Efficient Query Rewriting in the Description Logic $\mathcal{EL}$ and Beyond

Peter Hansen and **Carsten Lutz** and **İnanç Seylan**
University of Bremen, Germany
{hansen, clu, seylan}@informatik.uni-bremen.de

Frank Wolter
University of Liverpool, UK
frank@csc.liv.ac.uk

Abstract

We propose a new type of algorithm for computing first-order (FO) rewritings of concept queries under $\mathcal{ELH}^{dr}$-TBoxes. The algorithm is tailored towards efficient implementation, yet complete. It outputs a succinct non-recursive datalog rewriting if the input is FO-rewritable and otherwise reports non-FO-rewritability. We carry out experiments with real-world ontologies which demonstrate excellent performance in practice and show that TBoxes originating from applications admit FO-rewritings of reasonable size in almost all cases, even when in theory such rewritings are not guaranteed to exist.

1 Introduction

Ontology-based data access (OBDA) with Description Logics (DLs) is a very active topic of research, which has resulted in various approaches to implementing OBDA in practice [Poggi *et al.*, 2008; Lutz *et al.*, 2009; Pérez-Urbina *et al.*, 2010; Eiter *et al.*, 2012]. A particularly promising such approach is query rewriting, which enables implementations based on conventional relational database systems (RDBMSs), thus taking advantage of those systems' efficiency and maturity [Poggi *et al.*, 2008; Kontchakov *et al.*, 2013]. The general idea is to transform the original query q and the relevant TBox $\mathcal{T}$ into a first-order (FO) query $q_{\mathcal{T}}$ that is then handed over to the RDBMS for execution. One limitation of this approach is that, for the majority of description logics that are used as ontology languages, the query $q_{\mathcal{T}}$ is not guaranteed to exist. In fact, this is the case already for the members of the popular $\mathcal{EL}$ family of lightweight DLs [Baader *et al.*, 2005; Lutz *et al.*, 2009], which underly the EL profile of the OWL2 ontology language and are frequently used in health care and biology ontologies. This observation, however, does not rule out the possibility that FO-rewritings still exist in many practically relevant cases. In fact, TBoxes that emerge from practical applications tend to have a rather simple structure and one might thus speculate that, indeed, FO-rewritings under $\mathcal{EL}$-TBoxes tend to exist in practice.

In this paper, we consider the construction of FO-rewritings of concept queries under TBoxes that are formulated in the description logic $\mathcal{ELH}^{dr}$. The latter is a logical core of OWL2 EL that extends basic $\mathcal{EL}$ with role inclusions and domain/range restrictions on roles [Lutz *et al.*, 2009] while concept queries take the form $C(x)$ with C an $\mathcal{EL}$-concept. Constructing the desired rewritings is computationally hard: it follows from results in [Bienvenu *et al.*, 2013] that deciding whether a given concept query $C(x)$ is FO-rewritable under a given TBox $\mathcal{T}$ is PSPACE-complete both in $\mathcal{EL}$ and in $\mathcal{ELH}^{dr}$, and it even becomes EXPTIME-complete when the vocabulary of the admitted database instances (i. e., ABoxes) is an additional input—a feature that we allow in the current paper.

Existing approaches to query rewriting under DL TBoxes can be summarized as follows: (i) approaches that target rewritings into the more expressive query language datalog and which are incomplete in the sense that existing datalog-rewritings are not guaranteed to be found and, moreover, the generated datalog-rewritings are not necessarily non-recursive even if there is an FO-rewriting [Rosati, 2007; Pérez-Urbina *et al.*, 2010; Kaminski and Grau, 2013; Mora and Corcho, 2013]; (ii) backwards chaining approaches for existential rules (a strict generalization of $\mathcal{ELH}^{dr}$) which are complete in the sense that they find an FO-rewriting if there is one, but need not terminate otherwise [König *et al.*, 2012]; (iii) complete and terminating approaches which aim to prove upper complexity bounds, but which are not practically feasible [Bienvenu *et al.*, 2013; 2014].

The aim of this paper is *to design, for the first time, algorithms for computing FO-rewritings of concept queries under $\mathcal{EL}$- and $\mathcal{ELH}^{dr}$-TBoxes that are complete, terminating, and feasible in practice.* We start with a marriage of approaches (ii) and (iii) above to get the best of both worlds; in particular, (ii) appears to be practically much more feasible than (iii) while (iii) provides a way to achieve termination. The resulting backwards chaining algorithm is conceptually simple and constitutes a significant step towards our goal. However, it produces FO-rewritings that are unions of conjunctive queries (UCQs), which has two significant drawbacks: first, recent experiments [Lutz *et al.*, 2013] have shown that executing UCQ-rewritings on RDBMSs is prohibitively expensive while executing equivalent rewritings that take the form of non-recursive datalog programs is much more feasible; and second, UCQ-rewritings can be of excessive size even in practically relevant cases [Rosati and Almatelli, 2010].

To address these shortcomings, the main contribution of

this paper is to refine our initial backwards chaining algorithm to what we call a *decomposed algorithm*. While the initial algorithm uses tree-shaped conjunctive queries (CQs) as an internal data structure, the new algorithm only represents single nodes of tree-shaped CQs together with information of how to reassemble these nodes into full queries. This can be seen as a way to implement structure sharing and allows us to directly produce rewritings that are non-recursive datalog programs, avoiding UCQ-rewritings altogether. The algorithm runs in exponential time, is capable of deciding the existence of FO-rewritings in EXPTIME (which is optimal), and is capable of producing monadic non-recursive datalog rewritings that are of at most exponential size (but much smaller in practice). We also show that an exponential blowup is unavoidable when rewriting into monadic non-recursive datalog. Technically, the decomposed algorithm is much more subtle than the initial one.

We then evaluate the decomposed algorithm by carrying out experiments with seven ontologies from practical applications. We ask for an FO-rewriting of every concept query $A(x)$ with A a concept name from the ontology under consideration. Out of 10989 inputs in total and with a timeout of 30 seconds, the decomposed algorithm terminates on all but 127 inputs and on average needs less than 0.5 seconds per input. We also analyse the size of the generated non-recursive datalog rewritings, with extremely encouraging results. Our experiments show that the decomposed algorithm performs very well on ontologies from practical applications and also confirm our initial belief that such ontologies very often admit FO-rewritings. In particular, only 74 of 10989 inputs turn out to be not FO-rewritable.

Proof details are deferred to the appendix, available at http://www.informatik.uni-bremen.de/tdki/p.html.

2 Preliminaries

We use $\mathsf{N_C}$ and $\mathsf{N_R}$ to denote mutually disjoint countably infinite sets of concept and role names, respectively. An *$\mathcal{EL}$-concept* is formed according to the syntax rule $C ::= A \mid \top \mid C \sqcap C \mid \exists r.C$, where $A \in \mathsf{N_C}$ and $r \in \mathsf{N_R}$. Let C, D be $\mathcal{EL}$-concepts and r, s be role names. Then $C \sqsubseteq D$ is a *concept inclusion (CI)*, $r \sqsubseteq s$ is a *role inclusion (RI)*, $\mathsf{dom}(r) \sqsubseteq C$ is a *domain restriction*, and $\mathsf{ran}(r) \sqsubseteq C$ is a *range restriction*. An *$\mathcal{ELH}$-TBox* is a finite set of CIs and RIs and an $\mathcal{ELH}^{dr}$-TBox additionally admits domain and range restrictions. The semantics of concepts and TBoxes is defined as usual [Baader *et al.*, 2007; 2005]. For a CI or RI α, we write $\mathcal{T} \models \alpha$ if every model of $\mathcal{T}$ satisfies α; when $\mathcal{T}$ is empty, we write $\models \alpha$.

An ABox $\mathcal{A}$ is a set of *assertions* of the form $A(a)$ and $r(a, b)$ with A a concept name, r a role name, and a, b individual names from a countably infinite set $\mathsf{N_I}$. We use $\mathsf{ind}(\mathcal{A})$ to denote the set of individual names that occur in $\mathcal{A}$. A *concept query* is an expression $C(x)$ with C an $\mathcal{EL}$-concept. We write $\mathcal{A}, \mathcal{T} \models C(a)$ and say that *a is a certain answer to C given the ABox $\mathcal{A}$ and TBox $\mathcal{T}$* if $a \in \mathsf{ind}(\mathcal{A})$ and $a^{\mathcal{I}} \in C^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{A}$. A *signature* is a finite set $\Sigma \subseteq \mathsf{N_C} \cup \mathsf{N_R}$. An ABox is a *$\Sigma$-ABox* if it only uses concept and role names from Σ.

For an FO-formula $q(x)$ with one free variable x, we write $\mathcal{A} \models q(a)$ if $\mathcal{A}$ (viewed as an interpretation) satisfies q under the assignment that maps x to a. A concept query $C(x)$ is *FO-rewritable under a TBox $\mathcal{T}$ and signature Σ* if there is an FO-formula $\varphi(x)$ such that for all Σ-ABoxes $\mathcal{A}$ and individuals a, we have $\mathcal{A}, \mathcal{T} \models C(a)$ iff $\mathcal{A} \models \varphi(a)$. In this case, we call $\varphi(x)$ an *FO-rewriting* of $C(x)$ under $\mathcal{T}$ and Σ. When studying FO-rewritability of concept queries $C(x)$, we can assume w. l. o. g. that C is a concept name since $C(x)$ is FO-rewritable under a TBox $\mathcal{T}$ and Σ iff $A(x)$ is FO-rewritable under $\mathcal{T} \cup \{C \sqsubseteq A\}$ and Σ, where A is a fresh concept name [Bienvenu *et al.*, 2013].

We will also consider other forms of rewritings, in particular into *unions of conjunctive queries (UCQs)* and *non-recursive datalog programs*. Although, strictly speaking, non-recursive datalog rewritings are not FO-rewritings, the existence of either kind of rewriting coincides with the existence of an FO-rewriting [Bienvenu *et al.*, 2014]. In particular, non-recursive datalog rewritings can be viewed as a compact representation of a UCQ-rewriting. We will also consider monadic datalog as a special case, where all intensional (IDB) predicates are unary.

For any of these rewritings, if Σ is the set of all concept and role names in $\mathcal{T}$ and C, then $C(x)$ is rewritable under $\mathcal{T}$ and Σ iff it is rewritable under $\mathcal{T}$ and any signature. If this is the case, we say that $C(x)$ is rewritable under $\mathcal{T}$ and the *full signature*.

For our technical constructions, it will be convenient to view $\mathcal{EL}$-concepts as conjunctive queries (CQs) that take the form of a directed tree. We will represent such queries as sets of atoms of the form $A(x)$ and $r(x, y)$ with A a concept name, r a role name and x, y variables. Tree-shapedness of a conjunctive query q then means that the directed graph $(V, \{(x, y) \mid r(x, y) \in q\})$ is a tree, where V is the set of variables in q, and that $r(x, y), s(x, y) \in q$ implies $r = s$. We will not distinguish explicitly between an $\mathcal{EL}$-concept C and its query representation. We thus use $\mathsf{var}(C)$ to denote the set of variables that occur in C and x_ε to denote the root variable in C. For an $x \in \mathsf{var}(C)$, we use $C|_x$ to denote the $\mathcal{EL}$-concept represented by (the subtree rooted at) x. When we speak of a *top-level conjunct (tlc)* of an $\mathcal{EL}$-concept C, we mean a concept name A such that $A(x_\varepsilon) \in C$ or a concept $\exists r.D$ such that $r(x_\varepsilon, y) \in C$ and $D = C|_y$. We use $\mathsf{tlc}(C)$ to denote the set of all top-level conjuncts of C. For any syntactic object (such as a concept or a TBox), we define its *size* to be the number of symbols used to write it.

We show that we can remove domain and range restrictions from $\mathcal{ELH}^{dr}$-TBoxes and work with $\mathcal{ELH}$-TBoxes when developing algorithms that decide rewritability or compute rewritings.

Lemma 1. *For every $\mathcal{ELH}^{dr}$-TBox $\mathcal{T}$, signature Σ, and concept query $A_0(x)$, one can construct in polynomial time an $\mathcal{ELH}$-TBox $\mathcal{T}'$ and signature Σ' such that $A_0(x)$ is FO-rewritable under $\mathcal{T}$ and Σ iff it is FO-rewritable under $\mathcal{T}'$ and Σ'.*

Moreover, every UCQ and non-recursive datalog rewriting of $A_0(x)$ under $\mathcal{T}'$ and Σ' can be converted in linear time into a UCQ and, respectively, non-recursive datalog rewriting of

A_0 *under* $\mathcal{T}$ *and* Σ.

3 A Backwards Chaining Algorithm

The algorithm presented in this section constructs a set of UCQ-rewritings of $A_0(x)$ under an $\mathcal{ELH}$-TBox $\mathcal{T}$ by starting from $\{A_0\}$ and then exhaustively applying the axioms in $\mathcal{T}$ as rules in a backwards chaining manner. It terminates by either constructing a UCQ-rewriting or returning 'not FO-rewritable'. For simplicity and since the purpose of the algorithm presented here is mainly to prepare for the subsequent, more refined one, we consider rewritability for the full signature only.

Let A_0 and $\mathcal{T}$ be an input to the algorithm. We start with introducing the central backwards chaining steps. Let C and D be $\mathcal{EL}$-concepts and let α be a (concept or role) inclusion from $\mathcal{T}$. The notion of D *being obtained from* C *by applying* α is defined as follows.

(CI) Let $\alpha = E \sqsubseteq F$ be a CI, $x \in \mathsf{var}(C)$, and let there be at least one tlc G of $C|_x$ with $\models F \sqsubseteq G$. Then D is *obtained from* C *by applying* α *at* x if D can be obtained from C by

- removing $A(x)$ for all concept names A with $\models F \sqsubseteq A$;
- removing the subtree rooted at y whenever $r(x, y) \in C$ and $\models F \sqsubseteq \exists r.(C|_y)$ (including the edge $r(x, y)$);
- adding $A(x)$ for all concept names A that are a tlc of E;
- adding the subtree $\exists r.H$ to x for each $\exists r.H$ that is a tlc of E.

(RI) Let $\alpha = r \sqsubseteq s$ be an RI and let $s(x, y) \in C$. Then D is *obtained from* C *by applying* α *at* $s(x, y)$ if D can be obtained from C by replacing $s(x, y)$ by $r(x, y)$.

The following is immediate.

Lemma 2. *If* $\mathcal{T} \models C \sqsubseteq A_0$ *and* D *can be obtained from* C *by applying some inclusion in* $\mathcal{T}$*, then* $\mathcal{T} \models D \sqsubseteq A_0$.

Apart from applying backwards chaining, our algorithm also minimises the generated concepts to attain completeness and termination. To make this precise, we introduce some notation. Let C and D be $\mathcal{EL}$-concepts and $x \in \mathsf{var}(C)$. Then $C \setminus C|_x$ denotes the concept obtained by removing from C the subtree rooted at x, and we write $C \prec D$ if there exists $x \in \mathsf{var}(D)$ such that $C = D \setminus D|_x$. We use $\prec^*$ to denote the reflexive and transitive closure of $\prec$ and say that C *is* $\prec$*-minimal with* $\mathcal{T} \models C \sqsubseteq A_0$ if $\mathcal{T} \models C \sqsubseteq A_0$ and there is no $C' \prec C$ with $\mathcal{T} \models C' \sqsubseteq A_0$. Note that if $\mathcal{T} \models C \sqsubseteq A_0$, then it is possible to find in polynomial time a $C' \prec^* C$ that is $\prec$-minimal with $\mathcal{T} \models C' \sqsubseteq A_0$.

The constructions in [Bienvenu *et al.*, 2013] suggest that, to achieve termination, we can use a certain form of blocking, similar to the blocking used in DL tableau algorithms. Let $\mathsf{sub}(\mathcal{T})$ denote the set of subconcepts of (concepts that occur in) $\mathcal{T}$. For each $\mathcal{EL}$-concept C and $x \in \mathsf{var}(C)$, we set $\mathsf{con}^C_{\mathcal{T}}(x) := \{D \in \mathsf{sub}(\mathcal{T}) \mid \mathcal{T} \models C|_x \sqsubseteq D\}$. We say that C is *blocked* if there are $x_1, x_2, x_3 \in \mathsf{var}(C)$ such that

1. x_1 is a proper ancestor of x_2 is an ancestor of x_3 and
2. $\mathsf{con}^D_{\mathcal{T}}(x_1) = \mathsf{con}^D_{\mathcal{T}}(x_2)$ for $D \in \{C, C \setminus C|_{x_3}\}$.

procedure find-rewriting($A_0(x), \mathcal{T}$)
  $M := \{A_0\}$
  while there is a $C \in M$ and a concept D such that
    1. D can be obtained from C by applying some axiom in $\mathcal{T}$ and
    2. there is no $D' \prec D$ with $D' \in M$ then
    find a $D' \prec^* D$ that is $\prec$-minimal with $\mathcal{T} \models D' \sqsubseteq A_0$
    if D' is blocked then return 'not FO-rewritable'
    else add D' to M
  return the UCQ $\bigvee M$.

Figure 1: The backwards chaining algorithm

The algorithm is formulated in Figure 1. Note that, by Lemma 2, the concept D considered in the condition of the while loop satisfies $\mathcal{T} \models D \sqsubseteq A_0$ and thus we are guaranteed to find the desired D'. Also note that each of the potentially many candidates for D' will work.

Example 3. *Let* $\mathcal{T} = \{\ \exists r.(B_1 \sqcap B_2) \sqsubseteq A_0, \exists s.B_2 \sqsubseteq B_2\ \}$. *Starting with* $M = \{A_0\}$ *and applying the first CI to* A_0*, we get* $M = \{A_0, \exists r.(B_1 \sqcap B_2)\}$. *The second CI can be applied repeatedly, starting with* $\exists r.(B_1 \sqcap B_2)$*, and yields concepts*

$$D_1 = \exists r.(B_1 \sqcap \exists s.B_2),$$
$$D_2 = \exists r.(B_1 \sqcap \exists s.\exists s.B_2),$$
$$D_3 = \exists r.(B_1 \sqcap \exists s.\exists s.\exists s.B_2),\ \ldots,$$

all of which are $\prec$*-minimal with* $\mathcal{T} \models D_i \sqsubseteq A_0$. D_3 *is blocked and the algorithm will classify* A_0 *as 'not FO-rewritable' under* $\mathcal{T}$.

Now consider the TBox $\mathcal{T}' = \mathcal{T} \cup \{B_1 \sqsubseteq B_2\}$. *Already the concept* D_1 *is not* $\prec$*-minimal with* $\mathcal{T}' \models D_1 \sqsubseteq A_0$ *and instead of* D_1, $\exists r.B_1$ *is added to* M. *At this point, rule application stops and the UCQ* $\bigvee M$ *is returned as an FO-rewriting of* A_0 *under* $\mathcal{T}'$.

$\mathcal{T}'$ *illustrates that the algorithm is incomplete without the minimization step since this step prevents generation of the blocked concept* D_3.

We now establish correctness and termination.

