Deciding FO-Rewritability in $\mathcal{EL}$

Meghyn Bienvenu[1] and Carsten Lutz[2] and Frank Wolter[3]

[1] LRI - CNRS & Université Paris Sud, France
[2] Department of Computer Science, University of Bremen, Germany
[3] Department of Computer Science, University of Liverpool, UK

Abstract. We consider the problem of deciding, given an instance query $A(x)$, an $\mathcal{EL}$-TBox $\mathcal{T}$, and possibly an ABox signature Σ, whether $A(x)$ is FO-rewritable relative to $\mathcal{T}$ and Σ-ABoxes. Our main results are PSPACE-completeness for the case where Σ comprises all symbols and EXPTIME-completeness for the general case. We also show that the problem is in PTIME for classical TBoxes and that every instance query is FO-rewritable into a polynomial-size FO query relative to every (semi)-acyclic TBox (under some mild assumptions on the data).

1 Introduction

Over the last years, query answering over instance data has developed into one of the most prominent problems in description logic (DL) research. Many approaches aim at utilizing relational databases systems (RDBMSs), exploiting their mature technology, advanced optimization techniques, and the general infrastructure that those systems offer. Roughly, RDBMS-based approaches can be classified into *query rewriting approaches*, where the original query and the DL TBox are compiled into an SQL query that is passed to the RDBMS for execution [5], and *combined approaches*, where the consequences of the TBox are materialized in the data in a compact form and some query rewriting is used to ensure correct answers despite the compact representation [12, 11]. This division is by no means strict, as illustrated by the approach presented in [7] which is based on query rewriting, but also has strong similarities with combined approaches.

A fundamental difference between the query rewriting approach and the combined approach is that, in query rewriting, an exponential blowup of the query is often unavoidable [8] while the combined approach typically blows up both query and data only polynomially [12, 11]. It is thus unsurprising that query execution is more efficient in the combined approach than in the query rewriting approach, see the experiments in [11]. Depending on the application, however, there can still be good reasons to use pure query rewriting. *Ease of implementation*: Query rewriting approaches are often easier to implement as they do not involve a data completion phase. When the TBox is sufficiently small so that the exponential blowup of the query is not prohibitive or when only a prototype implementation is aimed at, it may not be worthwhile to implement a full combined approach. *Access limitations*: If the user does not have permission to modify the data in the database, materializing the consequences of the TBox in the data might simply be out of the question. This problem arises notably in information integration applications.

In this paper, we are interested in TBoxes formulated in the description logic $\mathcal{EL}$, which forms the basis of the OWL EL fragment of OWL 2 and is popular as a basic language for large-scale ontologies. In general, query rewriting approaches are not applicable to $\mathcal{EL}$ because instance query answering in this DL is PTIME-complete regarding data complexity while AC_0 data complexity marks the boundary of DLs for which the pure query rewriting approach can be made work [5]. For example, the query $A(x)$ cannot be answered by an SQL-based RDBMS in the presence of the very simple $\mathcal{EL}$-TBox $\mathcal{T} = \{\exists r.A \sqsubseteq A\}$, intuitively because $\mathcal{T}$ forces the concept name A to be *propagated unboundedly* along r-chains in the data and thus the rewritten query would have to express transitive closure of r. We say that $A(x)$ is not *FO-rewritable relative to* $\mathcal{T}$, alluding to the known equivalence of first-order (FO) formulas and SQL queries.

Of course, such an isolated example does not rule out the possibility that *some* $\mathcal{EL}$-TBoxes, including those that are used in applications, still enjoy FO-rewritability. For example, the query $A(x)$ is FO-rewritable relative to the $\mathcal{EL}$-TBox $\mathcal{T}' = \{A \sqsubseteq \exists r.A\}$: since the additional instances of A stipulated by $\mathcal{T}'$ are 'anonymous objects' (nulls in database parlance) rather than primary data objects, there is no unbounded propagation *through the data* and, in fact, we can simply drop $\mathcal{T}'$ when answering $A(x)$. Inspired by these observations, the aim of this paper is to study FO-rewritability on the level of individual TBoxes, essentially following the non-uniform approach initiated in [15]. In particular, we are interested in deciding, for a given instance query (IQ) $A(x)$ and $\mathcal{EL}$-TBox $\mathcal{T}$, whether q is FO-rewritable relative to $\mathcal{T}$. Sometimes, we additionally allow as a third input an ABox-signature Σ that restricts the symbols which can occur in the data [2, 3].

Our main result is that deciding FO-rewritability of IQs relative to *general* $\mathcal{EL}$*-TBoxes* (sets of concept inclusions $C \sqsubseteq D$) is PSPACE-complete when the ABox signature Σ is full (i.e., all symbols are allowed in the ABox) and EXPTIME-complete when Σ is given as an input. For proving these results, we establish some properties that are of independent interest, such as: (1) whenever an IQ is FO-rewritable, then it is FO-rewritable into a union of tree-shaped conjunctive queries; (2) an IQ is FO-rewritable relative to all ABoxes iff it is FO-rewritable relative to *tree-shaped* ABoxes (see Section 3 for a precise formulation). We also study more restricted forms of TBoxes, showing that FO-rewritability of IQs relative to *classical TBoxes* (sets of concept definitions $A \equiv C$ and concept implications $A \sqsubseteq C$ with A atomic, cycles allowed) is in PTIME, even when Σ is part of the input. For *semi-acyclic TBoxes* $\mathcal{T}$ (classical TBoxes without cycles that involve only concept definitions, but potentially with cycles that involve at least one concept inclusion), we observe that every IQ is FO-rewritable relative to $\mathcal{T}$ (for any ABox signature Σ) and that, under the mild assumption that the admitted databases have domain size at least two, even a polynomial-sized rewriting is possible. While it is not our primary aim in this first publication to actually generate FO-rewritings, we note that all our results come with effective procedures for doing this (the rewritings are of triple-exponential size in the worst case).

