Steps 1 and 2 of bc_{rCQ}.

It follows from Point 1 of Lemma 6 that whenever $\mathsf{min}(q_r) = \mathsf{min}(q_r')$ for two fork rewritings q_r and q_r' of q_0, then we can guide the very first minimization (after replacing the concept name N) during the runs of $\mathsf{bc}^+_{\text{AQ}}(Q_{q_r}, \mathcal{T}_{q_r}^{\mathsf{min}})$ and $\mathsf{bc}^+_{\text{AQ}}(Q_{q_r'}, \mathcal{T}_{q_r'}^{\mathsf{min}})$ (which involve 'don't care non-determinism') such that, in both cases, they query $\tau(p)$ is added to M. Consequently, the sets M computed in the limit are identical. It therefore suffices to consider for each $p \in R_0$ one run $\mathsf{bc}^+_{\text{AQ}}(Q_{q_r}, \mathcal{T}_{q_r}^{\mathsf{min}})$ such that $\mathsf{min}(q_r) = p$. We use M_p to denote the (finite or infinite) set of CQs M generated by such a run. Our main aim is to show the following.

Claim. For each $p \in R_0$, $R_p = \{\pi(q') \mid q' \in M_p\}$.

We argue that this establishes Proposition 2 and then prove the claim.

First let Q be FO-rewritable. Assume to the contrary of what we have to show that $\mathsf{bc}^+_{\mathrm{AQ}}(Q_{q_r}, \mathcal{T}^{\mathsf{min}}_{q_r})$ is infinite for some fork rewriting q_r. Let $\mathsf{min}(q_r) = p$. Then the set M_p contains infinitely many Σ'-queries, thus R_p contains infinitely many Σ-queries by injectivity of π. By Lemma 5, this means that Q is not FO-rewritable, a contradiction. We also have to show that

$$\bigvee_{p \text{ fork rewriting for } q_0} \quad \bigvee_{q' \in M_p|_\Sigma} \pi(q')$$

is a rewriting of Q. However, by the claim the above query is simply $\bigvee R|_\Sigma$, R the set computed by $\mathsf{bc}_{\mathrm{rCQ}}(Q)$. It thus suffices to invoke Lemma 5.

Conversely, let Q be non-FO-rewritable. By Lemma 5, $\mathsf{bc}_{\mathrm{rCQ}}(Q)$ is infinite and thus $R_p|_\Sigma$ is infinite for at least one p. By injectivity of π^-, $M_p|_\Sigma$ is infinite.

We now prove the claim. Let $p \in R_0$.

"$\subseteq$". Let $q \in R_p$. Then there is a sequence of CQs $p = q_1, \ldots, q_m = q$ such that each q_{i+1} is obtained from q_i by applying Steps 1 and 2 of the $\mathsf{bc}_{\mathrm{rCQ}}$ algorithm. We prove by induction on i that $\tau(q_i) \in M_p$, thus $q = \pi(\tau(q)) \in \{\pi(q') \mid q' \in M_p\}$ as required.

For the induction start, note that $\mathsf{bc}^+_{\mathrm{AQ}}(Q_{q_r}, \mathcal{T}^{\mathsf{min}}_{q_r})$ initializes M_p with $\{N(x)\}$, that $\bigsqcap_{C(x) \text{ a tree in } q_r} C^x_R \sqsubseteq N$ is in $\mathcal{T}_{q_r}$, and that the left-hand side of this CI is nothing but $\tau(q_r)$. Thus, $\mathsf{bc}^+_{\mathrm{AQ}}(Q_{q_r}, \mathcal{T}^{\mathsf{min}}_{q_r})$ adds a minimization of q_r to M. By Point 1 of Lemma 6, we can assume this minimization to be p, thus $p \in M$ as required.