Theorem 4. *The algorithm in Figure 1 returns a UCQ-rewriting* $\bigvee M$ *if* A_0 *is FO-rewritable under* $\mathcal{T}$ *and the full signature and 'not FO-rewritable' otherwise.*

The generated UCQ-rewritings are not necessarily of minimal size. It is possible to attain minimal-size rewritings by using a stronger form of minimality when constructing the concept D', namely by redefining the relation "$\prec$" so that $C \preceq D$ if there is a root-preserving homomorphism from C to D (c.f. *most-general rewritings* in [König *et al.*, 2012]). As a consequence, the $\prec$-minimal concept D' with $\mathcal{T} \models D' \sqsubseteq A$ can then be of size exponential in the size of D. However, D' can still be constructed in output-polynomial time.

We prove in the appendix that all concepts in M have out-degree at most n and depth at most 2^{2n}, n the size of $\mathcal{T}$. As remarked in [Bienvenu *et al.*, 2013], the size of UCQ-rewritings can be triple exponential in the size of $\mathcal{T}$, and thus the same is true for the runtime of the presented algorithm.

While this worst case is probably not encountered in practice, the size of M can become prohibitively large for realistic inputs. For this reason, we propose an improved algorithm in the subsequent section, which produces non-recursive datalog rewritings instead of UCQ-rewritings and whose runtime is at most single exponential.

4 A Decomposed Algorithm

The algorithm presented in this section consists of three phases. In Phase 1, a set Γ is computed that can be viewed as a decomposed representation of the set M from Section 3 in the sense that we store only single nodes of the tree-shaped concepts in M, rather than entire concepts. In many cases, we can already construct a non-recursive datalog rewriting after Phase 1. If this is not possible, we compute in Phase 2 a set Ω that enriches the node representation provided by Γ with sets of logical consequences that are relevant for Point 2 of the definition of *blocked* concepts. In Phase 3, we first execute a certain cycle check on Ω, which corresponds to checking the existence of a blocked concept in M. If no cycle is found, we then construct a non-recursive datalog rewriting.

Phase 1. Assume that $\mathcal{T}$ is a TBox, Σ an ABox-signature, and A_0 a concept name for which we want to compute an FO-rewriting under $\mathcal{T}$ and Σ. To present the algorithm, it is convenient to decompose conjunctions on the right-hand side of CIs, that is, to assume that $\mathcal{T}$ consists, apart from RIs, only of CIs of the form $C \sqsubseteq A$ with A a concept name and $C \sqsubseteq \exists r.D$. We start with describing the construction of a set Γ, whose elements we call node pairs. A *node pair* has the form (C, S), where $C \in \mathsf{sub}(\mathcal{T})$, and $S \subseteq \mathsf{sub}(\mathcal{T})$ is a set of concept names and concepts of the form $\exists r.C$. Intuitively, a node pair (C, S) describes a set of concepts D such that $\mathcal{T} \models D \sqsubseteq C$ and the following conditions are satisfied:

(i) the concept names that are tlcs of D are $S \cap \mathsf{N_C}$;

(ii) the existential restrictions that are the tlcs of D are obtained from the existential restrictions in S by replacing each $\exists r.E \in S$ with some $\exists s.E'$ such that $\mathcal{T} \models \exists s.E' \sqsubseteq \exists r.E$.

The computation of Γ starts with $\{(A_0, \{A_0\})\}$ and proceeds by exhaustively applying the following two rules:

(r1) if $(C, S) \in \Gamma$, $D \sqsubseteq A \in \mathcal{T}$ and $A \in S$, then extend Γ with $(C, (S \setminus \{A\}) \cup \mathsf{tlc}(D))$.

(r2) if $(C, S) \in \Gamma$, $D \sqsubseteq \exists r.F \in \mathcal{T}$, and there is an $\exists s.G \in S$ with $\mathcal{T} \models F \sqsubseteq G$ and $\mathcal{T} \models r \sqsubseteq s$, then extend Γ with $(C, (S \setminus \{\exists s.G \mid \mathcal{T} \models F \sqsubseteq G \text{ and } \mathcal{T} \models r \sqsubseteq s\}) \cup \mathsf{tlc}(D))$.

After applying either rule, we also have to add the pair $(G, \mathsf{tlc}(G))$ for every subconcept $\exists r.G$ of D to trigger further derivation.

Example 5. *Let $\mathcal{T} = \{\exists r.(B_1 \sqcap B_2) \sqsubseteq A_0,\ \exists s.B_2 \sqsubseteq B_2\}$, and Σ the full signature. Starting with the pair $(A_0, \{A_0\})$ and applying* (r1)*, we extend Γ by $(A_0, \{\exists r.(B_1 \sqcap B_2)\})$ and $((B_1 \sqcap B_2), \{B_1, B_2\})$.* (r1) *is now applicable to the latter, yielding $((B_1 \sqcap B_2), \{B_1, \exists s.B_2\})$ and $(B_2, \{B_2\})$. Another application of* (r1) *results in e. g. $(B_2, \{\exists s.B_2\})$ being added.*

From Γ we can extract a (potentially infinitary) UCQ-rewriting of A_0 under $\mathcal{T}$ and Σ, as follows. Start with defining Γ_Σ to be the set of all $(C, S) \in \Gamma$ such that $S \cap \mathsf{N_C} \subseteq \Sigma$. Then $\widehat{\Gamma}_\Sigma$ is obtained as the limit of the sequence of sets $\widehat{\Gamma}^0_\Sigma, \widehat{\Gamma}^1_\Sigma, \ldots$ defined as follows:

- $\widehat{\Gamma}^0_\Sigma := \{(C, \bigsqcap S) \mid (C, S) \in \Gamma_\Sigma \text{ and } S \subseteq \mathsf{N_C}\}$.
- $\widehat{\Gamma}^{i+1}_\Sigma$ is $\widehat{\Gamma}^i_\Sigma$ extended with all pairs (C, D) such that there is $(C, S) \in \Gamma_\Sigma$ with the following property: for each $\exists r.G \in S$ there are $(G, C_{r,G}) \in \widehat{\Gamma}^i_\Sigma$ and $s_{r,G} \in \Sigma$ such that $\mathcal{T} \models s_{r,G} \sqsubseteq r$ and
$$D = \bigsqcap_{A \in S \cap \mathsf{N_C}} A \ \sqcap \bigsqcap_{\exists r.G \in S} \exists s_{r,G}.C_{r,G}.$$

Note that if $(C, D) \in \widehat{\Gamma}_\Sigma$, then D uses only symbols from Σ. The set $\widehat{\Gamma}_\Sigma$ represents a UCQ-rewriting as follows.

Proposition 6 (Soundness and Completeness of $\widehat{\Gamma}_\Sigma$)**.** *For all Σ-ABoxes $\mathcal{A}$ and $a \in \mathsf{ind}(\mathcal{A})$, we have $\mathcal{A}, \mathcal{T} \models A_0(a)$ iff there is an $(A_0, D) \in \widehat{\Gamma}_\Sigma$ with $\mathcal{A} \models D(a)$.*

Γ_Σ provides us with a sufficient condition for FO-rewritability of A_0 under $\mathcal{T}$ and Σ and suggests a way to (sometimes) produce a non-recursive datalog rewriting. In fact, if Γ_Σ is *acyclic* in the sense that the directed graph

$$G = (\Gamma_\Sigma, \{((C, S), (C', S')) \mid S \text{ contains } \exists r.C' \text{ with } \mathcal{T} \models s \sqsubseteq r \text{ for some } s \in \Sigma\})$$

contains no cycle, then $\widehat{\Gamma}_\Sigma$ is finite and we obtain a non-recursive datalog program Π_{Γ_Σ} that is a rewriting of A_0 under $\mathcal{T}$ and Σ by taking the rule

$$P_C(x) \leftarrow \bigwedge_{A \in S} A(x) \ \wedge \bigwedge_{\exists r.D \in S} \ \bigvee_{\mathcal{T} \models s \sqsubseteq r, s \in \Sigma} \big(s(x, y_{r,D}) \wedge P_D(y_{r,D}) \big)$$

for each $(C, S) \in \Gamma_\Sigma$ and using P_{A_0} as the goal predicate. Note that the disjunctions can be removed by introducing auxiliary (monadic) IDB predicates, without causing a significant blowup. If Γ_Σ is acyclic, we output the above rewriting. Note that it is potentially much smaller than a UCQ-rewriting since it implements structure sharing. However, if Γ_Σ is not acyclic, then A_0 could still be FO-rewritable under $\mathcal{T}$ and Σ, but the above program will be recursive.

Example 5 (continued)**.** *Γ_Σ contains the node pairs $(A_0, \{\exists r.(B_1 \sqcap B_2\})$, $(B_1 \sqcap B_2, \{B_1, B_2\})$, $(B_2, \{\exists s.B_2\})$ and is thus cyclic. This is the expected outcome since, as argued in Example 3, A_0 is not FO-rewritable under $\mathcal{T}$ and Σ*

Now, let $\mathcal{T}' := \mathcal{T} \cup \{B_1 \sqsubseteq B_2\}$ as in the second part of Example 3. The resulting Γ_Σ still contains the above node pairs and is thus cyclic. To find out that A_0 is FO-rewritable under $\mathcal{T}'$ and Σ, the algorithm enters Phase 2.

Phase 2. In the second phase of the algorithm, we construct a set of node tuples Ω_Σ by further annotating (and duplicating) the pairs in Γ_Σ. A *node tuple* takes the form $t = (C_t, S_t, \mathsf{con}_t, E_t, \mathsf{xcon}_t)$ where C_t and S_t have the same form

as the components of node pairs in Γ_Σ, $C_t \in \mathsf{con}_t \subseteq \mathsf{sub}(\mathcal{T})$, E_t is the special symbol "$-$" or of the form $\exists s.C$ such that $\exists r.C \in S_t$ for some r with $\mathcal{T} \models s \sqsubseteq r$, and xcon_t is a subset of con_t or "$-$". Intuitively, a node tuple $t \in \Omega_\Sigma$ describes a set of concepts D such that $(C_t, D) \in \widehat{\Gamma}_\Sigma$ and apart from (i) and (ii) above the following additional conditions are satisfied:

(iii) $\mathcal{T} \models D \sqsubseteq E$ iff $E \in \mathsf{con}_t$, for each $E \in \mathsf{sub}(\mathcal{T})$;

(iv) if $E_t = \exists s.C$, then there is a tlc $\exists s.E$ in D and a leaf node in E such that $(C, E) \in \widehat{\Gamma}_\Sigma$ and for the concept D' obtained from D by dropping this node, we have $\mathcal{T} \models D' \sqsubseteq E$ iff $E \in \mathsf{xcon}_t$, for each $E \in \mathsf{sub}(\mathcal{T})$.

When S_t contains no existential restrictions, we use "$-$" for E_t and xcon_t. To understand E_t, it is useful to think of D as a tree and of E_t as a selected successor of the root of that tree. We start the construction of Ω_Σ with setting

$$\Omega_\Sigma = \{(C, S \cap \mathsf{N_C}, \mathsf{con}_\mathcal{T}(S \cap \mathsf{N_C}), -, -) \mid (C, S) \in \Gamma_\Sigma\},$$

where for a set of concepts M, $\mathsf{con}_\mathcal{T}(M)$ denotes the set of concepts $D \in \mathsf{sub}(\mathcal{T})$ such that $\mathcal{T} \models \bigsqcap M \sqsubseteq D$. We call the tuples in the set above *leaf tuples*. The final set Ω_Σ is constructed by exhaustively applying the following rule:

(r_Ω) If $t = (C_t, S_t, \mathsf{con}_t, E_t, \mathsf{xcon}_t)$ is a node tuple with $\exists r_0.D_0, \dots, \exists r_n.D_n$ the existential restrictions in S_t ($n \geq 0$) and there are role names $s_0, \dots, s_n \in \Sigma$ and node tuples $t_0, \dots, t_n \in \Omega_\Sigma$ and an $\ell \in \{0, \dots, n\}$ such that the following conditions are satisfied, then add t to Ω_Σ:

1. $\mathcal{T} \models s_i \sqsubseteq r_i$ and $C_{t_i} = D_i$ for $0 \leq i \leq n$;
2. $E_t = \exists s_\ell.D_\ell$;
3. there is a node pair $(C_t, S) \in \Gamma_\Sigma$ with $S_t \subseteq S$ and $S \cap \mathsf{N_C} = S_t \cap \mathsf{N_C}$;
4. $\mathsf{con}_t = \mathsf{con}_\mathcal{T}(M)$, where
$$M = (S_t \cap \mathsf{N_C}) \cup \{\exists s_i. \bigsqcap \mathsf{con}_{t_i} \mid i \leq n\};$$
5. $\mathsf{xcon}_t = \mathsf{con}_\mathcal{T}(M')$, where
$$M' = (S_t \cap \mathsf{N_C}) \cup \{\exists s_\ell. \bigsqcap \mathsf{xcon}_{t_\ell}\} \cup \{\exists s_i. \bigsqcap \mathsf{con}_{t_i} \mid \ell \neq i \leq n\}.$$

In Point 5, $\exists r.-$ is identified with $\top$. For $t, t' \in \Omega_\Sigma$, we write $t \rightsquigarrow_{\Omega_\Sigma} t'$ if there are $t_0, \dots, t_n \in \Omega_\Sigma$ that satisfy the conditions listed in (r_Ω) and such that $t' = t_\ell$, that is, t' is the tuple that was chosen for the selected successor.

Example 5 (continued). *For the TBox $\mathcal{T}$, Ω_Σ is initialized to*

$$\{\ (A_0, \{A_0\}, \{A_0\}, -, -), \quad (t_a)$$
$$(A_0, \emptyset, \emptyset, -, -), \quad (t_b)$$
$$(B_1 \sqcap B_2, \{B_1, B_2\}, \{B_1, B_2\}, -, -),\ \dots\ \}. \quad (t_c)$$

(r_Ω) *can be applied using $(A_0, \{\exists r.(B_1 \sqcap B_2)\}) \in \Gamma_\Sigma$ for (C_t, S) in Point 3, with $n = \ell = 0$, $s_0 = r$, and $t_0 = t_c$, adding to Ω_Σ the following node tuple t:*

$$(A_0, \{\exists r.(B_1 \sqcap B_2)\}, \{\exists r.(B_1 \sqcap B_2), A_0\}, \exists r.(B_1 \sqcap B_2), \emptyset).$$

We now have $t \rightsquigarrow_{\Omega_\Sigma} t_c$. Note that $M = \{\exists r.B_1 \sqcap B_2\}$ in Point 4 and that $M' = \emptyset$ in Point 5, resulting in $\mathsf{xcon}_t = \emptyset$.

Phase 3. The third and last phase of the algorithm first checks whether an FO-rewriting exists at all and, if so, produces a rewriting that takes the form of a non-recursive monadic datalog program.

We start with introducing the relevant notion of a cycle. A tuple $t \in \Omega_\Sigma$ is a *root tuple* if $C_t = A_0$, $A_0 \in \mathsf{con}_t$ and $A_0 \notin \mathsf{xcon}_t$. A *path through* Ω_Σ is a finite sequence of node tuples $t_1, \dots, t_k$ from Ω_Σ such that $t_i \rightsquigarrow_{\Omega_\Sigma} t_{i+1}$ for $1 \leq i < k$. A tuple $t \in \Omega_\Sigma$ is *looping* if there is a path $t_1, \dots, t_k$ through Ω_Σ such that $k > 1$, $t = t_1$, $\mathsf{con}_t = \mathsf{con}_{t_k}$, and $\mathsf{xcon}_t = \mathsf{xcon}_{t_k}$. We say that Ω_Σ *contains a root cycle* if there are tuples $t, t' \in \Omega_\Sigma$ such that t is a root tuple, t' is a looping tuple, and t' is reachable from t along $\rightsquigarrow_{\Omega_\Sigma}$.

Theorem 7. *A_0 is not FO-rewritable under $\mathcal{T}$ and Σ if and only if Ω_Σ contains a root cycle.*

Example 5 (continued). *After completing Phase 2 for $\mathcal{T}$, the algorithm checks Ω_Σ for cyclicity and finds a root cycle:*

$$(A_0, \{\exists r.(B_1 \sqcap B_2)\}, \{\exists r.(B_1 \sqcap B_2), A_0\}, \exists r.(B_1 \sqcap B_2), \emptyset,$$
$$(B_1 \sqcap B_2, \{B_1, \exists s.B_2\}, \{B_1 \sqcap B_2, B_1, B_2, \exists s.B_2\}, \exists s.B_2, \{B_1\}),$$
$$(B_2, \{\exists s.B_2\}, \{\exists s.B_2, B_2\}, \exists s.B_2, \emptyset.$$

The last node tuple possesses a reflexive $\rightsquigarrow_{\Omega_\Sigma}$ edge and leads to the root cycle.

For the TBox $\mathcal{T}'$, the second tuple in this path will have $\{B_1 \sqcap B_2, B_1, B_2\}$ as xcon_t (we do not need the selected successor to infer B_2 here), and the first tuple above will contain A_0 in xcon_t and will therefore not be a root tuple. Indeed, for $\mathcal{T}'$ the resulting set Ω_Σ does not contain a root cycle.

As suggested by Theorem 7, our algorithm first checks whether Ω_Σ contains a root cycle and, if so, returns 'not FO-rewritable'. Otherwise, it constructs a datalog program $\Pi_{A_0,\mathcal{T}}$ that is a rewriting of A_0 under $\mathcal{T}$ and non-recursive iff A_0 is FO-rewritable under $\mathcal{T}$ and Σ (which is guaranteed at this point by Theorem 7).

The IDB relations of $\Pi_{A_0,\mathcal{T}}$ take the form $P_{C,\mathsf{con},\mathsf{XCON}}$ where $C \in \mathsf{sub}(\mathcal{T})$, con is a subset of $\mathsf{sub}(\mathcal{T})$ and XCON is a set of such subsets or the special set $\{-\}$. We start with rules

$$P_{C_t,\mathsf{con}_t,\{-\}}(x) \leftarrow \bigwedge_{A \in S_t} A(x)$$

for all $t \in \Omega$ with $S_t \subseteq \mathsf{N_C}$ (which are of the form $(C_t, S_t, \mathsf{con}_t, -, -)$) and then exhaustively add a rule

$$P_{C_t,\mathsf{con}_t,\mathsf{XCON}}(x) \leftarrow \bigwedge_{A \in S_t \cap \mathsf{N_C}} A(x) \wedge \bigwedge_{i \leq n} \big(s_i(x, y_i) \wedge P_{C_{t_i},\mathsf{con}_{t_i},\mathsf{XCON}_i}(y_i) \big)$$

for all $t \in \Omega$ with existential restrictions $\exists r_0.D_0, \dots, \exists r_n.D_n$ the existential restrictions in S_t, tuples $t_0, \dots, t_n \in \Omega$, role names $s_0, \dots, s_n \in \Sigma$, $\ell \in \{0, \dots, n\}$, and sets $\mathsf{XCON}_0, \dots, \mathsf{XCON}_n$ such that

1. Conditions 1 to 5 from the rule (r_Ω) hold;
2. $P_{C_{t_i},\mathsf{con}_{t_i},\mathsf{XCON}_i}$ already occurs in $\Pi_{A_0,\mathcal{T}}$, for all $i \leq n$;

3. XCON consists of all sets $\mathsf{con}_{\mathcal{T}}(M')$ such that there is an $\ell' \leq n$ and an $\mathsf{xcon} \in \mathsf{XCON}_{\ell'}$ with

$$M' = (S_t \cap \mathsf{N_C}) \cup \{\exists s_{\ell'}. \bigsqcap \mathsf{xcon}\} \cup \{\exists s_i. \bigsqcap \mathsf{con}_{t_i} \mid \ell' \neq i \leq n\}.$$

In Point 3 above, we again identify $\exists r.-$ with $\top$. The goal predicates of $\Pi_{A_0,\mathcal{T}}$ are the predicates of the form $P_{A_0,\mathsf{con},\mathsf{XCON}}$ with $A_0 \in \mathsf{con}$ and $A_0 \notin \mathsf{xcon}$ for all $\mathsf{xcon} \in \mathsf{XCON}$. To eliminate 'accidental' recursiveness from $\Pi_{A_0,\mathcal{T}}$, remove all rules that contain a predicate which is not reachable from a goal predicate (defined in the obvious way).

