Non-Uniform Data Complexity of Query Answering in Description Logics

Carsten Lutz
Fachbereich Informatik
Universität Bremen, Germany

Frank Wolter
Department of Computer Science
University of Liverpool, UK

Abstract
In ontology-based data access (OBDA), ontologies are used as an interface for querying instance data. As the size of the data is typically much larger than the size of the ontology and query, data complexity is the most important measure of complexity in OBDA. In this paper, we propose a new method for investigating the data complexity of OBDA: instead of classifying whole logics according to their complexity, we aim at classifying each ontology within a given master language. Our results include a P/coNP-dichotomy theorem for ontologies of depth one in the description logic ALC\(^F\), the equivalence of a P/coNP-dichotomy theorem for ALC-ontologies to the non-uniform CSP dichotomy conjecture of Feder and Vardi, and a non-P/coNP-dichotomy theorem for ALC\(^F\) ontologies.

1 Introduction
In recent years, the use of ontologies to access instance data has become increasingly popular (Poggi et al. 2008; Dolby et al. 2008). The general idea is that an ontology provides a vocabulary or conceptual model for the application domain, which can then be used as an interface for querying instance data and to derive additional facts. In this emerging area, called ontology-based data access, it is a central research goal to identify ontology languages for which query answering scales to large amounts of instance data. Since the size of the data is typically very large compared to the size of the ontology and the size of the query, the central measure for such scalability is provided by data complexity—the complexity of query answering where only the data is considered to be an input, but both the query and the ontology are fixed.

In description logic (DL), ontologies take the form of a TBox, instance data is stored in an ABox, and the most important class of queries are conjunctive queries (CQs). A fundamental observation regarding this setup is that, for expressive DLs such as ALC and SHIQ, the complexity of query answering is coNP-complete (Hustadt, Motik, and Sattler 2007) and thus intractable (when speaking of complexity, we always mean data complexity). The most popular strategy to avoid this problem is to replace ALC and SHIQ with less expressive DLs that are Horn in the sense that they can be embedded into the Horn fragment of first-order (FO) logic. Horn DLs in this sense include, for example, logics from the EL and DL-Lite families as well as Horn-SHIQ, a large fragment of SHIQ for which CQ-answering is still in PTIME (Hustadt, Motik, and Sattler 2007; Eiter et al. 2008). While CQ-answering in Horn-SHIQ and the EL family of DLs is also hard for PTIME, the problem has even lower complexity in DL-Lite. In fact, the design goal of DL-Lite was to achieve FO-rewritability, i.e., that any CQ \(q\) and TBox \(T\) can be rewritten into an FO query \(q'\) such that the answers to \(q\) w.r.t. \(T\) coincide with the answers that a standard database system produces for \(q'\) (Calvanese et al. 2007). Achieving this goal requires CQ-answering to be in the circuit complexity class AC\(^0\) (Immerman 1999).

It thus seems that the data complexity of query answering in a DL context is well-understood. However, all results discussed above are on the level of logics, i.e., each result concerns a class of TBoxes that is defined syntactically through expressibility in a certain logic, but no attempt is made to identify more structure inside these classes. The aim of this paper is to advocate a fresh look on the subject, by taking a novel approach. Specifically, we advocate a non-uniform study of the complexity of query answering by considering data complexity on the level of individual TBoxes. We say that CQ-answering w.r.t. a TBox \(T\) is in PTIME if for every CQ \(q\), there is a PTIME algorithm that computes, given an ABox \(A\), the answers to \(q\) in \(A\) w.r.t. \(T\); CQ-answering w.r.t. \(T\) is coNP-hard if there exists a Boolean CQ \(q\) such that it is coNP-hard to answer \(q\) in ABoxes \(A\) w.r.t. \(T\). The goal of our approach is as follows:

Given a master DL \(L\), classify the TBoxes \(T\) in \(L\) according to the computational complexity of CQ-answering w.r.t. \(T\). If no exhaustive classification is possible, determine relevant classes of TBoxes for which a classification can be achieved.