Although we focus on simple IQs of the form $A(x)$, all results in this paper also apply to instance queries of the form $C(x)$ with C an $\mathcal{EL}$-concept. The treatment of conjunctive queries (CQs) is left for future work. We also discuss the connection of FO-rewritability in $\mathcal{EL}$ to boundedness in datalog and in the μ-calculus. Proof details are deferred to the long version at at http://www.informatik.uni-bremen.de/$\sim$clu/papers/.

2 Preliminaries

We remind the reader that $\mathcal{EL}$-concepts are built up from concept names and the concept $\top$ using conjunction $C \sqcap D$ and existential restriction $\exists r.C$. When we speak of a *TBox* without further qualification, we mean a *general TBox*, i.e., a finite set of *concept inclusions (CIs)* $C \sqsubseteq D$. Other forms of TBoxes will be introduced later as needed. An *ABox* is a finite set of *concept assertions* $A(a)$ and *role assertions* $r(a, b)$ where A is a concept name, r a role name, and a, b individual names. We use $\mathsf{Ind}(\mathcal{A})$ to denote the set of all individual names used in $\mathcal{A}$. It will sometimes be convenient to view an ABox $\mathcal{A}$ as an interpretation $\mathcal{I}_\mathcal{A}$, defined in the obvious way (see [15]).

Regarding query languages, we focus on *instance queries (IQ)*, which have the form $A(x)$ with A a concept name and x a variable. We write $\mathcal{T}, \mathcal{A} \models A(a)$ if $a^\mathcal{I} \in A^\mathcal{I}$ for all models $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{A}$ and call a a *certain answer* to $A(x)$ given $\mathcal{A}$ and $\mathcal{T}$. We use $\mathsf{cert}_\mathcal{T}(A(x), \mathcal{A})$ to denote the set of all certain answers to $A(x)$ given $\mathcal{A}$ and $\mathcal{T}$. To define FO-rewritability, we require first-order queries (FOQs), which are first-order formulas constructed from atoms $A(x)$, $r(x, y)$, and $x = y$. We use $\mathsf{ans}(\mathcal{I}, q)$ to denote the set of all answers to the FOQ q in the interpretation $\mathcal{I}$.

A *signature* is a set of concept and role names, which are uniformly called *symbols* in this context. A *Σ-ABox* is an ABox that uses only concept and role names from Σ. The *full* signature is the signature that contains all concept and role names.

Definition 1 (FO-rewritability). *Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox and Σ an ABox signature. An IQ q is* FO-rewritable relative to *$\mathcal{T}$ and Σ if there is a FOQ φ such that $\mathsf{cert}_\mathcal{T}(\mathcal{A}, q) = \mathsf{ans}(\mathcal{I}_\mathcal{A}, \varphi)$ for all Σ-ABoxes $\mathcal{A}$. Then φ is an* FO-rewriting of q relative to $\mathcal{T}$ and Σ.

Example 1. Recall from the introduction that $A(x)$ is not FO-rewritable relative to $\mathcal{T} = \{\exists r.A \sqsubseteq A\}$ and the full signature. If we add $\exists r.\top \sqsubseteq A$ to $\mathcal{T}$, then $A(x)$ is FO-rewritable relative to the resulting TBox and the full signature, and $\varphi(x) = A(x) \vee \exists y\, r(x, y)$ is an FO-rewriting. If we choose $\Sigma = \{A\}$, then $A(x)$ becomes FO-rewritable also relative to the original $\mathcal{T}$, with the trivial FO-rewriting $A(x)$. Conversely, if a query q is FO-rewritable relative to a TBox $\mathcal{T}'$ and a signature Σ, then q is FO-rewritable relative to $\mathcal{T}'$ and any $\Sigma' \subseteq \Sigma$ (take an FO-rewriting relative to $\mathcal{T}'$ and Σ and replace all atoms which involve predicates that are not in Σ' with false).

Sometimes, instance queries have the more general form $C(x)$ with C an $\mathcal{EL}$-concept. Since $C(x)$ is FO-rewritable relative to $\mathcal{T}$ and Σ whenever $A(x)$ is FO-rewritable relative to $\mathcal{T} \cup \{A \equiv C\}$ and Σ, A a fresh concept name, queries of this form are captured by the results in this paper.

3 General TBoxes – Upper Bounds

We first characterize failure of FO-rewritability of an IQ $A(x)$ relative to a TBox $\mathcal{T}$ and an ABox signature Σ in terms of the existence of certain Σ-ABoxes and then show how to decide the latter. The following result provides the starting point. An ABox is called *tree-shaped* if the directed graph $(\mathsf{Ind}(\mathcal{A}), \{(a, b) \mid r(a, b) \in \mathcal{A}\})$ is a tree and $r(a, b), s(a, b) \in \mathcal{A}$ implies $r = s$. A FOQ is a *tree-UCQ* if it is a disjunction

$q_1 \vee \cdots \vee q_n$ and each q_i is a conjunctive query (CQ) that is tree-shaped (defined in analogy with tree-shaped ABoxes) and where the root is the only answer variable; see e.g. [15] for details on CQs.