Theorem 8.

1. *The program $\Pi_{A_0,\mathcal{T}}$ is a rewriting of A_0 under $\mathcal{T}$.*
2. *If A_0 is FO-rewritable under $\mathcal{T}$ and Σ, then $\Pi_{A_0,\mathcal{T}}$ is non-recursive.*

Note that $\Pi_{A_0,\mathcal{T}}$ is of double exponential size in the worst case. It is possible to find a (monadic) program that is only single exponential in the worst case, but unlike the program $\Pi_{A_0,\mathcal{T}}$ it is also best-case exponential. This cannot be significantly improved without giving up monadicity.

Theorem 9. *If A_0 is FO-rewritable under $\mathcal{T}$ and Σ, then it has a monadic non-recursive datalog rewriting of size at most $2^{p(n)}$, n the size of $\mathcal{T}$ and $p()$ a polynomial.*

There is a family of TBoxes $\mathcal{T}_1, \mathcal{T}_2, \ldots$ such that for all $n \geq 1$, $\mathcal{T}_n$ is of size $\mathcal{O}(n^2)$, the concept name A_0 is FO-rewritable under $\mathcal{T}_n$, and the smallest non-recursive monadic datalog rewriting has size at least 2^n.

Let us briefly analyze the complexity of the decomposed algorithm. It is easy to verify that the number of Γ-pairs and Ω-tuples is singly exponential in the size of $\mathcal{T}$ and that all required operations for building Γ and Ω and for determining the existence of a root cycle require only polynomial time. By Theorem 7, we have thus found an ExpTime algorithm for deciding FO-rewritability of concept queries under $\mathcal{EL}$-TBoxes and ABox signatures, which is optimal [Bienvenu *et al.*, 2013].

5 Experiments

We have implemented the decomposed algorithm in the *Grind* system and conducted a number of experiments. The system can be downloaded from http://www.informatik.uni-bremen.de/∼hansen/grind, and is released under GPL. In the following, we point out some selected aspects of the implementation. We use numbers to represent subconcepts in $\mathcal{T}$, store the S-component of each pair $(C, S) \in \Gamma$, which is a set of subconcepts of $\mathcal{T}$, as an ordered set, and use *tries* as a data structure to store Γ. We remove pairs $(C, S) \in \Gamma$ where S is not minimal, that is, for which there is a $(C, S') \in \Gamma$ with $S' \subsetneq S$. It can be verified that this optimization does not compromise correctness. We eagerly check for cycles in Γ without waiting for Phase 1 to complete and immediate start Phase 2 once a cycle is found to check whether it gives raise to non-FO-rewritability. If this is not the case, we return to Phase 1.

The experiments were carried out on a Linux (3.2.0) machine with a 3.5 GHz quad-core processor and 8 GB of RAM.

TBox	CI	CN	RN	no	stop	time	RQ stop	RQ time
ENVO	1942	1558	7	7	100%	2s	92.6%	2m52s
FBbi	567	517	1	0	100%	3s	86.1%	19m25s
MOHSE	3665	2203	71	1	99.6%	6m35s	58.7%	7h17m
NBO	1468	962	8	6	100%	3s	61.5%	3h05m
not-galen	4636	2748	159	44	95.9%	1h15m	48.9%	11h43m
SO	3160	2095	12	15	99.8%	4m28s	77.9%	3h53m
XP	1046	906	27	1	100%	27s	0.0%	7h33m

Table 1: TBoxes used in the experiments.

Although a large number of $\mathcal{ELH}^{dr}$-TBoxes is available on the web and from various repositories, most of them are *acyclic TBoxes* in the traditional DL sense, that is, the left-hand sides of all CIs are concept names, there are no two CIs with the same left-hand side, and there are no syntactic cycles. Since concept queries are always FO-rewritable under such TBoxes [Bienvenu *et al.*, 2012a], they are not useful for our experiments. We have identified seven TBoxes that do not fall into this class, listed in Table 1 together with the number of concept inclusions (CI), concept names (CN), and role names (RN) that they contain. Role inclusions occur in MOHSE, not-galen, and SO. All TBoxes together with information about their origin are available at http://www.informatik.uni-bremen.de/∼hansen/grind. All experiments conducted were using the full ABox signature (which in practice is the most difficult case since smaller signatures result in fewer node pairs and node tuples).

For each of these TBoxes, we have applied the decomposed algorithm to every concept name in the TBox. In some rare cases, either the set Γ has reached excessive size or calculation of Ω took too long, resulting in non-termination. We have thus established a 30 second timeout which results in termination in almost all cases; the "stop" column of Table 1 shows the fraction of inputs on which the algorithm successfully terminated, and the "time" column lists the overall runtime needed to process all concept names from the ontology (including timeouts). The generated non-recursive datalog-rewritings are typically of very reasonable size. The number of rules in the rewriting is displayed in the upper part of Figure 2; for example, for NBO, about 55% of all rewritings consist of a single rule, about 18% have two or three rules, about 10% have 4–7 rules, and so on. Note that the x-axis has exponential scale. The size of the rule bodies is typically very small, between one and two atoms in the vast majority of cases, and we have never encountered a rule with more than ten body atoms. It is also interesting to consider the number of IDB predicates in a rewriting, as intuitively these correspond to views that have to be generated by a database system that executes the query. As shown in the lower part of Figure 2, this number is rather small, and considerably lower than the number of rules in the produced programs (we again use exponential scale on the x-axis).

The experiments also confirm our initial belief that ontologies which are used in practical applications have a simple structure. As shown in the "no" column of Table 1, the number of concept names that are not FO-rewritable is extremely small. Moreover, if a concept name was FO-rewritable, then we were always able to find a rewriting already in Phase 1 of

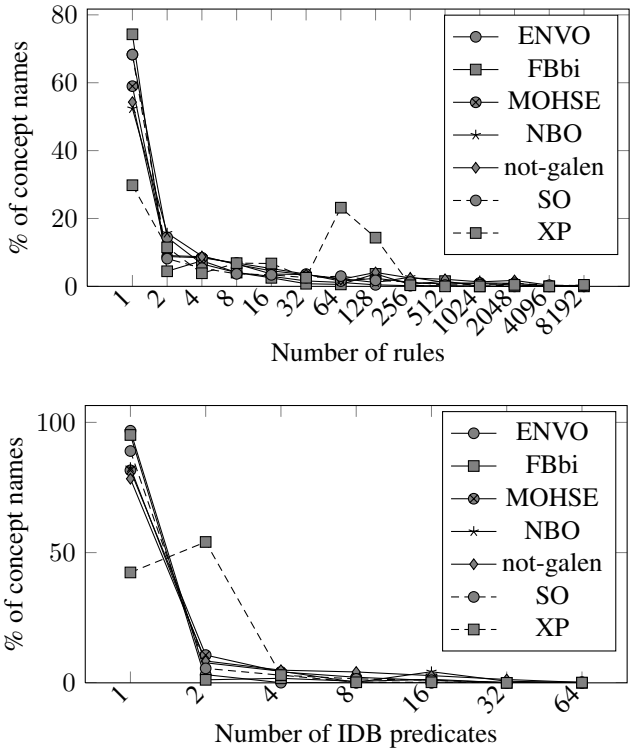

Figure 2: Number of rules and IDB predicates in the rewriting

our algorithm. Note, though, that for those cases that turned out to be not FO-rewritable, we had to go through Phases 2 and 3.

We have also compared the performance of Grind with that of REQUIEM, which implements an incomplete resolution-based approach to computing FO-rewritings for ontology languages up to $\mathcal{ELHI}$ [Pérez-Urbina *et al.*, 2010]. We use the same 30s timeout, which resulted in the termination rate and overall runtime (again including timeouts) displayed in the columns of Table 1 marked with "RQ". The termination cases correspond to positive answers, as REQUIEM cannot determine that an input is not FO-rewritable. Since REQUIEM tends to terminate only on relatively simple inputs, the constructed rewritings of the two tools are often identical. When they are not, either tool may produce a shorter rewriting.

6 Outlook

As future work, we plan to extend the algorithm and implementation from concept queries to conjunctive queries and to more expressive Horn-DLs such as $\mathcal{ELHI}$ and Horn-$\mathcal{SHIQ}$. This is non-trivial for several reasons; for example, with inverse roles the computation of the con and xcon sets in Phase 2 is no longer as straightforward as for $\mathcal{ELH}^{dr}$.

References

[Baader *et al.*, 2005] F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In *IJCAI*, pages 364–369, 2005.

[Baader *et al.*, 2007] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, and P.F. Patel-Schneider. *The Description Logic Handbook*. Cambridge University Press, 2007.

[Baget *et al.*, 2011] J.-F. Baget, M. Leclère, M.-L. Mugnier, and E. Salvat. On rules with existential variables: Walking the decidability line. In *AI*, 175(9-10):1620–1654, 2011.

[Bienvenu *et al.*, 2012a] M. Bienvenu, C. Lutz, and F. Wolter. Deciding FO-rewritability in $\mathcal{EL}$. In *DL*, pages 70–80, 2012.

[Bienvenu *et al.*, 2012b] M. Bienvenu, C. Lutz, and F. Wolter. Query containment in description logics reconsidered. In *KR*, pages 221–231, 2012.

[Bienvenu *et al.*, 2013] M. Bienvenu, C. Lutz, and F. Wolter. First order-rewritability of atomic queries in Horn description logics. In *IJCAI*, pages 754–760, 2013.

[Bienvenu *et al.*, 2014] M. Bienvenu, B. ten Cate, C. Lutz, and F. Wolter. Ontology-based data access: a study through Disjunctive Datalog, CSP, and MMSNP. In *TODS*, 39, 2014.

[Eiter *et al.*, 2012] T. Eiter, M. Ortiz, M. Simkus, T.-K. Tran, G. Xiao. Query rewriting for Horn-$\mathcal{SHIQ}$ plus rules. In *AAAI*, 2012.

[Kaminski and Grau, 2013] M. Kaminski and B. Cuenca Grau. Sufficient conditions for first-order and datalog rewritability in $\mathcal{ELU}$. In *DL*, pages 271–293, 2013.

[König *et al.*, 2012] M. König, M. Leclère, M.-L. Mugnier, and M. Thomazo. A sound and complete backward chaining algorithm for existential rules. In *RR*, pages 122–138, 2012.

[Kontchakov *et al.*, 2013] R. Kontchakov, M. Rodriguez-Muro, and M. Zakharyaschev. Ontology-based data access with databases: A short course. In *Reasoning Web*, pages 194–229, 2013.

[Lutz *et al.*, 2009] C. Lutz, D. Toman, and F. Wolter. Conjunctive query answering in the description logic $\mathcal{EL}$ using a relational database system. In *IJCAI*, pages 2070–2075, 2009.

[Lutz *et al.*, 2013] C. Lutz, İ. Seylan, D. Toman, and F. Wolter. The combined approach to OBDA: Taming role hierarchies using filters. In *ISWC*, pages 314–330, 2013.

[Mora and Corcho, 2013] J. Mora and Ó. Corcho. Engineering optimisations in query rewriting for OBDA. In *I-SEMANTICS*, pages 41–48, 2013.

[Pérez-Urbina *et al.*, 2010] H. Pérez-Urbina, B. Motik, and I. Horrocks. Tractable query answering and rewriting under description logic constraints. In *J. of Applied Logic*, 8(2):186–209, 2010.

[Poggi *et al.*, 2008] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Linking data to ontologies. In *J. on Data Semantics*, 10:133–173, 2008.

[Rosati and Almatelli, 2010] R. Rosati and A. Almatelli. Improving query answering over DL-Lite ontologies. In *KR*, pages 290–300, 2010.

[Rosati, 2007] R. Rosati. On conjunctive query answering in $\mathcal{EL}$. In *DL*, pages 451–458, 2007.

Appendix

A Proofs for Section 2

Lemma 1. *For every $\mathcal{ELH}^{dr}$-TBox $\mathcal{T}$, signature Σ, and concept query $A_0(x)$, one can construct in polynomial time an $\mathcal{ELH}$-TBox $\mathcal{T}'$ and signature Σ' such that $A_0(x)$ is FO-rewritable under $\mathcal{T}$ and Σ iff it is FO-rewritable under $\mathcal{T}'$ and Σ'.*

Moreover, every UCQ and non-recursive datalog rewriting of $A_0(x)$ under $\mathcal{T}'$ and Σ' can be converted in linear time into a UCQ and, respectively, non-recursive datalog rewriting of A_0 under $\mathcal{T}$ and Σ.

Proof. We recall the definition of $\mathcal{T}'$ and Σ'. Assume w.l.o.g. that $\mathcal{T}$ contains exactly one range restriction per role and no domain restrictions. Construct an $\mathcal{ELH}$-TBox $\mathcal{T}'$ as follows:

- replace every $\mathsf{ran}(r) \sqsubseteq C_r \in \mathcal{T}$ by $A_r \sqsubseteq C_r$ with A_r a fresh concept name;
- add $A_r \sqsubseteq A_s$ for every $r \sqsubseteq s \in \mathcal{T}$;
- replace every subconcept $\exists r.C$ in $\mathcal{T}$ with $\exists r.(A_r \sqcap C)$.

Let Σ' be Σ extended with all fresh concept names A_r.

We begin with the 'if' direction. Recall that FO-rewritability, UCQ-rewritability, and non-recursive datalog rewritability coincide. Thus, it is sufficient to show that one can convert in linear time any UCQ and non-recursive datalog rewriting of $A_0(x)$ under $\mathcal{T}'$ and Σ' into a UCQ and, respectively, non-recursive datalog rewriting of $A_0(x)$ under $\mathcal{T}$ and Σ.

Let Π' be a non-recursive datalog rewriting of $A_0(x)$ under $\mathcal{T}'$ and Σ' (the case of UCQ-rewritings is similar and omitted). Note that all symbols in Σ' are EDBs in Π' and so do not occur in the head of any rule in Π'. Construct Π from Π' by replacing in all rules of Π' all $A_r(x)$ with $r \in \Sigma$ by $r(y, x)$ where y is a fresh variable. We show that Π is a non-recursive datalog rewriting of $A_0(x)$ under $\mathcal{T}$ and Σ, that is, for all Σ-ABoxes $\mathcal{A}$ and individuals $a \in \mathsf{ind}(\mathcal{A})$, $a \in \Pi(\mathcal{A}) \Leftrightarrow \mathcal{A}, \mathcal{T} \models A_0(a)$. For any Σ-ABox $\mathcal{A}$, we can construct a Σ'-ABox $\mathcal{A}'$ by adding $A_r(a)$ for each assertion $r(b, a)$ in $\mathcal{A}$. Below, we show that for all Σ-ABoxes $\mathcal{A}$, the following holds:

1. $a \in \Pi(\mathcal{A})$ iff $a \in \Pi'(\mathcal{A}')$ and
2. $\mathcal{A}, \mathcal{T} \models A_0(a)$ iff $\mathcal{A}', \mathcal{T}' \models A_0(a)$.

This yields the desired result since we obtain $a \in \Pi(\mathcal{A})$ iff $a \in \Pi'(\mathcal{A}')$ (by Point 1) iff $\mathcal{A}', \mathcal{T}' \models A_0(a)$ (since Π' is a rewriting) iff $\mathcal{A}, \mathcal{T} \models A_0(a)$ (by Point 2).

In fact, Point 1 is an immediate consequence of the construction of Π and $\mathcal{A}'$. We thus concentrate on Point 2. First assume that $\mathcal{A}, \mathcal{T} \not\models A_0(a)$. Then there is a model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$ such that $a \notin A_0^{\mathcal{I}}$. Let $\mathcal{I}'$ be obtained from $\mathcal{I}$ by setting $A_r^{\mathcal{I}'} = \{d' \in \Delta^{\mathcal{I}} \mid \exists d' \ (d, d') \in r^{\mathcal{I}}\}$ for all $A_r \in \Sigma'$. It is easy to verify that $\mathcal{I}'$ is a model of $\mathcal{A}'$ and $\mathcal{T}'$. Consequently, $\mathcal{A}', \mathcal{T}' \not\models A_0(a)$. Conversely, assume that $\mathcal{A}', \mathcal{T}' \not\models A_0(a)$ and let $\mathcal{I}'$ be a model of $\mathcal{A}'$ and $\mathcal{T}'$ such that $a \notin A_0^{\mathcal{I}'}$. Let $\mathcal{I}$ be obtained from $\mathcal{I}'$ by setting $r^{\mathcal{I}} = r^{\mathcal{I}'} \cap (\Delta^{\mathcal{I}'} \times A_r^{\mathcal{I}'})$ for all $A_r \in \Sigma'$. It can be verified that $\mathcal{I}$ is a model of $\mathcal{A}$, in particular $r(b, c) \in \mathcal{A}$ implies $A_r(c) \in \mathcal{A}'$, and thus $(b, c) \in r^{\mathcal{I}}$. By construction of $\mathcal{I}$, we have $C^{\mathcal{I}} = C'^{\mathcal{I}'}$ whenever C is an $\mathcal{EL}$-concept and C' is obtained from C by replacing every subconcept $\exists r.D$ with $\exists r.(A_r \sqcap D)$ whenever $A_r \in \Sigma'$. Consequently and since $\mathcal{I}'$ is a model of $\mathcal{T}'$, $\mathcal{I}$ is a model of $\mathcal{T}$.

For the 'only if' direction, assume that $A_0(x)$ is FO-rewritable under $\mathcal{T}$ and Σ. Then there exists a UCQ $\varphi(x)$ that is a rewriting of $A_0(x)$ under $\mathcal{T}$ and Σ. We can assume that $\varphi(x)$ uses symbols from Σ only. From $\varphi(x)$, construct $\varphi'(x)$ by doing the following for all disjuncts $\psi(x)$ of $\varphi(x)$:

(a) for all $r(y, z) \in \psi(x)$ add $\bigvee_{\mathcal{T} \models s \sqsubseteq r, s \in \Sigma} A_s(z)$;

(b) delete all atoms $r(y, z) \in \psi(x)$ where $y \neq x$ and y does not appear in any other atom.

We now show that $\varphi'(x)$ is an FO-rewriting of A_0 under $\mathcal{T}'$ and Σ', that is, for all Σ'-ABoxes $\mathcal{A}'$ and individuals $a \in \mathsf{ind}(\mathcal{A}')$, $\mathcal{A}' \models \varphi'[a] \Leftrightarrow \mathcal{A}', \mathcal{T}' \models A_0(a)$. For any Σ'-ABox $\mathcal{A}'$, we construct a Σ-ABox $\mathcal{A}$ as follows:

(c) remove all assertions $r(b, c)$ such that $A_s(c) \notin \mathcal{A}'$ whenever $\mathcal{T} \models s \sqsubseteq r$;

(d) for all assertions $A_r(b)$, add $r(c, b)$ with c a fresh individual;

(e) delete all atoms $A_r(b)$.