In this paper, we consider as master DLs the basic expressive DL ALC, its extensions ALC\(^I\) with inverse roles and ALC\(^F\) with functional roles, and their union ALC\(^F\). Even for ALC, fully achieving the above goal is far beyond the scope of a single research paper. In fact, we show that a full complexity theoretic classification of ALC-TBoxes is
essentially equivalent to a full complexity theoretic classification of non-uniform constraint satisfaction problems with finite templates (CSPs). The latter is a major research programme combining complexity theory, graph theory, logic, and algebra, that is ongoing for many years, starting with Schaefer’s PTIME/np-dichotomy theorem which states that every binary CSP is in PTIME or NP-hard (Schaefer 1978). To appreciate this result, recall that Ladner’s theorem guarantees, in general, the existence of problems that are NP-intermediate and thus neither in PTIME nor NP-hard, unless PTIME = NP (Ladner 1975). Schaefer’s theorem was followed by a dichotomy result for CSPs with graph templates (Hell and Nešetřil 1990) and the seminal Feder-Vardi PTIME/np-dichotomy conjecture for all CSPs (Feder and Vardi 1993), confirmed for ternary CSPs in (Bulatov 2002).

Interesting results have also been obtained for other complexity classes such as $\text{AC}^0$ (Allender et al. 2005; Larose, Loten, and Tardif 2007). The state of the art is summarized, for example, in (Bulatov, Jeavons, and Krokhin 2005; Kun and Szegedy 2009; Bulatov 2011).

In the current paper, we mainly concentrate on characterizing the boundary between PTIME and coNP-hardness of CQ-answering w.r.t. DL TBoxes, but also establish a couple of results on FO-rewritability and $\text{AC}^0$. Our first main result states that there is a PTIME/coNP-dichotomy for CQ-answering w.r.t. $\text{ALCFI}$-TBoxes of depth one, i.e., TBoxes in which existential/universal restrictions are not nested. This case is relevant since most TBoxes from practical applications have depth one. In particular, all TBoxes formulated in DL-Lite and the extensions proposed in (Calvanese et al. 2006; Artale et al. 2009) have depth one, and the same is true for more than 85 percent of all TBoxes in the TONES repository (http://owls.cs.manchester.ac.uk/repository/).

Our second main result states that there is a PTIME/coNP-dichotomy for CQ-answering w.r.t. $\text{ALCF}$-TBoxes if and only if Feder and Vardi’s dichotomy conjecture for CSPs is true, and the same holds for $\text{ALCI}$-TBoxes. The proof of this result establishes the close link between CQ-answering in $\text{ACC}$ and CSP that was mentioned above. Arguably, linking the two worlds is even more significant for practical DL research than the dichotomy question itself. A similar link, at least in the direction from CSPs, has been established in (Calvanese et al. 2003) between view-based query answering for regular path queries and CSPs.

Finally, we show that there is no PTIME/coNP-dichotomy for CQ-answering w.r.t. $\text{ALCF}$-TBoxes (unless PTIME = NP); more specifically, we prove that for every decision problem $P$ in coNP there is an $\text{ALCF}$-TBox for which CQ-answering has the same complexity as $P$ and then apply Ladners Theorem. In contrast to the situation for $\text{ACI}$, we thus cannot expect an exhaustive classification of CQ-answering w.r.t. $\text{ALCF}$-TBoxes.

To prove our results, we introduce two new notions that are of independent interest and general utility. The first one is materializability of a TBox $\mathcal{T}$, which means that answering a CQ over an ABox $\mathcal{A}$ w.r.t. $\mathcal{T}$ can be reduced to query evaluation in a single model of $\mathcal{A}$ and $\mathcal{T}$. Note that such models play a crucial role in the context of Horn DLs, where they are often called least models or canonical models. For TBoxes of depth one, materializability precisely characterizes PTIME CQ-answering, which allows us to establish the first main result. For TBoxes of unrestricted depth, non-materializability still provides a sufficient condition for coNP-hardness of CQ-answering. The second notion is unraveling tolerance of a TBox $\mathcal{T}$, which provides a sufficient condition for query answering to be in PTIME. This yields a uniform and rather general PTIME upper bound for CQ-answering w.r.t. many $\text{ALCFI}$-TBoxes which strictly generalizes the known result that CQ-answering in Horn-$\text{ALCFI}$ is in PTIME. In CSP, unraveling tolerance corresponds to the existence of tree obstructions, a notion that characterizes the well-known consistency condition (Krokhin 2010; Dechter 2003).