Theorem 1. *Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox, Σ an ABox signature, and $A(x)$ an IQ. Then*

1. *If $A(x)$ is FO-rewritable relative to $\mathcal{T}$ and Σ, then there is a tree-UCQ that is an FO-rewriting of $A(x)$ relative to $\mathcal{T}$ and Σ;*
2. *If $\varphi(x)$ is an FO-rewriting of $A(x)$ relative to $\mathcal{T}$ and tree-shaped Σ-ABoxes and $\varphi(x)$ is a tree-UCQ, then $\varphi(x)$ is an FO-rewriting relative to $\mathcal{T}$ and Σ;*
3. *$A(x)$ is FO-rewritable relative to $\mathcal{T}$ and Σ iff $A(x)$ is FO-rewritable relative to $\mathcal{T}$ and tree-shaped Σ-ABoxes.*

Of the three points in Theorem 1, Point 1 is most laborious to prove. It involves applying an Ehrenfeucht-Fraïssé game and explicitly constructing a tree-UCQ as a disjunction of certain $\mathcal{EL}$-concepts (c.f. the characterization of FO-rewritability in terms of datalog boundedness given in [15] and its proof). Point 2 can then be derived from Point 1, and Point 3 is an immediate consequence of Points 1 and 2.

For a tree-shaped ABox $\mathcal{A}$ and $k \geq 0$, we use $\mathcal{A}|_k$ to denote the restriction of $\mathcal{A}$ to depth k. The following provides the first version of the announced characterization of FO-rewritability in terms of the existence of certain ABoxes.

Theorem 2. *Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox, Σ an ABox signature, and $A(x)$ an IQ. Then $A(x)$ is not FO-rewritable relative to $\mathcal{T}$ and Σ iff for every $k \geq 0$, there is a tree-shaped Σ-ABox $\mathcal{A}$ of depth exceeding k with root a_0 s.t. $\mathcal{T}, \mathcal{A} \models A(a_0)$ and $\mathcal{T}, \mathcal{A}|_k \not\models A(a_0)$.*

The proof of Theorem 2 builds on Point 1 of Theorem 1. Note that if $A(x)$ is FO-rewritable relative to $\mathcal{T}$ and Σ, then there is a $k \geq 0$ such that for all tree-shaped Σ-ABoxes $\mathcal{A}$ of depth exceeding k with root a_0, $\mathcal{T}, \mathcal{A} \models A(a_0)$ implies $\mathcal{T}, \mathcal{A}|_k \models A(a_0)$. In the proof of Theorem 2, we explicitly construct FO-rewritings which are tree-UCQs of outdegree at most $|\mathcal{T}|$ and depth at most k.

To proceed, it is convenient to work with TBoxes in *normal form*, where all CIs must be of one of the forms $A \sqsubseteq B_1$, $A \sqsubseteq \exists r.B$, $\top \sqsubseteq A$, $B_1 \sqcap B_2 \sqsubseteq A$, $\exists r.B \sqsubseteq A$ with A, B, B_1, B_2 concept names. This can be assumed without loss of generality:

Lemma 1. *For any $\mathcal{EL}$-TBox $\mathcal{T}$, ABox signature Σ, and IQ $A(x)$, there is a TBox $\mathcal{T}'$ in normal form such that for any FOQ φ, we have that $\varphi(x)$ is an FO-rewriting of $A(x)$ relative to $\mathcal{T}$ and Σ iff $\varphi(x)$ is an FO-rewriting of $A(x)$ relative to $\mathcal{T}'$ and Σ.*

To exploit Theorem 2 for building a decision procedure for FO-rewritability, we impose a bound on k. The next theorem is proved using Theorem 2 and a pumping argument.

Theorem 3. *Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox in normal form, Σ an ABox signature, $A(x)$ an IQ, and $n = |(\mathsf{sig}(\mathcal{T}) \cup \Sigma) \cap \mathsf{N_C}|$. Then $A(x)$ is not FO-rewritable relative to $\mathcal{T}$ and Σ iff there exists a tree-shaped Σ-ABox $\mathcal{A}$ of depth exceeding 2^n with root a_0 such that $\mathcal{T}, \mathcal{A} \models A(a_0)$ and $\mathcal{T}, \mathcal{A}|_{2^n} \not\models A(a_0)$.*

Note that, with the remark after Theorem 2, we obtain a triple exponential upper bound on the size of FO-rewritings. The bound in Theorem 3 is optimal in the sense that, for every $n \geq 1$, there is an $\mathcal{EL}$-TBox $\mathcal{T}$ and an IQ $A(x)$ such that $|\mathsf{sig}(\mathcal{T}) \cap \mathsf{N_C}| = n$,

$A(x)$ is FO-rewritable relative to $\mathcal{T}$ and the full Σ, and for all ABoxes of depth at least 2^n with root a_0, we have $\mathcal{T}, \mathcal{A} \models A(a_0)$ iff $\mathcal{T}, \mathcal{A}|_{2^n-1} \models A(a_0)$. Such a $\mathcal{T}$ can be constructed by simulating a binary counter, see Section 4 of [13]. Based on Theorem 3, we can establish the following result.

Theorem 4. *Deciding FO-rewritability of an IQ relative to an $\mathcal{EL}$-TBox and an ABox signature is in* EXPTIME.

The proof utilizes non-deterministic bottom-up automata on finite, ranked trees: we construct exponential-size automata that accept precisely the ABoxes $\mathcal{A}$ from Theorem 3 and then decide their emptiness in PTIME.

When Σ is full, the characterization given in Theorem 3 can be further improved. An ABox $\mathcal{A}$ is *linear* if it consists of role assertions $r_0(a_0, a_1), \dots, r_{n-1}(a_{n-1}, a_n)$ and concept assertions $A(a)$ with $a \in \{a_0, \dots, a_n\}$. Somewhat unexpectedly, with full Σ we can replace the tree-shaped ABoxes from Theorem 3 with linear ones.