Below, we show that for all Σ'-ABoxes $\mathcal{A}'$, the following holds:

1. $\mathcal{A}' \models \varphi'[a]$ iff $\mathcal{A} \models \varphi[a]$ and
2. $\mathcal{A}', \mathcal{T}' \models A_0(a)$ iff $\mathcal{A}, \mathcal{T} \models A_0(a)$.

This yields the desired result since we obtain $\mathcal{A}' \models \varphi'[a]$ iff $\mathcal{A} \models \varphi[a]$ (by Point 1) iff $\mathcal{A}, \mathcal{T} \models A_0(a)$ (since φ is a rewriting) iff $\mathcal{A}', \mathcal{T}' \models A_0(a)$ (by Point 2).

Point 1 holds by construction of φ' and of $\mathcal{A}$. In fact, a match of (some disjunct of) φ' in $\mathcal{A}'$ gives rise to a match of (the corresponding disjunct of) φ in $\mathcal{A}$ and, conversely, every match of φ in $\mathcal{A}$ is also a match of φ' in $\mathcal{A}'$. For the former, note that the removal of assertions in Step (c) of the construction of $\mathcal{A}$ is safe since every atom $r(x, y)$ in φ' is 'guarded' by an atom $A_r(y)$. Also, the original match needs to be extended to cover the atoms deleted in Step (b) of the construction of φ' which is possible by Step (d) of the construction of $\mathcal{A}$. For the latter, note that for any match of φ in $\mathcal{A}$, the match also satisfies the disjunctions added in Step (a) of the construction of φ' because there is an accompanying atom $r(x, y)$ and Step (c) of the construction of $\mathcal{A}$.

For Point 2, we start with the 'only if' direction. Thus assume that $\mathcal{A}, \mathcal{T} \not\models A_0(a)$. Then there is a model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$ such that $a \notin A_0^{\mathcal{I}}$. Assume w.l.o.g. that (†) $(d, c) \in r^{\mathcal{I}}$ with $c \in \mathsf{ind}(\mathcal{A})$ implies $d = b$ for some $b \in \mathsf{ind}(\mathcal{A})$ and $s(b, c) \in \mathcal{A}$ for some s with $\mathcal{T} \models s \sqsubseteq r$. Construct an interpretation $\mathcal{I}'$ by starting with $\mathcal{I}$ and then doing the following:

(f) set $A_r^{\mathcal{I}'} = \{d \mid (e, d) \in r^{\mathcal{I}}\}$;

(g) add (b, c) to the extension of r whenever $s(b, c) \in \mathcal{A}'$ and $\mathcal{T} \models s \sqsubseteq r$.

We verify that $\mathcal{I}'$ is a model of $\mathcal{A}'$ and $\mathcal{T}'$, which shows that $\mathcal{A}', \mathcal{T}' \not\models A_0(a)$. The assertions in $\mathcal{A}'$ removed in Step (c)

of the construction of $\mathcal{A}$ are satisfied by (g) and the assertions removed in Step (e) are satisfied because (d) ensures that $A_r(b) \in \mathcal{A}'$ implies $r(c,b) \in \mathcal{A}$ for some c and by (f). The role hierarchy statements in $\mathcal{T}'$ as well as the CIs $A_r \sqsubseteq C_r$ and $A_r \sqsubseteq A_s$ are satisfied by construction of $\mathcal{I}'$ and all remaining CIs (obtained from CIs in $\mathcal{T}$ by adding the concepts A_r inside existential restrictions) are satisfied because for each role name r, $r^{\mathcal{I}} = r^{\mathcal{I}'} \cap (\Delta^{\mathcal{I}'} \times A_r^{\mathcal{I}'})$. To see this, note that when an edge (b,c) is a added to $r^{\mathcal{I}'}$ in (g), then we cannot have $c \in A_r^{\mathcal{I}'}$; if we had, by (f) and (†) there would be an $s(d,c) \in \mathcal{A}$ with $\mathcal{T} \models s \sqsubseteq r$. Since $s(d,c)$ was not removed in (c), we have $A_t(c) \in \mathcal{A}'$ for some t with $\mathcal{T} \models t \sqsubseteq s$. Consequently, $s(b,c) \in \mathcal{A}'$ was also not removed in (c) and we have $(b,c) \in s^{\mathcal{I}}$, in contradiction to this edge being added in (g).

For the 'if' direction of Point 2, assume that $\mathcal{A}', \mathcal{T}' \not\models A_0(a)$ and thus there is a model $\mathcal{I}'$ of $\mathcal{A}'$ and $\mathcal{T}'$ such that $a \notin A_0^{\mathcal{I}}$. By the finite model property of $\mathcal{ALCH}$ [Baader *et al.*, 2007], we can assume $\mathcal{I}'$ to be finite. Let $\mathcal{A}_{\mathcal{I}'}$ be $\mathcal{I}'$ viewed as an ABox in the obvious way. Construct a model $\mathcal{I}$ by starting with $\mathcal{I}'$ and doing the following:

(h) for each assertion $r(c,b)$ added to $\mathcal{A}$ in Step (d), choose a model $\mathcal{J}$ of $\mathcal{A}_{\mathcal{I}'} \cup \{r(c,b)\}$ and $\mathcal{T}$ (which exists since any ABox is satisfiable under any TBox); by renaming elements, we can assume w.l.o.g. that $c \in \Delta^{\mathcal{J}}$ and that $\Delta^{\mathcal{J}}$ is disjoint from the model constructed thus far (which contains the element b since $\mathcal{I}'$ does); then take the disjoint union of the model constructed thus far with $\mathcal{J}$ and further add (c,b) to the extension of s whenever $\mathcal{T} \models r \sqsubseteq s$.

We show that $\mathcal{I}$ is a model of $\mathcal{A}$ and $\mathcal{T}$, which proves $\mathcal{A}, \mathcal{T} \not\models A_0(a)$. $\mathcal{I}$ satisfies all assertions in $\mathcal{A} \cap \mathcal{A}'$ since $\mathcal{I}'$ does. By construction, it clearly also satisfies all assertions added in Step (d) of the construction of $\mathcal{A}$. $\mathcal{I}$ satisfies all role hierarchy statements in $\mathcal{T}$ since $\mathcal{I}'$ and all of the models of $\mathcal{T}$ added in (h) do, and the additional edges added in (h) respect role hierarchy statements. It thus remains to prove that all CIs $C \sqsubseteq D$ in $\mathcal{T}$ are satisfied. Let $d \in C^{\mathcal{I}}$. First note that, by construction of $\mathcal{I}$, we have $d \in E^{\mathcal{I}}$ iff $d \in E^{\mathcal{I}'}$ for all $\mathcal{EL}$-concepts E and all $d \in \Delta^{\mathcal{I}'}$. Consequently, $d \in C^{\mathcal{I}} \cap \Delta^{\mathcal{I}'}$ implies $d \in D^{\mathcal{I}}$. Now let $d \in C^{\mathcal{I}}$ with $d \in \Delta^{\mathcal{J}}$, $\mathcal{J}$ a model added for an assertion $r(c,b)$ in (h). By construction of $\mathcal{I}$ and choice of $\mathcal{J}$, it follows that $d \in C^{\mathcal{J}}$. Consequently $d \in D^{\mathcal{J}}$, implying $d \in D^{\mathcal{I}}$ as required. ❑

B Proofs for Section 3

To prove Theorem 4, we first prove soundness of the algorithm in Figure 1, then its completeness, and finally termination for all inputs.

For the proof of soundness, we remind the reader of the standard chase procedure. The chase is a forward chaining procedure that exhaustively applies the CIs 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 materialised. To describe the procedure in detail and in the following proof 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 CI $C \sqsubseteq D \in \mathcal{T}$ 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 $\mathcal{A} := \mathcal{A} \cup \mathcal{A}_D$;
- an RI $r \sqsubseteq s \in \mathcal{T}$ and individuals $a, b \in \mathsf{ind}(\mathcal{A})$ such that $\mathcal{A} \models r(a,b)$ and then adding $s(a,b)$ to $\mathcal{A}$.

The *result of chasing $\mathcal{A}$ with $\mathcal{T}$*, denoted with $\mathsf{chase}_{\mathcal{T}}(\mathcal{A})$, is the ABox obtained by exhaustively applying chase steps to $\mathcal{A}$ in a fair way. It is standard to show that for all $\mathcal{EL}$-concepts C, we have $\mathcal{A}, \mathcal{T} \models C(a)$ iff $\mathsf{chase}_{\mathcal{T}}(\mathcal{A}) \models C(a)$.

Theorem 10 (Soundness)**.** *If the algorithm in Figure 1 returns $\bigvee M$, then $\bigvee M$ is an FO-rewriting of A_0 under $\mathcal{T}$ and the full signature.*

Proof. Let $\mathcal{A}$ be an ABox. We have to show:

1. if $\mathcal{A} \models \bigvee M(a_0)$, then $\mathcal{A}, \mathcal{T} \models A_0(a_0)$;
2. if $\mathcal{A}, \mathcal{T} \models A_0(a_0)$, then $\mathcal{A} \models \bigvee M(a_0)$.

For Point 1, assume that $\mathcal{A} \models \bigvee M(a_0)$. Then there is a $C \in M$ with $\mathcal{A} \models C(a_0)$. Consequently $\mathcal{A}, \mathcal{T} \models C(a_0)$. By construction of M, all its elements C satisfy $\mathcal{T} \models C \sqsubseteq A_0$, thus $\mathcal{A}, \mathcal{T} \models A_0(a_0)$ as required.

For Point 2, we essentially follow the proof strategy from [Baget *et al.*, 2011], based on the chase procedure. If $\mathcal{A}, \mathcal{T} \models A_0(a_0)$, then $A_0(a_0) \in \mathsf{chase}_{\mathcal{T}}(\mathcal{A})$ and consequently, there is a sequence of ABoxes $\mathcal{A} = \mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_k$ that *demonstrates* $A_0(a) \in \mathsf{chase}_{\mathcal{T}}(\mathcal{A})$, that is, each $\mathcal{A}_{i+1}$ is obtained from $\mathcal{A}_i$ by a single chase step and $A_0(a) \in \mathcal{A}_k$. It thus suffices to prove by induction on k that

(∗) if $\mathcal{A} = \mathcal{A}_0, \ldots, \mathcal{A}_k$ is a chase sequence that demonstrates $A_0(a_0) \in \mathsf{chase}_{\mathcal{T}}(\mathcal{A})$, then $\mathcal{A} \models \bigvee M(a_0)$.

The induction start is trivial: for $k = 0$, $A_0(a_0) \in \mathcal{A}_k$ implies $A_0(a_0) \in \mathcal{A}$. Since $A_0 \in M$, we have $\mathcal{A} \models \bigvee M(a_0)$. For the induction step, assume that $\mathcal{A} = \mathcal{A}_0, \ldots, \mathcal{A}_k$ is a chase sequence that demonstrates $A_0(a_0) \in \mathsf{chase}_{\mathcal{T}}(\mathcal{A})$, with $k > 0$. Applying IH to the subsequence $\mathcal{A}_1, \ldots, \mathcal{A}_k$, we obtain that $\mathcal{A}_1 \models \bigvee M(a_0)$. Thus there is a $C \in M$ with $\mathcal{A}_1 \models C(a_0)$. Next we look at the two possibilities for the chase step that led from $\mathcal{A}_0$ to $\mathcal{A}_1$.

(Application of a CI) Assume that $\mathcal{A}_1$ is obtained from $\mathcal{A}_0$ by choosing $E \sqsubseteq F \in \mathcal{T}$ 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 $\mathcal{A}_0$:

- $A(b)$ for each concept name A that occurs as a top-level conjunct in F;
- the sub-ABox $\mathcal{A}_{\exists r.G}$ rooted at b for each $\exists r.G$ that is a top-level conjunct of F.

We might or might not have $\mathcal{A}_0 \models C(a_0)$, depending on whether or not the truth of C at a_0 depends on the additions due to applying $E \sqsubseteq F$ as a (forward) rule at b. If $\mathcal{A}_0 \models C(a_0)$ does hold, then we are done. Otherwise, let h

be a homomorphism from C to $\mathcal{A}_1$ with $h(x_\varepsilon) = a_0$, and let $x_1, \ldots, x_n$ be all elements of $\mathsf{var}(C)$ such that $h(x_i) = b$ and there is at least one tlc G_i of $C|_{x_i}$ with $\models F \sqsubseteq G_i$. There must be at least one such x_i since, otherwise, h does not depend on any assertions added in the construction of $\mathcal{A}_1$ from $\mathcal{A}_0$ and thus witnesses $\mathcal{A}_0 \models C(a_0)$, a contradiction. Let the concepts $C_0, \ldots, C_n$ be such that $C_0 = C$ and C_{i+1} can be obtained from C_i by doing the following if $x_i \in \mathsf{var}(C_i)$ (otherwise, just set $C_{i+1} := C_i$):

1. remove $A(x_i)$ for all concept names A with $\models F \sqsubseteq A$;
2. remove the subtree rooted at y whenever $r(x_i, y) \in C$ and $\models F \sqsubseteq \exists r.(C|_y)$;
3. add $A(x_i)$ for all concept names A that are a top-level conjuncts of E;
4. add the subtree $\exists r.H$ to x_i for each $\exists r.H$ that is a top-level conjunct of E;
5. minimise the resulting C_i', that is, choose $C_{i+1} \prec^* C_i'$ such that C_{i+1} is $\prec$-minimal with $\mathcal{T} \models C_{i+1} \sqsubseteq A_0$.

It is easy to prove by induction on i that $C_i \in M$ for all $i \leq n$.

It thus remains to argue that $\mathcal{A}_0 \models C_n(a_0)$. To do this, we produce maps $h_0, \ldots, h_n$ such that h_i is a homomorphism from C_i to $\mathcal{A}_1$ with $h_i(x_\varepsilon) = a_0$ and such that $h_i(x_j) = b$ if $x_j \in \mathsf{var}(C_i)$, 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 C_i after the removals in Steps 1 and 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 $h_i(x_i) = b$, 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 $\mathsf{var}(C_{i+1})$.

By construction of the concepts $C_0, \ldots, C_n$ and the homomorphisms $h_0, \ldots, h_n$, there is no atom in C_n such that the image of the atom under h_n was added by applying $E \sqsubseteq F$ as a (forward) rule at b. To show this, assume to the contrary that there is such an atom $A(x)$ in C_n. There are two cases:

1. $h(x) = b$.
   Then $x = x_i$ for some i. Since $A(h(x)) = A(b)$ was added by the application of $E \sqsubseteq F$, A is a top-level conjunct of F. Consequently, $A(x)$ was removed in Step 1 when constructing C_{i+1} from C_i, in contradiction to $A(x)$ being in C_n.
2. $h(x) \neq b$.
   Then $h(x)$ is a non-root node of the sub-ABox $\mathcal{A}_{\exists r.G}$ of $\mathcal{A}_1$, where $\exists r.G$ is a top-level conjunct of F. Consider the unique path in C from x_ε to x, that is, the sequence of individuals $y_0, \ldots, y_\ell$ with $y_0 = x_\varepsilon$ and $y_\ell = x$ such that $r_i(y_i, y_{i+1}) \in C$ 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 $\mathcal{A}_1$ is tree-shaped, b must be on that second path. Let y_p be such that $h(y_p) = b$. We must have $y_p = x_i$ for some i. Then $r_p = r$ and $\mathcal{A}_{\exists r.G} \models \exists r.(C|_{y_{p+1}})(b)$. Hence $\models F \sqsubseteq \exists r.(C|_{y_{p+1}})$ since $\exists r.G$ is a top-level conjunct of F. Consequently, the subtree of C rooted at y_{p+1} was removed in Step 2 when constructing C_{i+1}, in contradiction to $A(x)$ being in C_n.

The case of role atoms is similar to subcase 2 above, but simpler. We have thus shown that there is no atom in C_n such that the image of this atom under h_n was added by applying $E \sqsubseteq F$ as a (forward) rule at b. Consequently $\mathcal{A}_0 \models C_n(a_0)$ via h_n.

(Application of an RI) Assume that $\mathcal{A}_1$ is obtained from $\mathcal{A}_0$ by choosing $r \sqsubseteq s \in \mathcal{T}$ and $b, c \in \mathsf{ind}(\mathcal{A})$ with $\mathcal{A}_0 \models r(b, c)$, and adding $s(b, c)$ to $\mathcal{A}_0$. It might be that $\mathcal{A}_0 \models C(a_0)$, regardless of the application of the RI. In this case we are done. Otherwise, let h be a homomorphism from C to $\mathcal{A}_1$ with $h(x_\varepsilon) = a_0$, and let $s(x_1, x_1'), \ldots, s(x_n, x_n')$ be all atoms of C with $h(x_i) = b$ and $h(x_i') = c$. There must be at least one such $s(x_i, x_i')$, since otherwise h would not depend on any assertions added in the construction of $\mathcal{A}_1$ from $\mathcal{A}_0$ and thus witnesses $\mathcal{A}_0 \models C(a_0)$, a contradiction. Let the concepts $C_0, \ldots, C_n$ be such that $C_0 = C$, and C_{i+1} can be obtained from C_i by doing the following if $s(x_i, x_i') \in C_i$ (otherwise, just set $C_{i+1} := C_i$):

1. replace the atom $s(x_i, x_i')$ by $r(x_i, x_i')$;
2. minimise the resulting C_i', that is, choose $C_{i+1} \prec^* C_i'$ such that C_{i+1} is $\prec$-minimal with $\mathcal{T} \models C_{i+1} \sqsubseteq A_0$.

It is easy to prove by induction on i that $C_i \in M$ for all $i \leq n$. It thus remains to argue that $\mathcal{A}_0 \models C_n(a_0)$. To do this, we produce maps $h_0, \ldots, h_n$ such that h_i is a homomorphism from C_i to $\mathcal{A}_1$ with $h_i(x_\varepsilon) = a_0$ and such that $h_i(x_j) = b$ if $x_j \in \mathsf{var}(C_i)$ and $h_i(x_j') = c$ if $x_j' \in \mathsf{var}(C_i)$, for all $i \leq n$. Start with $h_0 = h$. To produce h_{i+1} from h_i and to deal with the minimization in Step 2, restrict h_i' to $\mathsf{var}(C_{i+1})$. This is possible since $\mathcal{A}_0$ (and therefore $\mathcal{A}_1$) $\models r(b, c)$, and $h_i(x_i) = b$ and $h_i(x_i') = c$ if $x_i, x_i' \in C_i$.

By construction of the concepts $C_0, \ldots, C_n$ and the homomorphisms $h_0, \ldots, h_n$, there is no atom $s(x, x') \in C_n$ such that the image of the atom under h_n was added by applying $r \sqsubseteq s$ as a (forward) rule at $r(b, c)$. Consequently $\mathcal{A}_0 \models C_n(a_0)$ via h_n. ❏

In order to show completeness of our algorithm, we need some notation. For a tree-shaped ABox $\mathcal{A}$ with root $\rho_\mathcal{A}$, let $\mathcal{A}|_k$ denote the restriction of $\mathcal{A}$ to individuals whose distance from $\rho_\mathcal{A}$ does not exceed k. In what follows we use the same notation for $\mathcal{EL}$-concepts C corresponding to tree-shaped ABoxes. The following can be proved in the same way as Theorem 7 and 9 in [Bienvenu *et al.*, 2013].[1]

Lemma 11. *Let $\mathcal{T}$ be an $\mathcal{ELH}$-TBox and Σ an ABox signature. The instance query $A_0(x)$ is not FO-rewritable under $\mathcal{T}$ and Σ iff for every $k > 0$, there exists a tree-shaped Σ-ABox $\mathcal{A}$ of depth exceeding k with root $\rho_\mathcal{A}$ such that $\mathcal{A}, \mathcal{T} \models A_0(\rho_\mathcal{A})$ and $\mathcal{A}|_k, \mathcal{T} \not\models A_0(\rho_\mathcal{A})$. This is still true if "for every $k > 0$" is replaced with "for $k = 2^{3n^2}$".*

We are in the position now to prove the completeness result.