Our framework also allows to formally establish some common intuitions and beliefs held in the context of CQ-answering in DLs. For example, we show that for any $\text{ALCFI}$-TBox $\mathcal{T}$, CQ-answering is in PTIME if and only if answering positive existential queries is in PTIME and then apply mapping tolerance to an element. We make the following tolerance property and important for Horn DLs.

Most proofs are deferred to the long version, available at http://www.csc.liv.ac.uk/~frank/publ/publ.html.

2 Preliminaries

We start with introducing the DL $\text{ALC}$ and its extensions $\text{ALCI}$ and $\text{ALCFI}$ studied in this paper. $\text{ALC}$-concepts are constructed according to the rule

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

where $A$ ranges over concept names from a countably infinite set $\mathbb{N}_0$, $r$ ranges over role names from a countably infinite set $\mathbb{N}_R$. As usual, we use $\forall r.C$ as an abbreviation for $\lnot \exists r.\lnot C$. $\text{ALCI}$-concepts admit, in addition, inverse roles from $\mathbb{N}_{R}^{-} = \{ r^{-} \mid r \in \mathbb{N}_R \}$. We set $r^{-} = s$ if $r = s^\circ$ for a role name $s$. An $\text{ALCF}$-TBox is a finite set of concept inclusions (CIs) $C \sqsubseteq D$, where $C, D$ are $\text{ALC}$-concepts, and likewise for $\text{ALCI}$-TBoxes. An $\text{ALCFI}$-TBox is an $\text{ALCI}$ TBox that additionally admits functionality assertions $\text{func}(r)$, where $r \in \mathbb{N}_R \cup \mathbb{N}_{R}^{-}$.

An $\text{ABox}$ $\mathcal{A}$ is a finite 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 $\mathbb{N}_I$. In some cases, we drop the finiteness condition on ABoxes; then we explicitly speak about infinite ABoxes. We use $\text{Ind}(\mathcal{A})$ to denote the set of individual names used in the ABox $\mathcal{A}$ and sometimes write $r^{-}(a, b) \in \mathcal{A}$ instead of $r(b, a) \in \mathcal{A}$.

The semantics of DLs is given by interpretations $\mathcal{I} = (\Delta^C, \cdot^C)$, where $\Delta^C$ is a non-empty set and $\cdot^C$ maps each concept name $A \in \mathbb{N}_0$ to a subset $\Delta^C_A \subseteq \Delta^C$, each role name $r \in \mathbb{N}_R$ to a binary relation $r^\mathcal{I}$ on $\Delta^C$, and each individual name $a$ to an element. We make the unique name assumption, i.e., $a^\mathcal{I} \neq b^\mathcal{I}$ whenever $a \neq b$. The extension $\Delta^C_C \subseteq \Delta^C$ of a concept $C$ under the interpretation $\mathcal{I}$ is defined as usual.
see (Baader et al. 2003). For the purposes of this paper, it is often convenient to work with interpretations that interpret some individual names, but not all. In this case, we use \( \text{Ind}(I) \) to denote the set of individual names interpreted by \( I \). We say that \( I \) satisfies a CI \( C \subseteq D \) if \( C^I \subseteq D^I \), an assertion \( A(a) \) if \( a \in \text{Ind}(I) \) and \( a^I \in C^I \), an assertion \( r(a, b) \) if \( a, b \in \text{Ind}(I) \) and \( (a^I, b^I) \in r^I \), and a functionality assertion \( \text{func}(r) \) if \( r^I \) is a function. Finally, \( I \) is a model of a TBox \( T \) (ABox, A) if it satisfies all inclusions in \( T \) (all assertions in \( A \)). The class of all models of \( T \) and \( A \) is denoted by \( \text{Mod}(T, A) \). We call an ABox \( A \) consistent w.r.t. a TBox \( T \) if \( \text{Mod}(T, A) \neq \emptyset \).