Theorem 5. *Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox in normal form, $A(x)$ an IQ, and $n = |(\mathsf{sig}(\mathcal{T}) \cup \Sigma) \cap \mathsf{N_C}|$. Then $A(x)$ is not FO-rewritable relative to $\mathcal{T}$ (and the full Σ) iff there exists a linear ABox $\mathcal{A}$ of depth exceeding 2^n with root a_0 such that $\mathcal{T}, \mathcal{A} \models A(a_0)$ and $\mathcal{T}, \mathcal{A}|_{2^n} \not\models A(a_0)$.*

The surprisingly subtle proof of Theorem 5 is based on the careful extraction of a linear ABox from the tree-shaped one whose existence is guaranteed by Theorem 3. The subtlety is largely due to the fact that it is not sufficient to simply select a linear chain of individuals from the tree-shaped ABox; additionally, the concept assertions on that chain have to be modified in a very careful way.

The following example shows that, when Σ is not full, tree-shaped ABoxes in Theorem 3 cannot be replaced with linear ones.

Example 2. Let $\mathcal{T} = \{A_i \sqsubseteq X_i,\ B_i \sqcap X_i \sqsubseteq Y_i,\ \exists r.Y_i \sqsubseteq X_i \mid i \in \{1,2\}\} \cup$
$\{X_1 \sqcap X_2 \sqsubseteq X,\ B_1 \sqcap B_2 \sqsubseteq Z,\ \exists r.Z \sqsubseteq X\}$,
$\Sigma = \{A_1, A_2, B_1, B_2, r\}$, and take the IQ $X(x)$. The tree-shaped ABox

$$\mathcal{A} = \{r(a_0, a_{i,0}), r(a_{i,0}, a_{i,1}), \dots, r(a_{i,2^n-1}, a_{i,2^n}) \mid i \in \{1,2\}\} \cup$$
$$\{B(a_{i,0}), \dots, B(a_{i,2^n}), A_i(a_{i,2^n}) \mid i \in \{1,2\}\},$$

with n as in Theorems 3 and 5, is of depth exceeding 2^n and it can be verified that $\mathcal{T}, \mathcal{A} \models X(a_0)$, but $\mathcal{T}, \mathcal{A}|_{2^n} \not\models X(a_0)$. However, for all linear Σ-ABoxes $\mathcal{A}$, we have $\mathcal{T}, \mathcal{A} \models X(a_0)$ iff $\mathcal{T}, \mathcal{A}|_1 \models X(a_0)$.

Theorem 5 allows us to replace the non-deterministic tree automata in the proof of Theorem 6 with word automata, improving the upper bound to PSPACE.

Theorem 6. *Deciding FO-rewritability of an IQ relative to an $\mathcal{EL}$-TBox and the full ABox signature is in* PSPACE.

4 General TBoxes – Lower Bounds

We establish lower bounds that match the upper bounds from the previous section.

Theorem 7. *Deciding FO-rewritability of an IQ relative to a general $\mathcal{EL}$-TBox and an ABox signature Σ is (1)* PSPACE*-hard when Σ is full and (2)* EXPTIME*-hard when Σ is an input.*

The proof of Point 1 is by reduction of the word problem of polynomially space-bounded deterministic Turing machines (DTMs). For Point 2, we modify the proof of Point 1 to yield a reduction of the word problem of polynomially space-bounded alternating Turing machines (ATMs). We start with the former.

Let $M = (Q, \Omega, \Gamma, \delta, q_0, q_{\mathsf{acc}}, q_{\mathsf{rej}})$ be a DTM and $p(\cdot)$ its polynomial space bound. We assume w.l.o.g. that M terminates on every input, that it never attempts to move left on the left-most end of the tape, that $q_0 \notin \{q_{\mathsf{acc}}, q_{\mathsf{rej}}\}$, and that there are no transitions defined for q_{acc} and q_{rej}. Let $x \in \Omega^*$ be an input to M of length n. Our aim is to construct a TBox $\mathcal{T}$ and select a concept name B such that B is *not* FO-rewritable relative to $\mathcal{T}$ and the full signature iff M accepts x.

By Theorem 5, non-FO-rewritability of B w.r.t. $\mathcal{T}$ is witnessed by a sequence of linear ABoxes of increasing depth. In the reduction, these ABoxes take the form of longer and longer r-chains (with r a role name). The chains represent the computation of M on x, repeated over and over again. Specifically, the tape contents, the current state, and the head position are represented using the elements of $\Gamma \cup (\Gamma \times Q)$ as concept names. If, for example, $x = ab$ and the computation of M on x consists of the two configurations qab and $aq'b$,[4] then this is represented by ABoxes of the form

$$\{r(b_0, b_1), r(b_1, b_2), r(b_2, b_3), \ldots, r(b_{n-1}, b_n)\}$$

where additionally, the concept (q, a) is asserted for $b_0, b_4, b_8, \ldots$, a is asserted for $b_1, b_5, b_9, \ldots$ and for $b_2, b_6, b_{10}, \ldots$, and (q', b) for $b_3, b_7, b_{11}, \ldots$. If M accepts x, then B is propagated backwards along these chains (from b_0 to b_1 to b_2 etc) unboundedly far, starting from a single explicit occurrence of B asserted for b_0. If M rejects x or the chain in the ABox does not properly represent the computation of M on x, then B will already be implied by any subchain of length at most $p(n)^2$ and thus the unbounded propagation of B gets 'disrupted' resulting in FO-rewritability of B relative to $\mathcal{T}$.