Theorem 12 (Completeness)**.** *If the algorithm in Figure 1 returns 'not FO-rewritable', then A_0 has no FO-rewriting under $\mathcal{T}$ and the full signature.*

[1] The theorems in [Bienvenu *et al.*, 2013] do not cover role hierarchies; however, adding them to the proofs is straightforward.

Proof. It suffices to show that if the algorithm returns 'not FO-rewritable', then the condition of Lemma 11 holds. We use the $\mathcal{EL}$-concept corresponding to a tree-shaped ABox rather that the ABox itself to formulate the condition of Lemma 11, namely:

($*$) for every $k > 0$, there is a concept C whose depth exceeds k and such that $\mathcal{T} \models C \sqsubseteq A_0$ and $\mathcal{T} \not\models C|_k^- \sqsubseteq A_0$.

We show the following

Claim 1. If there is a concept C that is blocked with variables $x_1, x_2, x_3 \in \mathsf{var}(C)$, $\mathcal{T} \models C \sqsubseteq A_0$, and $\mathcal{T} \not\models C \backslash C|_{x_3} \sqsubseteq A_0$, then ($*$) holds.

If Claim 1 is proved, completeness of the algorithm follows: assume the algorithm returns 'not FO-rewritable'. Then there is a concept D that is $\prec$-minimal with $\mathcal{T} \models D \sqsubseteq A_0$ and that is blocked with variables x_1, x_2, x_3. By $\prec$-minimality of D, $\mathcal{T} \not\models (D \setminus D|_{x_3}) \sqsubseteq A_0$ and so ($*$) follows.

It thus remains to prove Claim 1. Consider a concept C that is blocked with variables $x_1, x_2, x_3 \in \mathsf{var}(C)$, $\mathcal{T} \models C \sqsubseteq A_0$, and $\mathcal{T} \not\models C \setminus C|_{x_3} \sqsubseteq A_0$ (otherwise it would not be in M). So the following conditions are satisfied:

1. x_1 is a proper ancestor of x_2, which is an ancestor of x_3,
2. $\mathsf{con}_{\mathcal{T}}^{C}(x_1) = \mathsf{con}_{\mathcal{T}}^{C}(x_2)$ and $\mathsf{con}_{\mathcal{T}}^{C\backslash C|_{x_3}}(x_1) = \mathsf{con}_{\mathcal{T}}^{C\backslash C|_{x_3}}(x_2)$.

Let G be obtained from C by replacing the subconcept $C|_{x_2}$ with a copy of $C|_{x_1}$ in which every variable x has been renamed to x'. It can be proved that

1. $\mathsf{con}_{\mathcal{T}}^{C}(x_1) = \mathsf{con}_{\mathcal{T}}^{G}(x_1') = \mathsf{con}_{\mathcal{T}}^{G}(x_1)$;
2. $\mathsf{con}_{\mathcal{T}}^{C\backslash C|_{x_3}}(x_1) = \mathsf{con}_{\mathcal{T}}^{G\backslash G|_{x_3'}}(x_1') = \mathsf{con}_{\mathcal{T}}^{G\backslash G|_{x_3'}}(x_1)$;
3. $\mathsf{con}_{\mathcal{T}}^{C}(x_\varepsilon) = \mathsf{con}_{\mathcal{T}}^{G}(x_\varepsilon)$;
4. $\mathsf{con}_{\mathcal{T}}^{C\backslash C|_{x_3}}(x_\varepsilon) = \mathsf{con}_{\mathcal{T}}^{G\backslash G|_{x_3'}}(x_\varepsilon)$.

In particular, we have that G is blocked with variables x_1, x_1', and x_3', $\mathcal{T} \models G \sqsubseteq A_0$ and $\mathcal{T} \not\models G \setminus G|_{x_3'} \sqsubseteq A_0$. Also note that the depth of x_3' in G is larger than the depth of x_3 in C. We now apply the same construction to G again until the copy of x_3 has depth exceeding k. For the resulting concept G' we have $\mathcal{T} \models G' \sqsubseteq A_0$ and $\mathcal{T} \not\models G'|_k^- \sqsubseteq A_0$, as required. ❑

It remains to prove termination of the algorithm in Figure 1. Let $\mathcal{T}$ be of size n. Every concept of depth larger than 2^{2n} must be blocked. For this reason, M can only contain concepts of depth at most 2^{2n}. Since every round of the while loop adds a fresh concept to M, it thus suffices to show that each concept in M has outdegree at most n. This is in fact a consequence of minimization. Assume that D' is $\prec$-minimal with $\mathcal{T} \models D' \sqsubseteq A_0$ and that, to the contrary of what we aim to show, there is some $x \in \mathsf{var}(D')$ that has successors $r_1(x, x_1), \ldots, r_{n+1}(x, x_{n+1})$. For each existential restriction $\exists r.C$ in $\mathsf{sub}(\mathcal{T})$ such that $\mathcal{T} \models D'|_x \sqsubseteq \exists r.C$, choose a successor x_i such that $r_i = r$ and $\mathcal{T} \models D'|_{x_i} \sqsubseteq C$. Let D'' be obtained from D' by dropping all subtrees rooted at nodes x_i that were not chosen in this way. Then we have $\mathcal{T} \models D'' \sqsubseteq A_0$. Since at least one x_i-rooted subtree was dropped in the construction of D'' from D', this contradicts the $\prec$-minimality of D'.

C Proofs for Section 4

We first observe that $\widehat{\Gamma}_\Sigma$ coincides with the set of pairs (C, D) in $\widehat{\Gamma}$ such that D only uses symbols from Σ, where $\widehat{\Gamma}$ is obtained from Γ in the same way as $\widehat{\Gamma}_\Sigma$ is obtained from Γ_Σ except that all conditions pertaining to the signature Σ are disregarded. Formally, define $\widehat{\Gamma}$ as the limit of the sequence of $\widehat{\Gamma}^0, \widehat{\Gamma}^1, \ldots$, where:

- $\widehat{\Gamma}^0 := \{(C, \bigsqcap S) \mid (C, S) \in \Gamma \text{ and } S \subseteq \mathsf{N_C}\}$.
- $\widehat{\Gamma}^{i+1}$ is $\widehat{\Gamma}^i$ extended with all pairs (C, D) such that there is $(C, S) \in \Gamma$ with the following property: for each $\exists r.G \in S$ there are $(G, C_{r,G}) \in \widehat{\Gamma}^i$ and $s_{r,G}$ such that $\mathcal{T} \models s_{r,G} \sqsubseteq r$ and $D = \bigsqcap_{A \in S \cap \mathsf{N_C}} A \sqcap \bigsqcap_{\exists r.G \in S} \exists s_{r,G}.C_{r,G}$.

Thus, Proposition 6 follows if for all ABoxes $\mathcal{A}$ and $a \in \mathsf{ind}(\mathcal{A})$, the following are equivalent:

- there is an $(A_0, D) \in \widehat{\Gamma}$ with $\mathcal{A} \models D(a)$,
- $\mathcal{A}, \mathcal{T} \models A_0(a)$.

We split the proof into a soundness and a completeness part.

Lemma 13 (Soundness of $\widehat{\Gamma}$)**.** *For all ABoxes $\mathcal{A}$ and $a \in \mathsf{ind}(\mathcal{A})$, if there is a $(A_0, D) \in \widehat{\Gamma}$ with $\mathcal{A} \models D(a)$, then $\mathcal{A}, \mathcal{T} \models A_0(a)$.*

Proof. We first show the following

Claim 1. For all $(C, S) \in \Gamma$ we have $\mathcal{T} \models \bigsqcap S \sqsubseteq C$.

Proof of Claim 1. Let $\Gamma_0, \ldots, \Gamma_k = \Gamma$ be the sets of pairs obtained by repeatedly applying rules (r1) and (r2) to the initial set $\Gamma_0 = \{(A_0, \{A_0\})\}$. We show by induction on i that for all $(C, S) \in \Gamma_i$ we have $\mathcal{T} \models \bigsqcap S \sqsubseteq C$. The induction start is trivial. For the inductive step, first suppose that Γ_{i+1} is obtained from Γ_i by applying Rule (r1). Then there are $(C, S) \in \Gamma_i$ and $D \sqsubseteq A \in \mathcal{T}$ with $A \in S$ such that Γ_{i+1} is Γ_i extended with the following tuples:

(a) $(C, (S \setminus \{A\}) \cup \mathsf{tlc}(D))$;

(b) for every $\exists r.G \in \mathsf{sub}(D)$, the pair $(G, \mathsf{tlc}(G))$.

The claim is trivially true for tuples added in (b). For the tuple in (a), let $S' = (S \setminus \{A\}) \cup \mathsf{tlc}(D)$. We have $\mathcal{T} \models \bigsqcap S' \sqsubseteq \bigsqcap S$ by $D \sqsubseteq A \in \mathcal{T}$. From this, the claim follows as a consequence of IH.

Now assume that Γ_{i+1} is obtained from Γ_i by applying Rule (r2). Then there are $(C, S) \in \Gamma_i$ and $D \sqsubseteq \exists r.F \in \mathcal{T}$ such that Γ_{i+1} is Γ_i extended with the following tuples:

(a) $(C, (S \setminus \{\exists s.G \mid \mathcal{T} \models F \sqsubseteq G \text{ and } \mathcal{T} \models r \sqsubseteq s\}) \cup \mathsf{tlc}(D)$;

(b) for every $\exists r.G \in \mathsf{sub}(D)$, the pair $(G, \mathsf{tlc}(G))$.

Again, the claim is trivially true for tuples added in (b). For the tuple in (a), let $S' = (S \setminus \{\exists s.G \mid \mathcal{T} \models F \sqsubseteq G \text{ and } \mathcal{T} \models r \sqsubseteq s\}) \cup \mathsf{tlc}(D)$. We have $\mathcal{T} \models \bigsqcap S' \sqsubseteq \bigsqcap S$ since for each $\exists s.G \in S$ with $\mathcal{T} \models F \sqsubseteq G$ and $\mathcal{T} \models r \sqsubseteq s$, it

holds that $\mathcal{T} \models D \sqsubseteq \exists s.G$. From this, the claim follows as a consequence of IH.

This finishes the proof of the claim.

Using Claim 1, one can now prove the corresponding result for $\widehat{\Gamma}$, which is as follows.

Claim 2. For all $(C, D) \in \widehat{\Gamma}$ we have $\mathcal{T} \models D \sqsubseteq C$.

The proof proceeds by induction on i to establish that for all $(C, D) \in \widehat{\Gamma}_i$, we have $\mathcal{T} \models D \sqsubseteq C$. Both the induction start and step are straightforward and essentially only require an appropriate invocation of Claim 1. Details are omitted.

Claim 2 directly implies Lemma 13: assume there is an $(A_0, D) \in \widehat{\Gamma}$ with $\mathcal{A} \models D(a)$. By Claim 2, $\mathcal{T} \models D \sqsubseteq A_0$. Thus $\mathcal{A}, \mathcal{T} \models A_0(a)$, as required. ❑

For the completeness part of Proposition 6, we require some preparation. First, for every $(C, D) \in \widehat{\Gamma}$ and variable $x \in \mathsf{var}(D)$, we denote by $\mu_{C,D}(x)$ the element of Γ used in the construction of D at variable x. Thus, for every $(C, S) \in \Gamma$ with $S \subseteq \mathsf{N_C}$ we set $\mu_{C,\sqcap S}(x_\varepsilon) = (C, S)$. Assume that $(C, D) \in \widehat{\Gamma}_{i+1}$ is constructed from some $(C, S) \in \Gamma$ and, for every $\exists r.G \in S$, some $(G, C_{r,G}) \in \widehat{\Gamma}_i$ and $s_{r,G}$ with $\mathcal{T} \models s_{r,G} \sqsubseteq r$ by putting $D = \bigsqcap (S \cap \mathsf{N_C}) \sqcap \bigsqcap_{\exists r.G \in S} \exists s_{r,G}.C_{r,G}$. Then we let $\mu_{C,D}(x_\varepsilon) = (C, S)$ and $\mu_{C,D}(x) = \mu_{G,C_{r,G}}(x)$ for $x \in \mathsf{var}(C_{r,G})$.

We also need a suitable version of the chase procedure that reflects our rules (r1) and (r2) for constructing Γ. This chase introduces new ABox elements that are called *nulls*. It applies the following three rules:

(Ch0) If $\mathcal{A} \models r(a, b)$, a, b are not nulls, $r \sqsubseteq s \in \mathcal{T}$, and $s(a, b) \notin \mathcal{A}$, then add $s(a, b)$ to $\mathcal{A}$;

(Ch1) If $\mathcal{A} \models C(a)$, a is not a null, $C \sqsubseteq A \in \mathcal{T}$, and A a concept name, and $A(a) \notin \mathcal{A}$, then add $A(a)$ to $\mathcal{A}$;

(Ch2) If $\mathcal{A} \models C(a)$, a is not a null, and $C \sqsubseteq \exists r.F \in \mathcal{T}$, and $\mathcal{A} \not\models \exists r.F(a)$, then add $\{s(a, b) \mid \mathcal{T} \models r \sqsubseteq s\}$ to $\mathcal{A}$ with b a fresh null; further add the sub-ABox $\mathcal{A}_E$ (using only fresh nulls) rooted at b for every $\exists s.E \in \mathsf{sub}(\mathcal{T})$ such that $\mathcal{T} \models F \sqsubseteq E$ and $\mathcal{T} \models r \sqsubseteq s$.

We set $A_0(a) \in \mathsf{chase}_\mathcal{T}(\mathcal{A})$ if there exists a sequence of ABoxes $\mathcal{A} = \mathcal{A}_0, \dots, \mathcal{A}_k$ such that each $\mathcal{A}_{i+1}$ is obtained from $\mathcal{A}_i$ by a single application of rules (Ch0), (Ch1) or (Ch2) and $A_0(a) \in \mathcal{A}_k$. We then say that the sequence $\mathcal{A}_0, \dots, \mathcal{A}_k$ *demonstrates* that $A_0(a) \in \mathsf{chase}_\mathcal{T}(\mathcal{A})$.

Lemma 14. *Let $\mathcal{T}$ be a TBox and $\mathcal{A}$ an ABox with $a \in \mathsf{ind}(\mathcal{A})$. Then $\mathcal{A}, \mathcal{T} \models A_0(a)$ iff $A_0(a) \in \mathsf{chase}_\mathcal{T}(\mathcal{A})$.*

We are now in the position to prove the completeness part of Proposition 6.

Lemma 15 (Completeness of $\widehat{\Gamma}$)**.** *For all ABoxes $\mathcal{A}$ and $a \in \mathsf{ind}(\mathcal{A})$, if $\mathcal{A}, \mathcal{T} \models A_0(a)$, then there is an $(A_0, D) \in \widehat{\Gamma}$ with $\mathcal{A} \models D(a)$.*

Proof. Assume that $\mathcal{A}, \mathcal{T} \models A_0(a)$. Then $A_0(a) \in \mathsf{chase}_\mathcal{T}(\mathcal{A})$ and consequently there is a sequence of ABoxes $\mathcal{A} = \mathcal{A}_0, \mathcal{A}_1, \dots, \mathcal{A}_k$ that demonstrates $A_0(a) \in \mathsf{chase}_\mathcal{T}(\mathcal{A})$. It thus suffices to prove by induction on k that

(∗) if $\mathcal{A} = \mathcal{A}_0, \dots, \mathcal{A}_k$ is a sequence that demonstrates $A_0(a) \in \mathsf{chase}_\mathcal{T}(\mathcal{A})$, then $\mathcal{A} \models D(a)$ for some $(A_0, D) \in \widehat{\Gamma}$.

The induction start is trivial: for $k = 0$, $A_0(a) \in \mathcal{A}_k$ implies $A_0(a) \in \mathcal{A}$, and clearly we have $(A_0, A_0) \in \widehat{\Gamma}$ since $(A_0, \{A_0\}) \in \Gamma$. For the induction step, assume that $\mathcal{A} = \mathcal{A}_0, \dots, \mathcal{A}_k$ is a chase sequence that demonstrates $A_0(a) \in \mathsf{chase}_\mathcal{T}(\mathcal{A})$, with $k > 0$. Applying IH to the subsequence $\mathcal{A}_1, \dots, \mathcal{A}_k$, we find an $(A_0, C) \in \widehat{\Gamma}$ with $\mathcal{A}_1 \models C(a)$. We consider each chase rule separately.

(Ch0) Assume first that $\mathcal{A}_1$ is obtained from $\mathcal{A}_0$ by rule (Ch0), i. e. by choosing $r \sqsubseteq s \in \mathcal{T}$, and individuals $a_0, a_1 \in \mathsf{ind}(\mathcal{A}_0)$ with $\mathcal{A}_0 \models r(a_0, a_1)$, and adding $s(a_0, a_1)$. Let h be a homomorphism from C to $\mathcal{A}_1$ with $h(x_\varepsilon) = a$, and let $(x_1, y_1), \dots, (x_k, y_k)$ be all pairs in $\mathsf{var}(C)$ such that $s(x_i, y_i) \in C$, $h(x_i) = a_0$, and $h(y_i) = a_1$. If this list of elements is empty, then $\mathcal{A}_0 \models C(a)$ and we are done. Otherwise, let the concept C' be obtained from C by replacing all $s(x_i, y_i)$ with $r(x_i, y_i)$. Clearly $\mathcal{A}_0 \models C'(a)$. We show that $(A_0, C') \in \widehat{\Gamma}$. Fix some (x_i, y_i) and consider the pair $\mu_{A_0,C}(x_i) = (F_i, S_i) \in \Gamma$. By construction of $\widehat{\Gamma}$ and since $s(x_i, y_i) \in C$, there must exist $\exists t.G \in S_i$ such that C was constructed using the role $s_{t,G} = s$ with $\mathcal{T} \models s_{t,G} \sqsubseteq t$ and using $(G, C_{t,G}) \in \widehat{\Gamma}$ such that $\mu_{A_0,C}(y_i) = \mu_{G,C_{t,G}}(y_i)$. We have $\mathcal{T} \models r \sqsubseteq t$. Thus, instead of using s (as $s_{t,G}$) in the construction of C one can use r (as $s_{t,G}$). By doing this for every $s(x_i, y_i) \in C$, we obtain that $(A_0, C') \in \widehat{\Gamma}$.