Throughout this paper, we consider various query languages which can all be seen as fragments of first-order logic. A first-order query (FOQ) \( q(\vec{x}) \) is a first-order formula with free variables \( \vec{x} \) constructed from atoms \( A(t) \), \( r(t, t') \), and \( t = t' \) (where \( A \in N_C \), \( r \in N_R \), and \( t, t' \) range over individual names and variables) using negation, conjunction, disjunction, and existential and universal quantification. The variables in \( \vec{x} \) are the answer variables of \( q \). A FOQ without answer variables is Boolean. We say that a tuple \( \vec{a} \subseteq \text{Ind}(A) \) is an answer to \( q(\vec{x}) \) in an interpretation \( I \) if \( I \models q(\vec{a}) \), where \( q(\vec{a}) \) results from replacing the answer variables \( \vec{x} \) in \( q(\vec{x}) \) with \( \vec{a} \). A tuple \( \vec{a} \subseteq \text{Ind}(A) \) of the same arity as \( \vec{x} \) is a certain answer to \( q(\vec{x}) \) in \( A \) given \( T \), in symbols \( (T, A) \models q(\vec{a}) \), if \( I \models q(\vec{a}) \) for all \( I \in \text{Mod}(T, A) \). Set \( \text{cert}_T(q, A) = \{ \vec{a} \mid (T, A) \models q(\vec{a}) \} \). Note that for Boolean queries \( q \), \( \text{cert}_T(q, A) = \{ \} \) if \( (T, A) \models q \) and \( \text{cert}_T(q, A) = \{ \vec{a} \} \) if \( (T, A) \not\models q \). Thus, computing the set \( \text{cert}_T(q, A) \) is equivalent to deciding \( (T, A) \models q \).

**Example 1.** (1) Let \( T = \{ \exists r . A \subseteq A \} \) and \( q_0(x) = A(x) \). For any ABox \( A \), \( \text{cert}_{T_0}(q_0, A) \) is the set of all \( a \in \text{Ind}(A) \) such that there is an \( r \)-path in \( A \) from \( a \) to some \( b \) with \( A(b) \in A \) (i.e., there are \( n \geq 1 \) and \( r(a_i, a_{i+1}) \in A \), \( 1 \leq i < n \), with \( a_0 = a \), \( a_n = b \), and \( A(b) \in A \)).

(2) We express \( k \)-colorability using a Boolean query. Consider an undirected graph represented as an ABox \( A \) with assertions \( r(a, b), r(b, a) \in A \) iff there is an edge between \( a \) and \( b \). Let \( A_1, \ldots, A_k, M \) be fresh concept names. Then \( A \) is \( k \)-colorable iff \( (T_k, A) \not\models \exists x . M(x) \), where \( T_k = \{ A_i \cap A_j \subseteq \bot \mid 1 \leq i < j \leq k \} \cup \{ A_i \cap \exists r . A_i \subseteq M \mid 1 \leq i \leq k \} \cup \{ \top \subseteq \bigcup_{1 \leq i \leq k} A_i \} \).

In the DL context, answers to FOQs are not always computable and therefore weaker query languages are used. We consider positive existential queries (PEQs), i.e., FOQs without negation, universal quantification, and equality and conjunctive queries (CQs) (i.e., PEQs without disjunction) as well as two weaker languages whose introduction requires some preliminaries. Recall that E\( L \)-concepts are constructed from \( N_C \) and \( N_R \) using conjunction, existential restriction, and the concept \( T \) (Baader, Brandt, and Lutz 2005). E\( L \)-concepts additionally admit inverse roles. If \( C \) is an E\( L \)-concept and \( a \in N_i \), then \( C(a) \) is called an E\( LI \)-query (ELIQ); if \( C \) is an E\( L \)-concept, then \( C(a) \) is called an E\( L \)-query (ELQ). Note that every ELIQ (and, therefore, every ELQ) can be regarded as an (acyclic) Boolean CQ. For example, the ELIQ \( \exists r . (A \cap \exists s . B)(a) \) is equivalent to the Boolean CQ \( \exists x . (r(a, x) \land A(x) \land s(y, x) \land B(y)) \).