The following CI in $\mathcal{T}$ results in backwards propagation of B provided that every ABox individual is labeled with at least one symbol from $\Gamma \cup \Gamma \times Q$:

$$\exists r.(A \sqcap B) \sqsubseteq B \text{ for all } A \in \Gamma \cup (\Gamma \times Q). \tag{1}$$

Disrupt the propagation of B when M does not accept x:

$$(a, q_{\mathsf{rej}}) \sqsubseteq B \text{ for all } a \in \Gamma.$$

We have to enforce that the ABox actually represents a (repeated) computation of M. To do this, we again use disruptions of the propagation of B: whenever an ABox $\mathcal{A}$

[4] $uqv \in \Gamma^* Q \Gamma^*$ means that M is in state q, the tape left of the head is labeled with u, and starting from the head position, the remaining tape is labeled with v.

represents a configuration sequence that is not a proper computation, then B is implied by a sub-chain of bounded length. Let forbid denote the set of all tuples (A_1, A_2, A_3, A) with $A_i \in \Gamma \cup (\Gamma \times Q)$ such that whenever three consecutive tape cells in a configuration c are labeled with A_1, A_2, A_3, then in the successor configuration c' of c, the tape cell corresponding to the middle cell *cannot* be labeled with A. Put

$$A \sqcap \exists r^{p(n)+1}.A_1 \sqcap \exists r^{p(n)}.A_2 \sqcap \exists r^{p(n)-1}.A_3 \sqsubseteq B \tag{2}$$

for all $(A_1, A_2, A_3, A) \in$ forbid. This ensures that the transition relation is respected and that the content of tape cells which are not under the head does not change. We also need to say that every tape cell has a unique label, that there is at not more than one head position per configuration, and at least one, again via disrupting the propagation of B:

$$\begin{array}{rlrl} A \sqcap A' \sqsubseteq B & \text{for all distinct } A, A' \in \Gamma \cup (\Gamma \times Q) & & \\ (a, q) \sqsubseteq H & \text{for all } (a, q) \in \Gamma \times Q & a \sqsubseteq \overline{H} & \text{for all } a \in \Gamma \\ \exists r^i.H \sqcap \exists r^j H \sqsubseteq B & \text{for } i < j < p(n) & \overline{H} \sqcap \exists r.\overline{H} \sqcap \cdots \sqcap \exists r^{p(n)-1}.\overline{H} \sqsubseteq B & \end{array}$$

where H is a concept name indicating that the head is on the current cell and $\overline{H}$ indicating that this is not the case. It remains to set up the initial configuration. It is tempting to introduce a concept name I that sets up the initial configuration and must be used at the end of the r-chain to start the propagation of B. However, since we assume the ABox signature Σ to be full, we can always put B itself at the end of the chain, avoiding I. To fix this issue, we refrain from introducing I, but utilize the final states q_{acc} and q_{rej}, enforcing that they must always be followed by the initial configuration. Let $A_1^{(0)}, \ldots, A_m^{(0)}$ be the concept names that describe the initial configuration, i.e., when the input x is $x_1 \cdots x_n$, then $A_0^{(0)} = (x_1, q_0)$, $A_i^{(0)} = x_i$ for $2 \leq i \leq n$, and $A_i^{(0)} = x_i$ is the blank symbol for $n < i \leq p(n)$. Now put

$$\exists r^i.q_{\mathsf{acc}} \sqsubseteq A_i^{(0)} \quad \text{and} \quad \exists r^i.q_{\mathsf{rej}} \sqsubseteq A_i^{(0)} \qquad \text{for } 1 \leq i \leq p. \tag{3}$$

Note that all witness ABoxes for non-FO-rewritability of B must eventually contain q_{acc} or q_{rej}, thus the initial configuration will be properly set up at some point: ABoxes $\mathcal{A}$ that do not contain these states do not represent a proper computation of M because M reaches q_{acc} or q_{rej} after at most $p(n)$ states, thus the propagation of B is disrupted in $\mathcal{A}$.

We now come to Point 2 of Theorem 7. When the ABox signature Σ is not required to be full, it is much simpler to set up the initial configuration. Indeed, we can then simply introduce the mentioned concept name I, add $I \sqsubseteq B$ to $\mathcal{T}$, and set $\Sigma = \Gamma \cup (\Gamma \times Q) \cup \{r, I\}$ to ensure that we must use I to start the propagation of B. We can further replace the CIs (3) above with $\exists r^i.I \sqsubseteq A_i^{(0)}$ and $\exists r^i.I \sqsubseteq A_i^{(0)}$ for $1 \leq i \leq p$ to ensure that I sets up the initial configuration as intended. With this modification, we can adapt the reduction from DTMs to ATMs in a straightforward way, thus improving the PSPACE lower bound to an EXPTIME one. We only give a brief sketch. Assume w.l.o.g. that each configuration of the ATM M has at most two successor configurations. We introduce a second role name $s \in \Sigma$, reserving r for the first successor

of a configuration and s for the second successor. Then the CIs (1) are replaced with

$$\exists r.(A \sqcap B) \sqcap \exists s.(A' \sqcap B) \sqsubseteq B \text{ for all } A, A' \in \Gamma \cup (\Gamma \times Q).$$

resulting in witness ABoxes to take the form of a tree-shaped ATM computation rather than a linear DTM computation. It remains to adapt the CIs (2) to reflect the branching of computations. The set forbid now contains tuples $(A_1, A_2, A_3, A'_1, A'_2, A'_3, A)$, where A_1, A_2, A_3 describe three neighboring cells in the left predecessor configuration and A'_1, A'_2, A'_3 the corresponding cells in the right predecessor configuration (for existential states of M, we simply assume that the left and right predecessor are identical). We can then replace (2) with

$$A \sqcap \exists r^{p(n)+1}.A_1 \sqcap \exists r^{p(n)}.A_2 \sqcap \exists r^{p(n)-1}.A_3 \sqcap \exists s^{p(n)+1}.A'_1 \sqcap \exists s^{p(n)}.A'_2 \sqcap \exists s^{p(n)-1}.A'_3 \sqsubseteq B.$$