(Ch1) Assume first that $\mathcal{A}_1$ is obtained from $\mathcal{A}_0$ by rule (Ch1), i. e. by choosing a CI $D \sqsubseteq A \in \mathcal{T}$, A a concept name, and an individual $b \in \mathsf{ind}(\mathcal{A}_0)$ with $\mathcal{A}_0 \models D(b)$, and adding $A(b)$. Let h be a homomorphism from C to $\mathcal{A}_1$ with $h(x_\varepsilon) = a$, and let $x_1, \dots, x_k$ be all elements of $\mathsf{var}(C)$ such that $A(x_i) \in C$ and $h(x_i) = b$. If this list of elements is empty, then $\mathcal{A}_0 \models C(a)$ and we are done. Otherwise, let the concept C' be obtained from C by replacing A with the concept D at every x_i, that is:

1. remove $A(x_i)$;
2. add $B(x_i)$ for each concept name $B \in \mathsf{tlc}(D)$;
3. add the subconcept $\exists r.F$ rooted at x_i for each $\exists r.F$ that is a tlc in D.

We show that $(A_0, C') \in \widehat{\Gamma}$ and $\mathcal{A}_0 \models C'(a)$, starting with the former. Fix some x_i and consider the pair $\mu_{A_0,C}(x_i) = (F_i, S_i) \in \Gamma$. By construction of $\widehat{\Gamma}$ and since $A(x_i) \in C$, we must have $A \in S_i$. Therefore, Rule (r1) from the construction of Γ is applicable to (F_i, S_i) and yields the pair

$$p_i = (F_i, S_i \setminus \{A\} \cup \mathsf{tlc}(D)) \in \Gamma.$$

When building up the pair $(A_0, C) \in \widehat{\Gamma}$ and dealing with node x_i, we can use the pair p_i in place of (F_i, S_i). Using the fact that all pairs $(G, \mathsf{tlc}(G))$ with $\exists r.G \in \mathsf{sub}(D)$ have been added to Γ, we can then reconstruct D and obtain $(A_0, C') \in \widehat{\Gamma}$, as required.

Now we show $\mathcal{A}_0 \models C'(a)$. Since $\mathcal{A}_0 \models D(b)$, there is a homomorphism h' from D to $\mathcal{A}_0$ that maps x_ε to b. We obtain

a homomorphism from C' to $\mathcal{A}_0$ that maps x_ε to a by starting with the homomorphism h and 'plugging in' h' at every node x_i to cover the subtrees generated by the subconcepts $\exists r.F$ of D added in Step 3 above.

(Ch2) Assume now that $\mathcal{A}_1$ is obtained from $\mathcal{A}_0$ by an application of the rule (Ch2), i. e. by choosing a CI $D \sqsubseteq \exists r.F \in \mathcal{T}$ and an individual $b \in \mathsf{ind}(\mathcal{A}_0)$ with $\mathcal{A}_0 \models D(b)$, letting R be the set of all roles s with $\mathcal{T} \models r \sqsubseteq s$, and adding $s(b, c)$ for all $s \in R$ and the sub-ABox $\mathcal{A}_E$ rooted at c for every $\exists s.E \in \mathsf{sub}(\mathcal{T})$ such that $\mathcal{T} \models F \sqsubseteq E$ and $s \in R$. Let h be a homomorphism from C to $\mathcal{A}_1$ with $h(x_\varepsilon) = a$, and let $x_1, \dots, x_k$ be all elements of $\mathsf{var}(C)$ such that $h(x_i) = b$ and there is an x_i' with $s(x_i, x_i') \in C$ for some $s \in R$ and $h(x_i') = c$. If this list of elements is empty, then $\mathcal{A}_0 \models C(a)$ and we are done. Otherwise, let the concept C' be obtained from C by replacing certain $\exists s.G$ with $\mathcal{T} \models F \sqsubseteq G$ and $s \in R$ with D at every x_i, that is:

1. if $s(x_i, x_i') \in C$, $\mathcal{T} \models F \sqsubseteq C|_{x_i'}$ and $s \in R$, then remove from C the edge $s(x_i, x_i')$ and the subtree rooted at x_i';
2. add $B(x_i)$ for each concept name $B \in \mathsf{tlc}(D)$;
3. add the subconcept $\exists t.E$ rooted at x_i for each $\exists t.E$ that is a tlc in D.

We show that there is a concept C'' such that (i) there is a homomorphism from C'' to C' and $(A_0, C'') \in \widehat{\Gamma}$ and (ii) $\mathcal{A}_0 \models C'(a)$, starting with (i). Fix some x_i and consider the pair $\mu_{A_0,C}(x_i) = (F_i, S_i) \in \Gamma$ and let

$$p_i = (F_i, (S_i \setminus \{\exists s.G \mid \mathcal{T} \models F \sqsubseteq G, s \in R\})) \cup \mathsf{tlc}(D).$$

To construct C'', we use the pair p_i in place of (F_i, S_i) at x_i. The following two claims ensure that this possible and yields a concept C'' such that there is a homomorphism from C'' to C' (in fact, with $\mathsf{tlc}(C''|x_i) \subseteq \mathsf{tlc}(C'|x_i)$):

1. $p_i \in \widehat{\Gamma}$.
2. If $\exists s_0.G_0 \in (S_i \setminus \{\exists s.G \mid \mathcal{T} \models F \sqsubseteq G, s \in R\})$, then there exists $s(x_i, y) \in C' \cap C$ with $\mathcal{T} \models s \sqsubseteq s_0$ and $\mu_{A_0,C}(y) = (G_0, S')$ for some S'.

For Point 1, we show that rule (r2) is applicable to (F_i, S_i). To this end, we have to find $\exists s.G \in S_i$ such that $\mathcal{T} \models F \sqsubseteq G$ and $s \in R$. We find x_i' such that $s'(x_i, x_i') \in C$ for some $s' \in R$ and $h(x_i') = c$. By construction of $\mathcal{A}_1$ we must have $\mathcal{T} \models F \sqsubseteq C|_{x_i'}$. By construction of $\widehat{\Gamma}$ and since $(A_0, C) \in \widehat{\Gamma}$ and $s'(x_i, x_i') \in C$, there must be an s with $\mathcal{T} \models s' \sqsubseteq s$ and an $\exists s.G \in S_i$ such that $(G, C|_{x_i'}) \in \widehat{\Gamma}$. By Claim 1 from the proof of Lemma 13, this yields $\mathcal{T} \models C|_{x_i'} \sqsubseteq G$. Together with $\mathcal{T} \models F \sqsubseteq C|_{x_i'}$ we obtain $\mathcal{T} \models F \sqsubseteq G$ and from $s' \in R$ and $\mathcal{T} \models s' \sqsubseteq s$ we obtain $s \in R$. This finishes the proof of Point 1.

For Point 2, assume $\exists s_0.G_0 \in S_i$. There exists $s(x_i, y) \in C$ with $\mathcal{T} \models s \sqsubseteq s_0$ and $\mu_{A_0,C}(y) = (G_0, S')$ for some S'. Assume that $s(x_i, y) \notin C'$. We show that then we obtain $\mathcal{T} \models F \sqsubseteq G_0$ and $s_0 \in R$, from which Point 2 follows. From $s(x_i, y) \in C \setminus C'$ and the construction of $\mathcal{A}_1$ we obtain $\mathcal{T} \models F \sqsubseteq C|_y$. By Claim 1 from the proof of Lemma 13, $\mathcal{T} \models C|_y \sqsubseteq G_0$. Thus $\mathcal{T} \models F \sqsubseteq G_0$. From $s(x_i, y) \in C \setminus C'$ we obtain $s \in R$. Combined with $\mathcal{T} \models s \sqsubseteq s_0$ we obtain $s_0 \in R$, as required.

It follows that when constructing the pair $(A_0, C'') \in \widehat{\Gamma}$ we can follow the construction of (A_0, C) and when dealing with node x_i use the pair p_i in place of (F_i, S_i) and then proceed in a way such that the subtree rooted at x_i in C'' is the conjunction of $\mathsf{tlc}(D)$ and a subset of $\mathsf{tlc}(C'|_{x_i})$.

For (ii), we have to show that $\mathcal{A}_0 \models C'(a)$. Since $\mathcal{A}_0 \models D(b)$, there is a homomorphism h' from D to $\mathcal{A}_0$ that maps x_ε to b. We obtain a homomorphism of from C' to $\mathcal{A}_0$ that maps x_ε to a by starting with the homomorphism h and 'plugging in' h' at every node x_i to cover the subtrees generated by the subconcepts $\exists t.E$ of D added in Step 3 above.

❑

We split the proof of Theorem 7 into a soundness and completeness part.

Lemma 16 (Soundness)**.** *If Ω_Σ contains a root cycle, then A_0 is not FO-rewritable under $\mathcal{T}$ and Σ.*

Proof. Assume that Ω_Σ contains a root cycle. We show that there exists a blocked Σ-concept C such that $\mathcal{T} \models C \sqsubseteq A_0$ and $\mathcal{T} \not\models (C \setminus C|_{x_3}) \sqsubseteq A_0$, where x_3 is as in the definition of blocked concepts. Once we have constructed such a concept C, non-FO-rewritability can be proved in the same way as Theorem 12.

We will construct the desired concept C as the limit of a finite sequence of concepts $C_0, \dots, C_m$. Along with this concept sequence, we construct a sequence of mappings $\mu_0, \dots, \mu_m$ where μ_i associates with each variable in $\mathsf{var}(C_i)$ a tuple from Ω_Σ. To start the construction, choose a root tuple $t_\varepsilon \in \Omega_\Sigma$ and a looping tuple $t_{\mathsf{loop}} \in \Omega_\Sigma$ that is reachable from t_ε along $\rightsquigarrow_{\Omega_\Sigma}$. Set $C_0 = \bigsqcap\{A \in \mathsf{N_C} \mid A \in S_{t_\varepsilon}\}$, and $\mu_0(x_\varepsilon) = t_\varepsilon$.

Now assume that C_ℓ is already defined. To construct $C_{\ell+1}$, choose an $x \in \mathsf{var}(C_\ell)$ with $\mu_\ell(x) = (C_x, S_x, \mathsf{con}_x, E_x, \mathsf{xcon}_x)$ such that the set of existential restrictions in S_x is non-empty but x does not yet have any successors in C_ℓ. Since the set of existential restrictions in S_x is non-empty, the tuple $\mu_\ell(x)$ is not a leaf tuple and by the construction of Ω_Σ this implies that there are tuples $t_0, \dots, t_n \in \Omega$ such that Rule (r_Ω) *can be applied to* $t_0, \dots, t_n$ *using roles* $s_0, \dots, s_n \in \Sigma$ *and selected successor* $j \leq n$ *to generate* $\mu_\ell(x)$; i.e., for $t = \mu_\ell(x)$ such that $\exists r_0.D_0, \dots, \exists r_n.D_n$ are the existential restrictions in S_t and we have

- $\mathcal{T} \models s_i \sqsubseteq r_i$ and $C_{t_i} = D_i$ for $0 \leq i \leq n$;
- $E_t = \exists s_j.D_j$;
- there is a node pair $(C_t, S) \in \Gamma$ with $S_t \subseteq S$ and $S \cap \mathsf{N_C} = S_t \cap \mathsf{N_C}$;
- $\mathsf{con}_t = \mathsf{con}_\mathcal{T}(M)$, where
$$M = \bigcup(S_t \cap \mathsf{N_C}) \cup \{\exists s_i.G \mid i \leq n \text{ and } G \in \mathsf{con}_{t_i}\};$$
- $\mathsf{xcon}_t = \mathsf{con}_\mathcal{T}(M')$, where
$$M' = \bigcup(S_t \cap \mathsf{N_C}) \cup \{\exists s_j.G \mid G \in \mathsf{xcon}_{t_j}\} \cup \{\exists s_i.G \mid j \neq i \leq n \text{ and } G \in \mathsf{con}_{t_i}\}.$$

To construct $C_{\ell+1}$, add $s_i(x, y_i)$ to C_ℓ, for fresh individual variables y_i and all $i \leq n$. Further add the assertion $A(y_i)$ for each $A \in S_{t_i} \cap \mathsf{N_C}$. The resulting concept is $C_{\ell+1}$. Finally, $\mu_{\ell+1}$ is μ_ℓ extended by setting $\mu_{\ell+1}(y_i) = t_i$ for all $i \leq n$.

Unless guided in an appropriate way, the above construction need not terminate and will not result in a concept that has the desired properties. Let $\Omega^0_\Sigma, \Omega^1_\Sigma, \dots \Omega^k_\Sigma$ with $\Omega^k_\Sigma = \Omega_\Sigma$ be the sequence of sets generated by repeated application of Rule (r_Ω). For each $t \in \Omega_\Sigma$, let $\mathsf{rank}(t)$ denote the smallest i such that $t \in \Omega^i_\Sigma$. We guide, in the above construction, the selection of the tuples from Ω_Σ in the following way.

Step 1. We construct $C_0, C_1, \dots, C_{m_1}$ such that $\mu_{m_1}(x_1) = t_{\mathsf{loop}}$ for some leaf variable x_1 of C_{m_1}. We know that t_{loop} is reachable from t_ε along $\rightsquigarrow_{\Omega_\Sigma}$. We start with t_ε as before but at each step ℓ, we choose $t_0, \dots, t_n$ such that Rule (r_Ω) can be applied to $t_0, \dots, t_n$ using roles $s_0, \dots, s_n$ and selected successor $j \leq n$ to generate $\mu_\ell(x)$ so that t_j brings us closer to t_{loop}, that is, the shortest $\rightsquigarrow_{\Omega_\Sigma}$-path from t_j to t_{loop} is shorter than the shortest such path from $\mu_\ell(x)$ to t_{loop}. Let y_j be the individual that we introduced as a successor of x with $\mu_{\ell+1}(y_j) = t_j$. In step $\ell + 1$, we continue from y_j.

Step 2. Step 1 guarantees that we have $C_0, C_1, \dots, C_{m_1}$ with $\mu_{m_1}(x_1) = t_{\mathsf{loop}}$ for some leaf variable x_1 of C_{m_1}. In Step 2 we extend the sequence $C_0, C_1, \dots, C_{m_1}$ by $C_{m_1+1}, \dots, C_{m_2}$ in such a way that $\mu_{m_2}(x_2) = t'$ for some leaf variable x_2 of C_{m_2} such that the con and xcon components of t_{loop} and t' coincide. Since t_{loop} is looping, there is a tuple $t' \in \Omega_\Sigma$ such that t' is reachable from t_{loop} on a $\rightsquigarrow_{\Omega_\Sigma}$-path and the con and xcon components of t_{loop} agree with those of t'. To construct $C_{m_1+1}, \dots, C_{m_2}$, we start with x_1 and follow the same strategy as in Step 1, but this time to reach t' instead of t_{loop}.

Step 3. Let $C_0, C_1, \dots, C_{m_2}$ be the sequence constructed in Step 2. To finish the construction of C, we expand this sequence as follows. Assume $n \geq m_2$, C_n has been constructed, and $x \in \mathsf{var}(C_n)$ is such that x does not have any successors in C_n but there is some $\exists r.D \in S_x$. Then we choose $t_0, \dots, t_n$ such that Rule (r_Ω) can be applied to $t_0, \dots, t_n$ using roles $s_0, \dots, s_n$ and selected successor $j \leq n$ to generate $\mu_\ell(x)$ so that $t_0, \dots, t_n$ brings us closer to a leaf tuple, i.e., $\mathsf{rank}(t_i) < \mathsf{rank}(\mu_\ell(x))$ for all $i \in \{0, \dots, n\}$. Note that, by construction of Ω_Σ, this is always possible.

It should be clear that this way of guiding tuple selection guarantees termination. Call the concept and mapping obtained in the limit C and μ, respectively. For each $x \in \mathsf{var}(C)$, p_x is defined as the (unique) path $x_0, \dots, x_k$ such that $x_0 = x$, x_k is a leaf of C, and for all $i \in \{0, \dots, k-1\}$, if $t_0, \dots, t_n$ are the tuples that we picked in the inductive construction above while adding successors of x_i using Rule (r_Ω) with selected successor $j \leq n$, then we have $\mu(x_{i+1}) = t_j$.

Claim. *For all $x \in \mathsf{var}(C)$ we have*

1. $\mathsf{con}_{\mu(x)} = \mathsf{con}^C_{\mathcal{T}}(x)$*;*
2. *if $y \neq x$ is the last element of p_x, then* $\mathsf{xcon}_{\mu(x)} = \mathsf{con}^{C \setminus C|_y}_{\mathcal{T}}(x)$.

Proof of claim. The proof is by induction on the length of the path p_x. As the base case, p_x is of length one and thus, $\mu(x)$ is a leaf tuple. By definition, this implies that S_x consists of concept names. Let $S_{\mu(x)} = \{A_1, \dots, A_k\}$. By our construction, $C|_x = A_1 \sqcap \dots \sqcap A_k$. Let $D \in \mathsf{sub}(\mathcal{T})$. By definition, $D \in \mathsf{con}_{\mu(x)}$ iff $\mathcal{T} \models A_1 \sqcap \dots \sqcap A_k \sqsubseteq D$, which is Point 1 in the claim. Since $\mu(x)$ is a leaf, we have $y = x$ for the last element y of p_x. Hence Point 2 in the claim holds trivially and the base case is proved.

For the inductive step, suppose p_x is of length $i > 1$. Let $t_0, \dots, t_n$ be the tuples that we picked in the inductive construction of C while adding successors $y_0, \dots, y_n$ of x using Rule (r_Ω) with roles $s_0, \dots, s_n$ and selected successor $\ell \leq n$. Assume the existential restrictions in $S_{\mu(x)}$ are $\exists r_0.D_0, \dots, \exists r_n.D_n$ and we have

- $s_k(x, y_k) \in C$ and $C_{\mu(y_k)} = D_k$ for $0 \leq k \leq n$.
- $\mathsf{con}_{\mu(x)} = \mathsf{con}_{\mathcal{T}}(M)$, where
$$M = \bigcup (S_{\mu(x)} \cap \mathsf{N_C}) \cup \{\exists s_k. \bigsqcap \mathsf{con}_{\mu(y_k)} \mid k \leq n\}$$
- $\mathsf{xcon}_{\mu(x)} = \mathsf{con}_{\mathcal{T}}(M')$, where
$$M' = \bigcup (S_{\mu(x)} \cap \mathsf{N_C}) \cup \{\exists s_\ell. \bigsqcap \mathsf{xcon}_{\mu(y_\ell)}\} \cup \{\exists s_k. \bigsqcap \mathsf{con}_{\mu(y_k)} \mid \ell \neq k \leq n\}.$$

Observe that

$$C|_x = \bigsqcap (S_{\mu(x)} \cap \mathsf{N_C}) \sqcap \exists s_0.C|_{y_0} \sqcap \dots \exists s_n.C|_{y_n}.$$

The IH implies

- $\mathsf{con}_{\mu(y_k)} = \mathsf{con}^C_{\mathcal{T}}(y_k)$ for all $k \leq n$;
- $\mathsf{xcon}_{\mu(y_\ell)} = \mathsf{con}^{C \setminus C|_y}_{\mathcal{T}}(y_\ell)$ for the last element y of p_x.