In what follows, we will not distinguish between an ELIQ and its translation into a Boolean CQ and freely apply notions introduced for FOQs also to ELIQs and ELQs.

For an ABox \( A \), we denote by \( \text{Ind}(A) \) the interpretation with \( A^A = \text{Ind}(A) \), \( \forall A \subseteq \text{Ind}(A) \), and \( A \models A \), \( \{ \{ a \} \subseteq \text{Ind}(A) \} \subseteq \{ \{ a, b \} \subseteq \text{Ind}(A) \} \) for any \( A \in N_C \) and \( r \in N_R \). Note that \( \text{Ind}(I) = \text{Ind}(A) \).

In what follows, we sometimes slightly abuse notation and use FOQ to denote the set of all first-order queries, and likewise for CQ, PEQ, ELIQ, and ELQ. We now introduce the main notions investigated in this paper.

**Definition 2.** Let \( T \) be an A\( C\)F\( LI \)-TBox and let \( Q \in \{ \text{CQ, PEQ, ELIQ, ELQ} \} \). Then

- **Q-answering w.r.t. \( T \) is in PTIME** if for every \( q(\vec{x}) \in Q \), there is a polytime algorithm that computes, given an ABox \( A \), the answer \( \text{cert}_T(q, A) \);
- **Q-answering w.r.t. \( T \) is coNP-hard** if there is a Boolean \( q \in Q \) such that, given an ABox \( A \), it is coNP-hard to decide whether \( (T, A) \models q \);
- **\( T \) is FO-re-writable for \( Q \) iff** for every Boolean \( q \in Q \), one can effectively construct a FOQ \( q'(\vec{x}) \) such that for every ABox \( A \), \( \text{cert}_T(q, A) = \{ \vec{a} \mid I_A \models q'(\vec{a}) \} \).

Note that Q-answering w.r.t. \( T \) is in PTIME iff for every Boolean query \( q \in Q \), there is a polytime algorithm deciding, given an ABox \( A \), whether \( (T, A) \models q \). If \( T \) is FO-re-writable, then Q-answering w.r.t. \( T \) is in PTIME (even in AC\( ^0 \)) as this is the data complexity of evaluating a FOQ in a single interpretation. We give some examples that illustrate the above notions.

**Example 3.** (1) CQ-answering w.r.t. \( T_r \) from Example 1 is in PTIME since for any ABox \( A \), \( \text{cert}_{T_r}(q, A) \) can be computed as follows. Let \( A' \) by the ABox obtained from \( A \) by adding \( A(a) \) to \( A \) if there is an \( r \)-path from \( a \) to some \( b \) with \( A(b) \in A \). Then \( \text{cert}_{T_r}(q, A) = \{ \vec{a} \mid I_{A'} \models q(\vec{a}) \} \) which can be computed in AC\( ^0 \) by evaluating a FOQ in a single interpretation.

(2) On the other hand, it follows from the fact that transitive closure cannot be expressed in FO that \( T_r \) is not FO-re-writable for CQ (or even ELQ).

(3) Consider the TBoxes \( T_k \) expressing \( k \)-colorability from Example 3. Clearly, for \( k \geq 3 \), CQ-answering w.r.t. \( T_k \) is coNP-hard since \( k \)-colorability is NP-hard. However, in contrast to the tractability of 2-colorability, CQ-answering w.r.t. \( T_2 \) is coNP-hard as well (even ELQ-answering is). This follows from Theorem 11 below and, intuitively, is the case because \( T_2 \) entails a disjunction: there is an ABox \( A = \{ B(a) \} \) such that \( (T_2, A) \models A_1(a) \lor A_2(a) \), but neither \( (T_2, A) \models A_1(a) \) nor \( (T_2, A) \models A_2(a) \).