5 Classical TBoxes

A *classical TBox* $\mathcal{T}$ is a finite set of *concept definitions* $A \equiv C$ and CIs $A \sqsubseteq C$ where A is a concept name. No concept name is allowed to occur more than once on the left hand side of a statement in $\mathcal{T}$.

Theorem 8. *Deciding FO-rewritability of an IQ relative to a classical $\mathcal{EL}$-TBox and an ABox signature is in* PTIME.

We give examples illustrating FO-rewritability in classical TBoxes and give the main idea of the proof.

Example 3. (a) The IQ $A(x)$ is not FO-rewritable relative to the classical TBox $\{A \equiv \exists r.A\}$ and the full ABox signature.

(b) The concept name A has a cyclic definition in the TBox $\mathcal{T} = \{\ A \equiv B \sqcap \exists r.A, B \sqsubseteq \exists r.A\ \}$ which often indicates non-FO-rewritability, but in this case the IQ $A(x)$ has an FO-rewriting relative to $\mathcal{T}$ and the full ABox signature, namely $\varphi(x) = A(x) \vee B(x)$.

To present the idea of the proof, we use an appropriate normal form for classical TBoxes. A concept name A is *defined in* $\mathcal{T}$ if there is a definition $A \equiv C \in \mathcal{T}$ and *primitive* otherwise; A is *non-conjunctive* in $\mathcal{T}$ if it occurs in $\mathcal{T}$, but there is no CI of the form $A \equiv B_1 \sqcap \cdots \sqcap B_n$ in $\mathcal{T}$ with $n \geq 1$ and $B_1, \ldots, B_n$ concept names. We use non-conj$(\mathcal{T})$ to denote the set of non-conjunctive concepts in $\mathcal{T}$. A classical TBox $\mathcal{T}$ is in *normal form* if it is a set of statements $A \equiv \exists r.B$ and $A \equiv B_1 \sqcap \cdots \sqcap B_n$ where $B, B_1, \ldots, B_n$ are concept names and $B_1, \ldots, B_n$ are non-conjunctive. For every classical TBox $\mathcal{T}$, one can construct in polynomial time a classical TBox $\mathcal{T}'$ in normal form that uses additional concept names such that $\mathcal{T}' \models \mathcal{T}$ and every model of $\mathcal{T}$ can be expanded to a model of $\mathcal{T}'$ [10]. It is not hard to verify that an IQ $A(x)$ is FO-rewritable relative to $\mathcal{T}$ and Σ if and only if it is FO-rewritable relative to $\mathcal{T}'$ and Σ, provided that A is not among the new concept names introduced during the construction of $\mathcal{T}'$. For a classical TBox $\mathcal{T}$ in normal form and a concept name A, define

$$\text{non-conj}_{\mathcal{T}}(A) = \begin{cases} \{A\} & \text{if } A \text{ is non-conjunctive in } \mathcal{T} \\ \{B_1, \ldots, B_n\} & \text{if } A \equiv B_1 \sqcap \cdots \sqcap B_n \in \mathcal{T}. \end{cases}$$

Our polytime algorithm utilizes an ABox introduced in [10, 9] in the context of conservative extensions and logical difference: given a classical TBox $\mathcal{T}$ in normal form and an ABox signature Σ, we compute in polytime a polysize Σ-ABox $\mathcal{A}_{\mathcal{T},\Sigma}$ with individual names a_B, B non-conjunctive in $\mathcal{T}$, such that for any Σ-ABox $\mathcal{A}$, individual name a in $\mathcal{A}$, and concept name A the following conditions are equivalent:

- $\mathcal{T}, \mathcal{A} \not\models A(a)$;
- there exists $B \in \text{non-conj}_{\mathcal{T}}(A)$ such that $(\mathcal{A}, a)$ is simulated by $(\mathcal{A}_{\mathcal{T},\Sigma}, a_B)$ (see appendix of long version of this paper).

It follows that to check whether $A(x)$ is FO-rewritable, instead of considering arbitrary tree-shaped Σ-ABoxes $\mathcal{A}$ and $\mathcal{A}|_k$ as in Theorem 2, it suffices to consider the tree unfolding of $\mathcal{A}_{\mathcal{T},\Sigma}$ at a_B and its restriction to depth k, for all $B \in \text{non-conj}_{\mathcal{T}}(A)$. The original search problem has been reduced to the problem of analysing the tree unfolding of $\mathcal{A}_{\mathcal{T},\Sigma}$. A polytime algorithm performing that analysis is given in the long version.

6 Semi-Acyclic TBoxes

It is easy to see that every IQ is FO-rewritable relative to every acyclic $\mathcal{EL}$-TBox and every ABox signature Σ. We observe that the same holds for semi-acyclic TBoxes, where some cycles are still allowed, and that it is possible to find rewritings of polynomial size when only databases of domain size at least two are admitted.