Hence we obtain by the semantics that $\mathsf{con}_{\mu(x)} = \mathsf{con}^C_{\mathcal{T}}(x)$ and $\mathsf{xcon}_{\mu(x)} = \mathsf{con}^{C \setminus C|_y}_{\mathcal{T}}(x)$, as required. ⊣

Recall that there is a path $x_0, \dots, x_k \in \mathsf{var}(C)$ such that x_0 is the root of C, x_k is a leaf of C, for all $i < k$, x_{i+1} represents the selected successor of $\mu(x_i)$, and there are nodes x_p and x_q on the path such that $p < q$, $\mu(x_p) = t_{\mathsf{loop}}$, and $\mu(x_q) = t'$. By the claim we have

- $\mathsf{con}^C_{\mathcal{T}}(x_p) = \mathsf{con}_{\mu(x_p)} = \mathsf{con}_{\mu(x_q)} = \mathsf{con}^C_{\mathcal{T}}(x_q)$;
- $\mathsf{con}^{C \setminus C|_{x_k}}_{\mathcal{T}}(x_p) = \mathsf{xcon}_{\mu(x_p)} = \mathsf{xcon}_{\mu(x_q)} = \mathsf{con}^{C \setminus C|_{x_k}}_{\mathcal{T}}(x_q)$.

It follows that C is blocked and $\mathcal{T} \models C \sqsubseteq A_0$ but $\mathcal{T} \not\models C \setminus C|_{x_k} \sqsubseteq A_0$, as required. ❑

Lemma 17 (Completeness)**.** *If A_0 is not FO-rewritable under $\mathcal{T}$ and Σ, then Ω_Σ contains a root cycle.*

Proof. We show the contrapositive and assume that Ω_Σ contains no root cycle. Let

$$\Lambda_\Sigma := \{C \mid (A_0, C') \in \widehat{\Gamma}_\Sigma,\ C \prec^* C',\ \text{and } C \text{ is } \prec\text{-minimal with } \mathcal{T} \models C \sqsubseteq A_0\}.$$

By Proposition 6, $\bigvee \Lambda_\Sigma$ is a (potentially infinite) FO-rewriting of A_0 under $\mathcal{T}$ and Σ, thus it suffices to show that Λ_Σ is finite. Since the outdegree of concepts in Λ_Σ is bounded

by $|\mathcal{T}|$, it remains to show that the depth of concepts in Λ_Σ is bounded by $2^{2|\mathcal{T}|}$.

Assume to the contrary that there is a $C \in \Lambda_\Sigma$ that contains a path $x_0, \dots, x_m$ (starting at the root x_0) with $m > 2^{2|\mathcal{T}|}$. By definition of Λ_Σ, there is a $(A_0, C') \in \widehat{\Gamma}_\Sigma$ with $C \prec^* C'$. Note that C can be obtained from C' by removing subtrees. Thus, each node in C is in the domain of the function $\mu_{A_0,C'}$ from the completeness proof for $\widehat{\Gamma}$. Associate with each node $x \in \mathsf{var}(C)$ a node tuple $\zeta(a) \in \Omega_\Sigma$ in a bottom-up way as follows:

(a) for each leaf x of C with $\mu_{A_0,C'}(x) = (D, S)$ set

$$\zeta(x) = (D, S \cap \mathsf{N_C}, \mathsf{con}_\mathcal{T}(S \cap \mathsf{N_C}), -, -).$$

(b) for each non-leaf node x of C with $\mu_{A_0,C'}(x) = (D, S)$ set

$$\zeta(x) = (D, S', \mathsf{con}_\mathcal{T}(M), E, \mathsf{con}_\mathcal{T}(M'))$$

where

- $S' = (S \cap \mathsf{N_C}) \cup \{\exists r.F \in S \mid s(x,y) \in C \text{ for some } s \text{ with } \mathcal{T} \models s \sqsubseteq r \text{ and } \mu_{A_0,C}(y) = (F, S'')\}$;
- $M = (S \cap \mathsf{N_C}) \cup \{\exists r. \bigsqcap \mathsf{con}_{\zeta(y)} \mid r(x,y) \in C\}$;
- choose some $\exists r.F \in S'$ and $s(x,y) \in C$ such that $\mathcal{T} \models s \sqsubseteq r$, $\mu_{A_0,C}(y) = (F, S'')$, and if $x = x_i$ is on the path $x_0, \dots, x_m$, then $y = x_{i+1}$.[2] Then set
  - $E = \exists s.F$ and
  - $M' = (S \cap \mathsf{N_C}) \cup \{\exists s. \bigsqcap \mathsf{xcon}_{\zeta(y)}\} \cup \{\exists r.\mathsf{con}_{\zeta(y')} \mid r(x,y') \in C \text{ and } y' \neq y\}$.

One can show by induction on the co-depth of nodes $x \in \mathsf{var}(C)$ that $\zeta(x) \in \Omega_\Sigma$ and that

1. $\mathsf{con}^C_\mathcal{T}(x) = \mathsf{con}_{\zeta(x)}$, for all $x \in \mathsf{var}(C)$;
2. $\mathsf{con}^{C \setminus C|_{x_m}}_\mathcal{T}(x_i) = \mathsf{xcon}_{\zeta(x_i)}$, for all $i < m$.

By Point 1 above and since $C, \mathcal{T} \models A_0(x_0)$, we have $A_0 \in \mathsf{con}_{\zeta(x_0)}$. Since C is $\prec$-minimal with $\mathcal{T} \models C \sqsubseteq A_0$, we have $\mathcal{T} \not\models C \setminus C|_{x_m} \sqsubseteq A_0$ and so $A_0 \notin \mathsf{xcon}_{\zeta(x_0)}$ by Point 2 above. By construction of ζ, $\zeta(x_i) \rightsquigarrow_{\Omega_\Sigma} \zeta(x_{i+1})$ for all $i < m$. Since $m > 2^{2|\mathcal{T}|}$, among the $x_0, \dots, x_m$ there must be x_i and x_j such that $i \neq j$ and the con and xcon components of $\zeta(x_i)$ and $\zeta(x_j)$ are identical. We have thus shown that Ω_Σ contains a root cycle, which contradicts our assumption that there is no such cycle. ❑

We fix some notation for the semantics of datalog programs. Assume Π is a monadic datalog program with goal predicate $G(x)$. For an ABox $\mathcal{A}$, we define a sequence of ABoxes $\Pi^0(\mathcal{A}), \Pi^1(\mathcal{A}), \dots$ by setting:

- $\Pi^0(\mathcal{A}) = \mathcal{A}$;
- $\Pi^{n+1}(\mathcal{A})$ is defined by adding to $\Pi^n(\mathcal{A})$ the set of all $P(a)$ with $a \in \mathsf{ind}(\mathcal{A})$ and P an IDB of Π such that there exists a rule $P(x) \leftarrow \varphi$ in Π and a variable assigment π with $\pi(x) = a$ and $\Pi^n(\mathcal{A}) \models_\pi \varphi$.

[2] Note that the required $\exists r.F$ and $s(x,y)$ indeed exist. In particular, when $x = x_i$ then by definition of $\widehat{\Gamma}_\Sigma$ we find an $\exists r.F \in S$ and $s(x, x_{i+1}) \in C'$ such that $\mathcal{T} \models s \sqsubseteq r$ and $\mu_{A_0,C} = (F, S'')$; since $s(x, x_{i+1}) \in C$, by definition of S' we also have $\exists r.F \in S$.

Set $\Pi(\mathcal{A}) = \bigcup_{n \geq 0} \Pi^n(\mathcal{A})$. We use $\mathsf{con}^\mathcal{A}_\mathcal{T}(a)$ to denote $\{D \in \mathsf{sub}(\mathcal{T}) \mid \mathcal{A}, \mathcal{T} \models D(a)\}$.

Theorem 8.

1. *The program $\Pi_{A_0,\mathcal{T}}$ is a rewriting of A_0 under $\mathcal{T}$.*
2. *If A_0 is FO-rewritable under $\mathcal{T}$ and Σ, then $\Pi_{A_0,\mathcal{T}}$ is non-recursive.*

Proof. For Point 1, first assume $P_{A_0,\mathsf{con}_0,\mathsf{XCON}_0}(a_0) \in \Pi_{A_0,\mathcal{T}}(\mathcal{A})$ for a goal predicate $P_{A_0,\mathsf{con}_0,\mathsf{XCON}_0}$ of $\Pi_{A_0,\mathcal{T}}$. We have to show that $\mathcal{A}, \mathcal{T} \models A_0(a_0)$. The following claim can be proved by induction on n using the construction of $\Pi_{A_0,\mathcal{T}}$:

Claim. For all $n \geq 0$, $P_{C,\mathsf{con},\mathsf{XCON}}(a) \in \Pi^n(\mathcal{A})$ implies $\mathsf{con} \subseteq \mathsf{con}^\mathcal{A}_\mathcal{T}(a)$.

We leave details to the reader, but note that in the induction step, it is crucial to use Condition 4 of the rule (r_Ω).

Now, since $P_{A_0,\mathsf{con}_0,\mathsf{XCON}_0}(a_0) \in \Pi_{A_0,\mathcal{T}}(\mathcal{A})$ and $A_0 \in \mathsf{con}_0$ (by the definition of goal predicates in $\Pi_{\mathcal{T},A_0}$), we obtain $A_0 \in \mathsf{con}^\mathcal{A}_\mathcal{T}(a_0)$, as required.

Conversely, assume that $\mathcal{A}, \mathcal{T} \models A_0(a_0)$. We have to show that $G(a_0) \in \Pi_{\mathcal{T},A_0}(\mathcal{A})$ for a goal predicate G of $\Pi_{A_0,\mathcal{T}}$. By Proposition 6, there is an $(A_0, C') \in \widehat{\Gamma}_\Sigma$ with $\mathcal{A} \models C'(a_0)$. Let $C \prec^* C'$ be minimal with $\mathcal{T} \models C \sqsubseteq A_0$. Since C homomorphically maps to $\mathcal{A}$, it suffices to show that $G(x_\varepsilon) \in \Pi_{A_0,\mathcal{T}}(C)$ for some goal predicate G and with x_ε the root of C (where C is viewed as an ABox).

In what follows, we will make use of the function $\mu_{A_0,C'}$ from the completeness proof for $\widehat{\Gamma}_\Sigma$, which associates each $x \in \mathsf{var}(C')$ with a tuple $\mu_{A_0,C'}(x)$ from Γ_Σ. For brevity, when $\mu_{A_0,C'}(x) = (D, S)$, we use C_x to denote D and S_x to denote S. Note that C can be obtained from C' by removing subtrees and thus, each node in C is in the domain of the function $\mu_{A_0,C'}$. For each $s(x,y) \in C'$, we use $\mathsf{ex}(y)$ to denote the existential restriction $\exists r.D \in S_x$ for which the successor y was generated during the construction of $(A_0, C') \in \widehat{\Gamma}_\Sigma$.

For each $x \in \mathsf{var}(C)$, let $\mathsf{XCON}^C_\mathcal{T}(x) = \{-\}$ if x is a leaf in C and otherwise let $\mathsf{XCON}^C_\mathcal{T}(x)$ denote the set of all sets $\mathsf{con}^{C \setminus C|_y}_\mathcal{T}(x)$ where y is a leaf node of C that is in the subtree rooted at x, but distinct from x. We aim to show that for each $x \in \mathsf{var}(C)$, the following rule R_x is in $\Pi_{A_0,\mathcal{T}}$:

$$P_{C_x,\mathsf{con}^C_\mathcal{T}(x),\mathsf{XCON}^C_\mathcal{T}(x)}(v) \leftarrow \bigwedge_{A(x) \in C} A(v) \wedge \bigwedge_{s(x,y) \in C} (s(v,w_y) \wedge P_{C_y,\mathsf{con}^C_\mathcal{T}(y),\mathsf{XCON}^C_\mathcal{T}(y)}(w_y)).$$

It is then easy to prove by induction on the co-depth of x that for all $x \in \mathsf{var}(C)$, we have $P_{C_x,\mathsf{con}^C_\mathcal{T}(x),\mathsf{XCON}^C_\mathcal{T}(x)}(x) \in \Pi_{\mathcal{T},A_0}(C)$. From $C, \mathcal{T} \models A_0(x_\varepsilon)$, we obtain $A_0 \in \mathsf{con}^C_\mathcal{T}(x_\varepsilon)$; moreover, the definition of $\mu_{A_0,C'}$ yields $C_{x_\varepsilon} = A_0$; and finally, the minimality of C ensures that $A_0 \notin \mathsf{xcon}$ for all $\mathsf{xcon} \in \mathsf{XCON}^C_\mathcal{T}(x_\varepsilon)$. Consequently, $P_{C_{x_\varepsilon},\mathsf{con}^C_\mathcal{T}(x_\varepsilon),\mathsf{XCON}^C_\mathcal{T}(x_\varepsilon)}$ is a goal predicate of $\Pi_{A_0,\mathcal{T}}$ and we are done.

The proof that R_x is a rule in $\Pi_{A_0,\mathcal{T}}$ for each $x \in \mathsf{var}(C)$ is by induction on the co-depth of x. Along with the induction, we associate a tuple $t_x \in \Omega$ with each $x \in \mathsf{var}(C)$ such that $C_{t_x} = C_x$ and $\mathsf{con}_{t_x} = \mathsf{con}^C_{\mathcal{T}}(x)$.

Start with x being a leaf node. We first show that there is a $t_x \in \Omega$ with $C_{t_x} = C_x$ and $\mathsf{con}_{t_x} = \mathsf{con}^C_{\mathcal{T}}(x)$. By construction of $(A_0, C') \in \widehat{\Gamma}_\Sigma$, we have $(C_x, S_x) \in \Gamma_\Sigma$ and $S_x \cap \mathsf{N_C} = \{A \mid A(x) \in C'\}$. By construction of C from C', this yields $S_x \cap \mathsf{N_C} = \{A \mid A(x) \in C\}$. Consequently, $\mathsf{con}^C_{\mathcal{T}}(x) = \mathsf{con}_{\mathcal{T}}(S_x \cap \mathsf{N_C})$. Thus $t_x = (C_x, S_x \cap \mathsf{N_C}, \mathsf{con}_{\mathcal{T}}(S_x \cap \mathsf{N_C}), -, -)$ is one of the tuples initially added to Ω, and is as required. It now suffices to note that, By definition of $\Pi_{A_0,\mathcal{T}}$ (rules added initially), $t_x \in \Omega$ results in the rule R_x to be included in $\Pi_{A_0,\mathcal{T}}$.

Assume now that $x \in \mathsf{var}(C)$ is a non-leaf. As in the induction start, $(C_x, S_x) \in \Gamma_\Sigma$. Let S be obtained from S_x by removing all existential restrictions $\mathsf{ex}(y)$ such that $r(x, y) \in C' \setminus C$ and let the existential restrictions in S be $\exists r_0.D_0, \ldots, \exists r_n.D_n$. By construction of $\widehat{\Gamma}_\Sigma$ and of C from C', for each $i \leq n$ we find an $s_i(x, y_i) \in C'$ with $s_i \in \Sigma$, $\mathcal{T} \models s_i \sqsubseteq r_i$, and $C_{y_i} = D_i$. Let $\ell \leq n$ be arbitrary. We aim to show that $t_{y_0}, \ldots, t_{y_n}$ together with $s_0, \ldots, s_n$ and ℓ result in the rule (r_Ω) to add a tuple t_x to Ω such that $C_{t_x} = C_x$, $S_t = S$, and $\mathsf{con}_{t_x} = \mathsf{con}^C_{\mathcal{T}}(x)$. This can actually be verified by checking Conditions 1 to 5 of (r_Ω), recalling from above that

- (for Condition 1) $\mathcal{T} \models s_i \sqsubseteq r_i$ and $C_{y_i} = D_i$,
- (for Condition 3) $(C_x, S_x) \in \Gamma_\Sigma$, $S \subseteq S_x$, and $S_x \cap \mathsf{N_C} = S \cap \mathsf{N_C}$;
- (for Condition 4) $\mathsf{con}_{t_{y_i}} = \mathsf{con}^C_{\mathcal{T}}(y_i)$ and $\mathsf{con}^C_{\mathcal{T}}(x) = \mathsf{con}_{\mathcal{T}}(M)$ where

$$M = (S \cap \mathsf{N_C}) \cup \{\exists s_i. \bigsqcap \mathsf{con}^C_{\mathcal{T}}(y_i) \mid i \leq n\}.$$

It remains to show that R_x is added as a rule in $\Pi_{A_0,\mathcal{T}}$ (in the second step). This is witnessed by the tuples $t_x, t_{y_0}, \ldots, t_{y_n} \in \Omega$, the role names $s_0, \ldots, s_n$, and the $\ell \leq n$ chosen above. In fact, we have already shown that Conditions 1 to 5 of (r_Ω) are satisfied. It remains to verify that Conditions 2 and 3 from the second step of the construction of $\Pi_{A_0,\mathcal{T}}$ are satisfied. In fact, Condition 2 is implied by $R_{y_0}, \ldots, R_{y_n}$ being rules in $\Pi_{A_0,\mathcal{T}}$. For Condition 3, we have to show that $\mathsf{XCON}^C_{\mathcal{T}}(x)$ consists of all sets $\mathsf{con}_{\mathcal{T}}(M')$ such that there is an $\ell' \leq n$ and an $\mathsf{xcon} \in \mathsf{XCON}^C_{\mathcal{T}}(y_{\ell'})$ with

$$M' = (S \cap \mathsf{N_C}) \cup \{\exists s_{\ell'}. \bigsqcap \mathsf{xcon}\} \cup \{\exists s_i. \bigsqcap \mathsf{con}^C_{\mathcal{T}}(y_i) \mid \ell' \neq i \leq n\}.$$

This, however, is straightforward by the definition of $\mathsf{XCON}^C_{\mathcal{T}}$ and the semantics.

Now for Point 2 of the theorem. Assume that A_0 is FO-rewritable under $\mathcal{T}$ and Σ and that, to the contrary of what we aim to show, $\Pi_{A_0,\mathcal{T}}$ is recursive. Due to the elimination of accidental recursiveness, we find rules

$$\begin{aligned} P_{C_0,\mathsf{con}_0,\mathsf{XCON}_0}(x) &\leftarrow \varphi_0(x) \\ &\cdots \\ P_{C_m,\mathsf{con}_m,\mathsf{XCON}_m}(x) &\leftarrow \varphi_m(x) \end{aligned}$$

such that

(i) $P_{C_0,\mathsf{con}_0,\mathsf{XCON}_0}$ is a goal predicate,

(ii) $P_{C_i\mathsf{con}_i,\mathsf{XCON}_i}$ occurs in φ_{i-1} for $1 \leq i \leq m$ and

(iii) $P_{C_m,\mathsf{con}_m,\mathsf{XCON}_m} = P_{C_p,\mathsf{con}_p,\mathsf{XCON}_p}$ for some $p < m$.

We show that this situation gives raise to a $\rightsquigarrow_{\Omega_\Sigma}$-path through Ω_Σ that starts at a root tuple and has length exceeding $2^{2|\mathcal{T}|}$. Since any such path must contain a root cycle, by Theorem 7 we have derived a contradiction to the FO-rewritability of A_0 under $\mathcal{T}$ and Σ. We first establish a technical claim that is easily proved by considering the construction of $\Pi_{A_0,\mathcal{T}}$ (and Ω_Σ).

Claim. If the predicate $P_{C,\mathsf{con},\mathsf{XCON}}$ occurs in $\Pi_{A_0,\mathcal{T}}$, then for every $\mathsf{xcon} \in \mathsf{XCON}$, there is a tuple $t \in \Omega_\Sigma$ that is of the form $(C, S_t, E_t, \mathsf{con}, \mathsf{xcon})$.