The above notions of complexity are rather robust under changing the query language: neither PTIME upper bounds nor FO-re-writability depend on whether we consider PEQs, CQs, or ELIQs.

**Theorem 4.** For all A\( C\)F\( LI \)-TBoxes \( T \), the following equivalences hold:
1. **CQ-answering w.r.t.** $T$ is in PTIME iff PEQ-answering w.r.t. $T$ is in PTIME iff ELIQ-answering w.r.t. $T$ is in PTIME.

2. **$T$ is FO-rewritable for CQ iff it is FO-rewritable for PEQ iff it is FO-rewritable for ELIQ** (if PTIME $\neq$ coNP).

If $T$ is an ALCFI-TBox, then we can replace ELIQ in Points 1 and 2 with ELQ.

The proof is based on Theorems 9 and 11 below. Theorem 4 gives an explanation for the fact that, in the logic-based setting, the data complexity of answering PEQs, CQs, and ELIQs has turned out to be identical for many DLs. It allows us to (sometimes) speak of the ‘complexity of query answering’ without reference to a query language.

### 3 Materializability

An important tool for analyzing the complexity of query answering is the notion of materializability of a TBox $T$, which means that computing $\text{cert}_T(q, A)$ can be reduced to computing the answers to $q$ in a single model of $T$ and $A$. Our central result is that non-materializability of a TBox is a sufficient condition for query answering being coNP-hard.

**Definition 5.** Let $T$ be an ALCFI-TBox, $A$ an ABox, and $Q \in \{\text{CQ, PEQ, ELIQ, ELQ}\}$. A model $I$ of $T$ and $A$ is called a Q-materialization of $T$ and $A$ if for all queries $q(T, A)$ and potential answers $\bar{a} \subseteq \text{Ind}(A)$, we have $I \models q(\bar{a})$ iff $T, A \models q(\bar{a})$.

The TBox $T$ is Q-materializable if for every ABox $A$ that is consistent w.r.t. $T$, there is a Q-materialization of $T$ and $A$.

In Example 3 (1), the interpretation $I_A$ is a PEQ-materialization of $T_r$ and $A$. The notion of Q-materialization can be viewed as a generalization of the various kinds of canonical models that are often used to prove results for query answering in ‘Horn DLs’ such as $\mathcal{EL}$ and DL-Lite (Lutz, Toman, and Wolter 2009; Kontchakov et al. 2010). In fact, the ELQ-materialization in the next example is exactly the ‘compact canonical model’ from (Lutz, Toman, and Wolter 2009).

**Example 6.** Let $T = \{A \subseteq \exists r. A\}$ and $A$ be an ABox with $X = \{A(a) \mid A(a) \in A\} \neq \emptyset$. Let $d_r$ be a fresh individual and obtain $J^A$ from $I_A$ by setting $A^{J_A} = A^{I_A} \cup \{d_r\}$ and $r^{J_A} = r^{I_A} \cup \{(a, d_r) \mid A(a) \in X\} \cup \{(d_r, d_r)\}$.

Then $J_A$ is an ELQ-materialization of $T$ and $A$.

Clearly, a PEQ-materialization is a CQ-materialization if it is an ELIQ-materialization is an ELQ-materialization. The converse holds for the CQ-PEQ case only, as shown later on. The following example demonstrates that ELQ-materializations are different from ELIQ-materialization. A similar argument separates ELIQ-materializations from CQ-materializations.

**Example 7.** Let $T$ and $A$ be as in Example 6, and $J_A$ the ELQ-materialization constructed in that example. Then $J_A$ is not a Q-materialization for any $Q \in \{\text{ELIQ, CQ, PEQ}\}$. For example, for

$$A_0 = \{B_1(a), B_2(b), A(a), A(b)\}\text{ and } q = (B_1 \cap \exists r. \exists r'. B_2)(a),$$

we have $J_{A_0} \models q$, but $(T, A_0) \not\models q$. An ELIQ/PEQ-materialization of $T$ and $A$ is obtained by unfolding $J_A$: instead of using only one additional individual $d_r$ as a witness for $\exists r. A$, one attaches to each $a \in X$ an infinite $r$-path of $A$-nodes. Observe that every CQ/PEQ-materialization of $T_r$ and $A_0$ must be infinite.