A *semi-acyclic TBox* is defined like a classical TBox, except that definitorial cycles are disallowed, i.e., there cannot be concept definitions $A_0 \equiv C_0, \ldots, A_{n-1} \equiv C_{n-1}$ such that A_i occurs in $C_{i+1 \bmod n}$. Note that cycles via concept *inclusions*, such as $A \sqsubseteq \exists r.A$, are still permitted. Let $\mathcal{T}$ be a semi-acyclic TBox and Σ an ABox signature. For an $\mathcal{EL}$-concept C, we use $\mathsf{pre}_{\mathcal{T},\Sigma}(C)$ to denote the FO-formula $\bigvee_{B \in \Sigma \mid \mathcal{T} \models B \sqsubseteq C} B(x)$. For all concept names A and $\mathcal{EL}$-concepts C and D and role names r, set

$$
\begin{aligned}
\varphi^{\Sigma}_{\top,\mathcal{T}}(x) &= \mathsf{true} && \\
\varphi^{\Sigma}_{A,\mathcal{T}}(x) &= \mathsf{pre}_{\mathcal{T},\Sigma}(A) && \text{if } A \text{ is primitive} \\
\varphi^{\Sigma}_{A,\mathcal{T}}(x) &= \varphi^{\Sigma}_{C,\mathcal{T}}(x) && \text{if } A \equiv C \in \mathcal{T} \\
\varphi^{\Sigma}_{C \sqcap D,\mathcal{T}}(x) &= \varphi^{\Sigma}_{C,\mathcal{T}}(x) \wedge \varphi^{\Sigma}_{D,\mathcal{T}}(x) && \\
\varphi^{\Sigma}_{\exists r.C,\mathcal{T}}(x) &= \mathsf{pre}_{\mathcal{T},\Sigma}(\exists r.C) \vee \exists y.(r(x,y) \wedge \varphi^{\Sigma}_{C,\mathcal{T}}[y/x]) && \text{if } r \in \Sigma \\
\varphi^{\Sigma}_{\exists r.C,\mathcal{T}}(x) &= \mathsf{pre}_{\mathcal{T},\Sigma}(\exists r.C) && \text{if } r \notin \Sigma
\end{aligned}
$$

where $\varphi[x/y]$ denotes the result of first renaming all bound variables in φ so that y does not occur, and then replacing the free variable x of φ with y.

Lemma 2. *For all IQs $A(x)$, $\varphi^{\Sigma}_{A,\mathcal{T}}(x)$ is an FO-rewriting of $A(x)$ relative to $\mathcal{T}$ and Σ.*

The size of $\varphi^{\Sigma}_{A,\mathcal{T}}(x)$ can clearly be exponential in the size of $\mathcal{T}$, for example when $A = A_n$ and $\mathcal{T} = \{A_i \equiv \exists r.A_{i-1} \sqcap \exists s.A_{i-1} \mid 1 \leq i \leq n\}$. To reduce $\varphi^{\Sigma}_{A,\mathcal{T}}$ to polynomial size, we can use Avigad's observation that FO supports structure sharing [1]. More precisely, let φ be a positive FOQ (such as $\varphi^{\Sigma}_{A,\mathcal{T}}$) whose subformulas

include $\psi(x_1), \ldots, \psi(x_n)$. The multiple occurences of ψ can be avoided by rewriting φ to $\exists u \forall y \forall z \big(\ (\psi(y) \leftrightarrow z = u) \rightarrow \varphi' \big)$ where φ' is φ with each $\psi(x_i)$ replaced with $y = x_i \rightarrow z = u$. Intuitively, we iterate over all y and memorize whether $\psi(y)$ holds using identity of z with u. Since we need at least two different 'values' for z to make this trick work, the resulting FOQ is an FO-rewriting only on ABoxes with at least two individual names.

7 Related Work

In [15], deciding FO-rewritability is studied in the context of the expressive DL $\mathcal{ALCFI}$ and several of its fragments. In general, though, the setup in that paper is different: while we are interested in deciding FO-rewritability of a single query relative to a TBox, the results in [15] concern deciding whether, for a given TBox $\mathcal{T}$, *all* queries are FO-rewritable relative to $\mathcal{T}$. It is shown that this problem is decidable for Horn-$\mathcal{ALCFI}$-TBoxes of depth at most two and for Horn-$\mathcal{ALCF}$-TBoxes (queries are IQs or, equivalently, CQs). As a by-product of these results, a close connection between FO-rewritability of TBoxes formulated in Horn DLs and boundedness of datalog programs is observed, see e.g. [6, 17] for the latter problem. In its original formulation, the following result is established for a larger class of TBoxes, namely materializable $\mathcal{ALCFI}$-TBoxes of depth one.

Lemma 3 ([15]). *For every (general) $\mathcal{EL}$-TBox $\mathcal{T}$ in normal form, there is a datalog program $\Pi_\mathcal{T}$ such that for every ABox signature Σ and IQ $A(x)$, the predicate A is bounded in $\Pi_\mathcal{T}$ relative to Σ-databases iff $A(x)$ is FO-rewritable relative to $\mathcal{T}$ and Σ.*