For the induction step, assume that q_{i+1} is obtained from q_i by application of a concept inclusion $\alpha = C \sqsubseteq D \in \mathcal{T}$ at x, resulting in a CQ $\widehat{q}$, and subsequent minimization of $\widehat{q}$ according to Step 2 of the $\mathsf{bc}_{\mathrm{rCQ}}$ algorithm. We first show that it is possible to apply a concept inclusion $\alpha' \in \mathcal{T}_{q_r}$ at a variable x' in $\tau(q_i)$ such that the resulting CQ is $\tau(\widehat{q})$. There are two cases:

1. $x \notin \mathsf{core}(p)$. Then x and the subtree below it are present in $\tau(q_i)$. We apply $\alpha' = \alpha$ at $x' = x$ in $\tau(q_i)$.
2. $x \in \mathsf{core}(p)$. We apply $C^x_R \sqsubseteq D^x_R \in \mathcal{T}_{q_r}$ at x_0 in $\tau(q_i)$.

In both cases, it can be verified that the resulting query is $\tau(\widehat{q})$. It remains to apply Point 1 of Lemma 6: since $\widehat{q}$ is a derivative of q_r and q_{i+1} is the result of minimizing $\widehat{q}$, we can minimize $\tau(\widehat{q})$ to obtain $\tau(q_{i+1})$.

"$\supseteq$". Let $q \in M_p$. Then M_p contains a sequence of $\mathcal{EL}$-concepts $N(x) = q_0, \ldots, q_m = q$ such that each q_{i+1} is obtained from q_i by applying Steps 1 and 2 of the $\mathsf{bc}^+_{\mathrm{AQ}}$ algorithm, using TBox $\mathcal{T}_{q_r}$ in Step 1 and $\mathcal{T}^{\mathsf{min}}_{q_r}$ in Step 2. We prove by induction on i that $\pi(q_i) \in R_p$ for $1 \leq i \leq m$.

For the induction start, note that M_p is initialized to $\{N(x)\}$, and the concept inclusion $\bigsqcap_{C(x) \text{ a tree in } q_r} C^x_R \sqsubseteq N$ is in $\mathcal{T}_{q_r}$; no other concept inclusion can be applied to $N(x)$. Consequently, Step 1 of the $\mathsf{bc}^+_{\mathrm{AQ}}$ algorithm produces a query $\widehat{q}$ that is the left-hand side of this concept inclusion. By Point 2 of Lemma 6, we can assume w.l.o.g. that the minimization q_0 of $\widehat{q}$ satisfies $\pi(q_0) = p$, thus $q_0 \in R_p$.

For the induction step, assume that q_{i+1} is obtained from q_i by application of a concept inclusion $\alpha \in \mathcal{T}_{q_r}$ at x, yielding a query $\widehat{q}$, and subsequent minimization based on $\mathcal{T}_{q_r}^{\mathsf{min}}$. We show that a concept inclusion $\alpha' \in \mathcal{T}$ is applicable in $\pi(q_i)$ to yield q_{i+1}. There are two possibilities:

1. α is applied at a variable $x \neq x_0$ in q_i, thus is not of the form $E_R^x \sqsubseteq F_R^x$, as q_i is conformant. Then x and the subtree below it are present in $\pi(q_i)$. The concept inclusion α, which is present in $\mathcal{T}$, can be applied at x in $\pi(q_i)$, so $\alpha' = \alpha$.
2. $\alpha = E_R^x \sqsubseteq F_R^x$ is applied at x_0 of q_i. Application of α results in:
   (a) removal of atom $A^x(x_0)$ if $F_R^x = A^x$;
   (b) removal of all existential subtrees rooted at some y, together with the atom $r(x_0, y)$, whenever $r(x_0, y) \in q_i$ and $\models F \sqsubseteq \exists r.(q_i|_y)$;
   (c) adding of $A^x(x_0)$ if $F_R^x = A^x$;
   (d) adding of $r(x_0, y)$ and G as a CQ with root y if $F_R^x = \exists r.G$.

   The concept inclusion $\alpha' = E \sqsubseteq F \in \mathcal{T}$ is applicable at x in $\pi(q_i)$. Removal of atoms in (a) and (b) is done correspondingly at variable x in $\pi(q_i)$, the same holds for the adding of concept names or subtrees in (c) and (d).

In both cases, it can be verified that the resulting query is $\pi(\widehat{q})$. It remains to observe that by Point 3 of Lemma 6, $\mathsf{bc}_{\mathrm{rCQ}}$ can minimize $\pi(\widehat{q})$ to produce $\pi(q_{i+1})$. $\square$