Before we link materializations to complexity, we characterize them semantically in terms of simulations and homomorphisms. This is interesting in its own right and establishes a close connection between materialization and initial models as studied in model theory, algebraic specification, and logic programming (Malcev 1971; Meseguer and Goguen 1985; Makowsky 1987).

A simulation from an interpretation $I_1$ to an interpretation $I_2$ is a relation $S \subseteq \Delta^2 \times \Delta^2$ such that the domain of $S$ contains $\{a^2 \mid a \in \text{Ind}(I_1)\}$ and

1. for all $A \in N_C$: if $(d_1, d_2) \in S$, then $d_2 \in A^{\Delta_2}$;
2. for all $r \in N_R$: if $(d_1, d_2) \in S$ and $(d_1, d'_1) \in r^{\Delta_1}$, then there exists $d'_2 \in A^{\Delta_2}$ such that $(d'_1, d'_2) \in r$ and $(d_2, d'_2) \in r^{\Delta_2}$;
3. for all $a \in \text{Ind}(I_1)$: $a \in \text{Ind}(I_2)$ and $(a^{\Delta_2}, a^{\Delta_2}) \in S$.

We call $S$ an i-simulation if Condition 2 is satisfied also for inverse roles and a homomorphism if $S$ is a function. An interpretation $I$ is called hom-initial (sim-initial, i-sim-initial) in a class $\mathcal{K}$ of interpretations if for every $J \in \mathcal{K}$, there exists a homomorphism (a simulation, an i-simulation) from $I$ to $J$.

An interpretation $I$ is generated if every $d \in \Delta^2$ is reachable from some $a^2$, $a \in \text{Ind}(I)$, in the undirected graph $(\Delta^2, \{(d, d') \mid (d, d') \in \bigcup_{r \in N_R} r^{\Delta_2}\})$.

**Lemma 8.** Let $T$ be an ALCFI-TBox, $A$ an ABox, and $I \in \text{Mod}(T, A)$. Then

1. $I$ is an ELIQ-materialization of $T$ and $A$ iff it is i-sim-initial in $\text{Mod}(I, A)$;
2. if $I$ is countable and generated, then it is a CQ-materialization of $T$ and $A$ iff it is a PEQ-materialization of $T$ and $A$ iff it is hom-initial in $\text{Mod}(I, A)$.

If $T$ is an ALCFI-TBox, then $I$ is an ELQ-materialization of $T$ and $A$ iff it is sim-initial in $\text{Mod}(I, A)$.

**Proof.** (Sketch) The proofs of “$\implies$” are straightforward since matches of PEQs, CQs, and ELIQs are preserved under i-simulations and homomorphisms, and matches of ELQs are preserved under simulations. We thus concentrate on “$\impliedby$”.

(1) Assume $I$ is an ELIQ-materialization and let $J \in \text{Mod}(T, A)$. If $J$ has finite outdegree, an i-simulation from $I$ to $J$ can be constructed in the same way as in standard proofs showing that simulations characterize the expressive power of $\mathcal{EL}$-concepts (Lutz, Piro, and Wolter 2011). If $J$ has infinite outdegree, then one can construct a selective unfolding $J^* \in \text{Mod}(T, A)$ of $J$ whose outdegree is finite and such that there is a homomorphism from $J^*$ to $J$. It remains to compose an i-simulation from $I$ to $J^*$ with the homomorphism from $J^*$ to $J$. 