In [15], the program $\Pi_\mathcal{T}$ is of exponential size. Since we are only interested in $\mathcal{EL}$-TBoxes, it is easy to find a $\Pi_\mathcal{T}$ of polynomial size. More specifically, $\Pi_\mathcal{T}$ consists of

$$
\begin{array}{lll}
A(x) \leftarrow \mathsf{true} & \text{if } \top \sqsubseteq A \in \mathcal{T} & \\
B(x) \leftarrow r(x,y), A(x) & \text{if } \exists r.A \sqsubseteq B \in \mathcal{T} & X_{\exists r.A}(x) \leftarrow B(x) \ \text{ if } B \sqsubseteq \exists r.A \in \mathcal{T} \\
B(x) \leftarrow A_1(x), A_2(x) & \text{if } A_1 \sqcap A_2 \sqsubseteq B \in \mathcal{T} & \text{(where possibly } A_1 = A_2\text{)} \\
B(x) \leftarrow X_{\exists r.A}(x) & \text{if } \exists r.B_0 \sqsubseteq B \in \mathcal{T} \text{ and } \mathcal{T} \models A \sqsubseteq B_0 &
\end{array}
$$

This allows to carry over the 2ExpTime upper bound for predicate boundedness of connected monadic datalog programs [6] to FO-rewritability of an IQ relative to a general $\mathcal{EL}$-TBoxes and an ABox signature.[5]

Note that boundedness has been studied also in the context of the μ-calculus and monadic second order (MSO) logic [16, 4]. Here, an ExpTime upper bound is known from [16] and it seems likely that this result can be utilized to find an alternative proof of Theorem 6. In particular, it is possible to find a *μ-calculus rewriting* φ of an IQ $A(x)$ relative to an $\mathcal{EL}$-TBox $\mathcal{T}$ and ABox signature Σ: proceeding similarly to the construction of the above datalog program $\Pi_\mathcal{T}$, we can find a μ-calculus formula $\varphi_{\mathcal{T},\Sigma}$ such that for all Σ-ABoxes $\mathcal{A}$ and $a \in \mathsf{Ind}(\mathcal{A})$, we have $\mathcal{T}, \mathcal{A} \models A(a)$ iff $\mathcal{I}_\mathcal{A}, a \models \varphi$. When simultaneous fixpoints are admitted, φ even has polynomial size.

[5] Note that we explicitly fix the signature Σ of the databases over which boundedness of $\Pi_\mathcal{T}$ is considered instead of assuming that only the EDB predicates can be used in the data as in [6]; this is only for simplicity and, in fact, it is easy to adapt $\Pi_\mathcal{T}$ to the latter assumption.

8 Conclusions

It would be interesting to generalize the results presented in this paper to more expressive DLs and to more expressive query languages. Regarding the former, we note that using the techniques in [14, 15] it is possible to derive a NEXPTIME upper bound for deciding FO-rewritability of IQs relative to Horn-$\mathcal{ALCI}$-TBoxes and ABox signatures. Regarding the latter, CQs are a natural choice and we believe that a mix of techniques from this paper and those in [3] might provide a good starting point. It is interesting to note that FO-rewritability of all IQ-atoms $A(x)$ in a CQ q does not imply that q is FO-rewritable and the converse fails, too.

Acknowledgements. C. Lutz was supported by the DFG SFB/TR 8 'Spatial Cognition'.

References

1. J. Avigad. Eliminating definitions and skolem functions in first-order logic. In Proc. of LICS, pages 139–146. IEEE Computer Society, 2001.
2. F. Baader, M. Bienvenu, C. Lutz, and F. Wolter. Query and predicate emptiness in description logics. In Proc. of KR. AAAI Press, 2010.
3. M. Bienvenu, C. Lutz, and F. Wolter. Query containment in description logics reconsidered. In Proc. of KR, 2012. To appear.
4. A. Blumensath, M. Otto, and M. Weyer. Decidability results for the boundedness problem. Manuscript, 2012.
5. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. J. of Automated Reasoning, 39(3):385–429, 2007.
6. S. S. Cosmadakis, H. Gaifman, P. C. Kanellakis, and M. Y. Vardi. Decidable optimization problems for database logic programs. In Proc. of STOC, pages 477–490. ACM, 1988.
7. G. Gottlob and T. Schwentick. Rewriting ontological queries into small nonrecursive datalog programs. In Proc. of DL. CEUR-WS, 2011.
8. S. Kikot, R. Kontchakov, V. V. Podolskii, and M. Zakharyaschev. Exponential lower bounds and separation for query rewriting. CoRR, abs/1202.4193, 2012.
9. B. Konev, M. Ludwig, D. Walther, and F. Wolter. The logical diff for the lightweight description logic $\mathcal{EL}$. Technical report, U. of Liverpool, http://www.liv.ac.uk/~frank/publ/, 2011.
10. B. Konev, D. Walther, and F. Wolter. The logical difference problem for description logic terminologies. In Proc. of IJCAR, pages 259–274. Springer, 2008.
11. R. Kontchakov, C. Lutz, D. Toman, F. Wolter, and M. Zakharyaschev. The combined approach to query answering in DL-Lite. In Proc. of KR. AAAI Press, 2010.
12. C. Lutz, D. Toman, and F. Wolter. Conjunctive query answering in the description logic $\mathcal{EL}$ using a relational database system. In Proc. of IJCAI, pages 2070–2075. AAAI Press, 2009.
13. C. Lutz and F. Wolter. Deciding inseparability and conservative extensions in the description logic $\mathcal{EL}$. In J. of Symbolic Computation 45(2): 194-228, 2010.
14. C. Lutz and F. Wolter. Non-uniform data complexity of query answering in description logics. In Proc. of DL. CEUR-WS, 2011.
15. C. Lutz and F. Wolter. Non-uniform data complexity of query answering in description logics. In Proc. of KR, 2012. To appear.
16. M. Otto. Eliminating recursion in the μ-calculus. In Proc. of STACS, pages 531–540. Springer, 1999.
17. R. van der Meyden. Predicate boundedness of linear monadic datalog is in PSPACE. Int. J. Found. Comput. Sci., 11(4):591–612, 2000.