We now construct the $\rightsquigarrow_{\Omega_\Sigma}$-path through Ω_Σ by traveling backwards along "$\rightsquigarrow_{\Omega_\Sigma}$". With each tuple $t \in \Omega_\Sigma$ on the path, we associate one of the rules $P_{C_i,\mathsf{con}_i,\mathsf{XCON}_i}(x) \leftarrow \varphi_i(x)$ from above such that $C_t = C_i$, $\mathsf{con}_t = \mathsf{con}_i$ and $\mathsf{xcon}_t \in \mathsf{XCON}_i$. To start the path, we choose a tuple $t \in \Omega_\Sigma$ such that $C_t = C_m$, $\mathsf{con}_t = \mathsf{con}_m$, and $\mathsf{xcon}_t \in \mathsf{XCON}_m$, which exists by the claim above.

Assume that an initial piece $\widehat{t}_k \rightsquigarrow_{\Omega_\Sigma} \cdots \rightsquigarrow_{\Omega_\Sigma} \widehat{t}_0$ of the path has already been constructed. To extend it, let the rule associated with $\widehat{t}_k$ be the j-th one from the list above. Since $P_{C_j,\mathsf{con}_j,\mathsf{XCON}_j}$ occurs in φ_{j-1}, the presence of the $j-1$-st rule in $\Pi_{A_0,\mathcal{T}}$ yields tuples $t, t_0, \ldots, t_n$, role names $s_0, \ldots, s_n \in \Sigma$, $\ell, \ell' \in \{0, \ldots, n\}$, and sets $\mathsf{XCON}_0, \ldots, \mathsf{XCON}_n$ such that

(a) $C_{j-1} = C_t$ and $\mathsf{con}_{j-1} = \mathsf{con}_t$;

(b) $C_{t_{\ell'}} = C_j = C_{\widehat{t}_k}$, $\mathsf{con}_{t_{\ell'}} = \mathsf{con}_j = \mathsf{con}_{\widehat{t}_k}$, and $\mathsf{XCON}_{\ell'} = \mathsf{XCON}_j$;

(c) Conditions 1 to 3 from the construction of $\Pi_{A_0,\mathcal{T}}$ are satisfied.

We can assume w.l.o.g. that $\ell = \ell'$ and $t_{\ell'} = \widehat{t}_k$. In fact, let t' be as t, but with E_t the ℓ'-th existential restriction in place of the ℓ-th one and with the xcon-component obtained as in Condition 5 of (r_Ω) but with the new ℓ and by using $\widehat{t}_k$ in place of $t_{\ell'}$ as the ℓ'-th successor tuple. It can be verified that $t', t_0, \ldots, \widehat{t}_k, \ldots, t_n$, $s_0, \ldots, s_n$, and ℓ' still satisfy Conditions 1 to 5 from (r_Ω); in particular, the con-component of t is unaffected by replacing $t_{\ell'}$ with $\widehat{t}_k$ due to Point (b) above. Thus, $t' \in \Omega$. Moreover, it can be verified that the rule in $\Pi_{A_0,\mathcal{T}}$ generated by $t', t_0, \ldots, \widehat{t}_k, \ldots, t_n$, $s_0, \ldots, s_n$, and ℓ' is stilly exactly the $j-1$-st one above.

We want to use t as the next tuple on the path and associate it with the $j-1$-st rule above. We have to show that $t \rightsquigarrow_{\Omega_\Sigma} t_k$ and that $\mathsf{xcon}_t \in \mathsf{XCON}_{j-1}$. The former is immediate due to Condition 1 of the construction of $\Pi_{A_0,\mathcal{T}}$. This condition also implies, via Point 5 of the rule (r_Ω), that $\mathsf{xcon}_t = \mathsf{con}_{\mathcal{T}}(M')$ where

$$M' = (S_t \cap \mathsf{N_C}) \cup \{\exists s_\ell. \bigsqcap \mathsf{xcon}_{t_\ell}\} \cup \{\exists s_i. \bigsqcap \mathsf{con}_{t_i} \mid \ell \neq i \leq n\}.$$

Since $t_\ell = \widehat{t}_k$ and $\mathsf{xcon}_{\widehat{t}_k} \in \mathsf{XCON}_j = \mathsf{XCON}_\ell$, it is thus an immediate consequence of Condition 3 from the construction of $\Pi_{A_0,\mathcal{T}}$ that $\mathsf{xcon}_t \in \mathsf{XCON}_{j-1}$ as required.

Proceeding in this way, we can continue to build up a backwards path through Ω_Σ. When we treat a tuple t_k associated with the p-th rule, we can choose either to go to the $p-1$-st rule or to the $m-1$-st rule. We choose the second alternative sufficiently often so that the length of the path exceeds $2^{2|\mathcal{T}|}$. Eventually: we choose the first alternative to make sure that the constructed path starts with tuple $t \in \Omega$ that is associated with the 0-th rule above. Since $P_{C_0,\mathsf{con}_0,\mathsf{XCON}_0}$ is a goal predicate, we have $A_0 \in \mathsf{con}_0$ and $A_0 \notin \mathsf{xcon}$ for all $\mathsf{xcon} \in \mathsf{XCON}_0$. Consequently, t must be a root tuple and thus the constructed path is a root cycle, as desired. ❑

Theorem 9. *If A_0 is FO-rewritable under $\mathcal{T}$ and Σ, then it has a monadic non-recursive datalog rewriting of size at most $2^{p(n)}$, n the size of $\mathcal{T}$ and $p()$ a polynomial.*

There is a family of TBoxes $\mathcal{T}_1, \mathcal{T}_2, \ldots$ such that for all $n \geq 1$, $\mathcal{T}_n$ is of size $\mathcal{O}(n^2)$, the concept name A_0 is FO-rewritable under $\mathcal{T}_n$, and the smallest non-recursive monadic datalog rewriting has size at least 2^n.

Proof. For the first part of the theorem, assume that A_0 is FO-rewritable under $\mathcal{T}$ and Σ. Let n be the size of $\mathcal{T}$. Consider the datalog rewriting Π_{Γ_Σ} constructed after Proposition 6. Π_{Γ_Σ} is guaranteed to be a rewriting of A_0 under $\mathcal{T}$ and Σ, but it might be recursive even if A_0 is FO-rewritable under $\mathcal{T}$ and Σ. Moreover, Π_{Γ_Σ} is of size single exponential in n and every rule body contains at most n atoms. We establish the following central claim.

Claim 1. If $\mathcal{A}$ is a Σ-ABox and $a \in \Pi_{\Gamma_\Sigma}(\mathcal{A})$, then there is a tree-shaped Σ-ABox $\mathcal{A}'$ with root a and of depth at most 2^{3n^2} such that $a \in \Pi_{\Gamma_\Sigma}(\mathcal{A}')$ and there is a homomorphism from $\mathcal{A}'$ to $\mathcal{A}$ that is the identity on a.

To prove Claim 1, let $a \in \Pi_{\Gamma_\Sigma}(\mathcal{A})$. Then $\mathcal{A}, \mathcal{T} \models A_0(a)$. It follows from results on *unraveling tolerance* in [Bienvenu *et al.*, 2012b] that then also $\mathcal{A}_u, \mathcal{T} \models A_0(a)$, where $\mathcal{A}_u$ is the unraveling of $\mathcal{A}$ into a (potentially infinite) tree-shaped ABox with root a. There is a homomorphism from $\mathcal{A}_u$ to $\mathcal{A}$ that is the identity on a. From $\mathcal{A}_u, \mathcal{T} \models A_0(a)$, we obtain $a \in \Pi_{\Gamma_\Sigma}(\mathcal{A}_u)$ and thus there is a proof tree for $A_0(a)$ from $\mathcal{A}_u$ and $\Pi_{\Gamma_\Sigma}(a)$. We show that there is such a proof tree that only uses individuals in $\mathcal{A}'_u$ whose distance from the root is at most 2^{3n^2}. Then the restriction of $\mathcal{A}_u$ to depth 2^{3n^2} is the desired ABox $\mathcal{A}'$. Assume to the contrary that there is no proof tree of the required form. Then the restriction $\mathcal{A}'$ of $\mathcal{A}_u$ to depth 2^{3n^2} satisfies $a \notin \Pi_{\Gamma_\Sigma}(\mathcal{A}')$, thus $\mathcal{A}', \mathcal{T} \not\models A_0(a)$. Moreover, since $a \in \Pi_{\Gamma_\Sigma}(\mathcal{A}_u)$ there is a finite subset $\mathcal{A}''$ of $\mathcal{A}_u$ (which w.l.o.g. can be assumed to be tree-shaped and contain $\mathcal{A}'$) such that $a \in \Pi_{\Gamma_\Sigma}(\mathcal{A}'')$, thus $\mathcal{A}'', \mathcal{T} \models A_0(a)$. By Lemma 11, we obtain a contradiction to A_0 being FO-rewritable under $\mathcal{T}$ and Σ. This finishes the proof of Claim 1.

By Claim 1, for any Σ-ABox $\mathcal{A}$ with $a \in \Pi_{\Gamma_\Sigma}(\mathcal{A})$, there is a proof tree for $A_0(a)$ from $\mathcal{A}_u$ and $\Pi_{\Gamma_\Sigma}(a)$ iff there is such a proof tree of depth at most 2^{3n^2}. To see this, note that IDB predicates occur in the bodies of the rules in Π_{Γ_Σ} in a very restricted way: when the variable in the head is x, then the only occurrences of IDB predicates in the rule body are of the form $r(x, y) \wedge P(y)$ where P is the IDB predicate and r is EDB. Consequently, the depth of a proof tree for a tree-shaped ABox cannot exceed the depth of that ABox and the homomorphism mentioned in Claim 1 allows us to transfer proof trees from the tree-shaped ABox $\mathcal{A}'$ to the original ABox $\mathcal{A}$.

We are now ready to establish the first part of the theorem. Given that Π_{Γ_Σ} is a (potentially cyclic) rewriting of A_0 and $\mathcal{T}$ and that we only need to worry about proof trees of depth at most 2^{3n^2}, we can break the cycles in Π_{Γ_Σ} in a straightforward way:

- replace each IDB predicate P with P_i, $i \leq 2^{3n^2}$;
- each rule $P(x) \leftarrow \varphi$ is replaced by the set of all rules $P_i(x) \leftarrow \varphi'$ where φ' is obtained from φ by replacing each occurrence of an IDB predicate P' with P'_j, for some $j < i$.

For the second part of the theorem, we first observe that one can assume w.l.o.g. that the monadic datalog programs are connected. We call a rule $P(x) \leftarrow \varphi$ *connected* if the graph whose vertices are x and the remaining variables in φ and whose edges are $\{x_1, x_2\}$ for $r(x_1, x_2) \in \varphi$ is connected. A monadic datalog program is *connected* if all its rules are connected.

Claim 2. If Π is a non-recursive monadic datalog rewriting of A_0 relative to $\mathcal{T}$, then there exists a connected non-recursive monadic datalog rewriting Π' of A_0 relative to $\mathcal{T}$ whose size does not exceed the size of Π.

To prove Claim 2, assume that Π is given. Define Π' by replacing each rule $P(x) \leftarrow \varphi$ with the rule $P(x) \leftarrow \varphi'$, where φ' is obtained from φ by removing all atoms $A(y)$ and $r(x_1, x_2)$ whose variables are not connected to x in φ (thus, if x does not occur in φ then $P(x) \leftarrow \varphi$ is replaced by $P(x) \leftarrow \mathsf{true}$. We show that Π' is a datalog rewriting of A_0 relative to $\mathcal{T}$. Clearly $\Pi(\mathcal{A}) \subseteq \Pi'(\mathcal{A})$ for any ABox $\mathcal{A}$. Thus, it remains to show that if $G(a) \in \Pi'(\mathcal{A})$ for an ABox $\mathcal{A}$, $a \in \mathsf{ind}(\mathcal{A})$, and goal predicate G, then $G'(a) \in \Pi(\mathcal{A})$ for some goal predicate G'. Assume this is not the case. Take $\mathcal{A}$, $a \in \mathsf{ind}(\mathcal{A})$, and goal predicate G such that $G(a) \in \Pi'(\mathcal{A})$ but $G'(a) \notin \Pi(\mathcal{A})$ for any goal predicate G'. Define an ABox $\mathcal{A}'$ by adding to $\mathcal{A}$ the assertions $A(c)$ and $r(c, c)$ for every concept name A and role name r in Π and a fresh individual name c. Clearly $G(a) \in \Pi(\mathcal{A}')$ since the atoms removed from Π can all be satisfied in c. Hence $\mathcal{A}', \mathcal{T} \models A_0(a)$ since Π is a rewriting of A_0 relative to $\mathcal{T}$. But then $\mathcal{A}, \mathcal{T} \models A_0(a)$ since $\mathcal{T}$ is an $\mathcal{EL}$-TBox and c is disconnected from the individuals in $\mathcal{A}$. We have derived a contradiction to the assumption that Π is a rewriting of A_0 relative to $\mathcal{T}$. This finishes the proof of Claim 2.

We now provide a reformulation of (the second part of) Theorem 9. A *tree-UCQ-rewriting of A_0 relative to $\mathcal{T}$* is a disjunction $\bigvee M$, where M is a finite set of $\mathcal{EL}$-concepts.

Claim 3. The second part of Theorem 9 follows if there is a family of TBoxes $\mathcal{T}_1, \mathcal{T}_2, \ldots$ such that for all $n \geq 1$, $\mathcal{T}_n$ is of size $\mathcal{O}(n^2)$, the concept name A_0 is tree-UCQ-rewritable under $\mathcal{T}_n$, but any tree-UCQ-rewriting of A_0 under $\mathcal{T}_n$ contains a concept C of depth at least 2^n.

To prove Claim 3, let $\mathcal{T}_n$ be one of the TBoxes from the claim. Let Π be a non-recursive monadic datalog rewriting of A_0 under $\mathcal{T}$. We have to show that Π has size at least 2^n. By Claim 2, we may assume that Π is connected. Let $\bigvee M$ be a tree-UCQ-rewriting of A_0 under $\mathcal{T}_n$. We may assume that each $C \in M$ is $\prec$-minimal (i.e., if $C' \prec C$, then $\mathcal{T} \not\models C' \sqsubseteq A_0$). Since $\mathcal{T}_n$ is a TBox from the claim, we find a $C \in M$ whose depth is at least 2^n. We now have $G(x_\varepsilon) \in \Pi(\mathcal{A}_C)$ for the root x_ε of the ABox $\mathcal{A}_C$ corresponding to C. Using the assumptions that Π is connected and non-recursive and that C is $\prec$-minimal with $\mathcal{T} \models C \sqsubseteq A_0$ it follows immediately that there is a sequence of distinct rules $P_0(x) \leftarrow \varphi_0, \dots, P_m(x) \leftarrow \varphi_m$ in Π with P_{i+1} in φ_i for $i < m$ and $\sum_{i=1}^{m} |\varphi_i| \geq 2^n$, as required. This finishes the proof of Claim 3.

It thus remains to identify TBoxes $\mathcal{T}_1, \mathcal{T}_2, \dots$ with the properties of Claim 3. As an abbreviation, define $C_0 := B_0$ and $C_i := B_i \sqcap \exists r.C_{i-1}$. Let $n \geq 1$. Then $\mathcal{T}_n$ consists of the following CIs, for all $i < n$:

$$
\begin{array}{rcll}
\exists r.(A_0 \sqcap B_i) & \sqsubseteq & A_0 & \\
\exists r.(A_0 \sqcap \overline{B}_i) & \sqsubseteq & A_0 & \\
B_i \sqcap \exists r.B_j & \sqsubseteq & A_0 & \text{for } 1 \leq j < n \\
 & & & \text{with } i \neq j+1 \\
\overline{B}_i \sqcap \exists r.B_j & \sqsubseteq & A_0 & \text{for } 1 \leq j < n \\
 & & & \text{with } i \neq j+1 \\
B_i \sqcap \exists r.\overline{B}_j & \sqsubseteq & A_0 & \text{for } 1 \leq j < n \\
 & & & \text{with } i \neq j+1 \\
\overline{B}_i \sqcap \exists r.\overline{B}_j & \sqsubseteq & A_0 & \text{for } 1 \leq j < n \\
 & & & \text{with } i \neq j+1 \\
B_i \sqcap \exists r^{n+1}.(B_i \sqcap \exists r.C_{i-1}) & \sqsubseteq & A_0 & \\
\overline{B}_i \sqcap \exists r^{n+1}.(\overline{B}_i \sqcap \exists r.C_{i-1}) & \sqsubseteq & A_0 & \\
B_i \sqcap \exists r^{n+1}.(\overline{B}_i \sqcap \exists r^j.\overline{B}_{i-j}) & \sqsubseteq & A_0 & \text{for } 1 \leq j < i \\
\overline{B}_i \sqcap \exists r^{n+1}.(B_i \sqcap \exists r^j.\overline{B}_{i-j}) & \sqsubseteq & A_0 & \text{for } 1 \leq j < i \\
C_{n-1} & \sqsubseteq & A_0. &
\end{array}
$$

Note that the concept names $B_0, \dots, B_{n-1}$ and $\overline{B}_0, \dots, \overline{B}_{n-1}$ are trivially FO-rewritable because they do not occur on the right-hand side of any CI. Because of the first two CIs, the concept name A_0 propagates 'backwards' along r-chains in an ABox, which in principle gives raise to non-FO-rewritability of A_0. FO-rewritability is regained, though, by enforcing the existence of a binary counter in the ABox via the concept names $B_0, \dots, B_{n-1}$ and $\overline{B}_0, \dots, \overline{B}_{n-1}$, which represent positive and negative bits, respectively. The counter is 'spaced out' in the sense that every individual on a backwards r-chain stores only a single bit of the counter. Concept inclusions 3-6 ensure that the bits appear in the right order and CIs 7-10 make sure that the counter is incremented properly, by otherwise entailing A_0, thus disrupting the unbounded propagation. The last concept inclusion entails A_0 if the counter value has reached maximum, thus stopping unbounded propagation after at most $2^n \cdot n$ steps.

A concrete tree-UCQ-rewriting for A_0 under $\mathcal{T}$ is as follows. For any concept C and $i \geq 0$, define a set of concepts $S[C]$ as follows: $S_0[C] = \{C\}$ and $S_{i+1}[C] := \{\exists r.(B_i \sqcap D), \exists r.(\overline{B}_i \sqcap D) \mid i \leq n \text{ and } D \in S_i[C]\}$. Let M be the set of all left-hand sides of concept inclusions 3-11. Now the tree UCQ-rewriting is

$$\varphi = \bigvee_{i \leq 2^{n+1}} \bigvee_{C \in M \cup \{A_0\}} \bigvee_{D \in S_i[C]} D.$$

Clearly, φ is a tree-UCQ of depth exceeding 2^n. It remains to note that every tree-UCQ-rewriting of A_0 under $\mathcal{T}$ must comprise a concept C of depth at least 2^n. Let $X_1, \dots, X_m$, $m = 2^{n+1} - n$, be the sequence of concept names from the set $\{B_0, \dots, B_{n-1}, \overline{B}_0, \dots, \overline{B}_{n-1}\}$ that represent the counter sequence $0, 1, \dots, 2^n - 1$, let $D_0 := A_0$ and $D_{i+1} := B_{i+1} \sqcap \exists r.D_i$ for all $i < 2^n - 1$. Then every tree-UCQ-rewriting of A_0 under $\mathcal{T}$ must comprise the tree-UCQ D_{2^n-1}, which is of depth 2^n. ❏