For (2), we show that any countable and generated CQ-materialization is hom-initial. If \( I \) is such a CQ-materialization and \( \mathcal{J} \in \text{Mod}(T,A) \), then by the semantics of CQs we can find a homomorphism from any finite subinterpretation of \( I \) to \( \mathcal{J} \). If \( \mathcal{J} \) is of finite outdegree, we can assemble all those homomorphisms into a homomorphism from \( I \) to \( \mathcal{J} \) in a direct way (using that \( I \) is countable and generated). For \( \mathcal{J} \) of non-finite outdegree, we compose the homomorphism from \( I \) to \( \mathcal{J}^{*} \) with the homomorphism from \( \mathcal{J}^{*} \) to \( \mathcal{J} \), with \( \mathcal{J}^{*} \) constructed as in (1). The claim for \( \text{ALCF-TBoxes} \) is proved similarly to (1).

The technical condition of being generated in Point 2 of Lemma 8 cannot be dropped without losing correctness (see long version). It is open whether countability can be dropped.

Despite the fact that materializations may look different for different query languages, we show that materializability coincides for PEQs, CQs, and ELIQs.

**Theorem 9.** Let \( T \) be an \( \text{ALCF-TBox} \). Then
1. \( T \) is PEQ-materializable iff \( T \) is CQ-materializable iff \( T \) is ELIQ-materializable;
2. the above is the case iff for every \( A \in \text{Mod}(T,A) \) contains a hom-initial \( I \) iff for every \( A \in \text{Mod}(T,A) \) contains an i-sim-initial \( I \).

If \( T \) is an \( \text{ALCF-TBox} \), the above are equivalent to ELQ-materializability and for every \( A \in \text{Mod}(T,A) \) contains a sim-initial \( I \).

This theorem is essentially a consequence of Lemma 8. The proof of “\( \leq \)” in Point 2 requires a selective unfolding technique similar to the one used in the proof of Lemma 8 to transform an i-simulation into a homomorphism. Also due to this technique, the conditions of generativeness and countability from Lemma 8 can be avoided in Theorem 9.

Because of Theorem 9, we sometimes speak of materializability without reference to a query language and of materializations instead of PEQ-materializations. Interestingly, materializability is equivalent to another notion that is well-known in the context of Horn DLs, called convexity or the disjunction property and closely related to PTIME reasoning (Baader, Brandt, and Lutz 2005). Specifically, a TBox \( T \) has the ABox disjunction property if for all ABoxes \( A \) and ELIQs \( C_{1}(a_{1}),\ldots,C_{n}(a_{n}) \), it follows from \( (T,A) \models C_{1}(a_{1}) \lor \ldots \lor C_{n}(a_{n}) \) that \( (T,A) \models C_{i}(a_{i}) \) for some \( i \leq n \).

**Theorem 10.** An \( \text{ALCF-TBox} \) \( T \) is materializable iff it has the disjunction property.

**Proof.** For the nontrivial “\( \leq \)” direction, let \( A \) be an ABox that is consistent w.r.t. \( T \) and such that there is no ELIQ-materialization of \( T \) and \( A \). Then \( \mathcal{T} \cup A \cup \Gamma \) is not satisfiable, where

\[
\Gamma = \{ \neg C(a) \mid (T,A) \not\models C(a),a \in \text{Ind}(A),C(a) \text{ ELIQ} \}.
\]

In fact, any satisfying interpretation would be an ELIQ-materialization. By compactness, there is a finite subset \( \Gamma' \) of \( \Gamma \) such that \( \mathcal{T} \cup A \cup \Gamma' \) is not satisfiable, i.e. \( (T,A) \not\models \bigvee \neg C(a) \in \Gamma' \), \( C(a) \). By definition of \( \Gamma' \), \( (T,A) \not\models C(a) \), for all \( C(a) \in \Gamma' \). Thus, \( T \) lacks the ABox disjunction property.

Based on Theorems 9 and 10, we now establish the announced link between materializability and complexity.

**Theorem 11.** If an \( \text{ALCF-TBox} \) \( T \) is not materializable, then ELIQ-answering (ELQ-answering) is \( \text{coNP-hard} \) w.r.t. \( T \).

The proof exploits failure of the ABox disjunction property to generalize the reduction of 24+2-SAT used in (Schaerf 1993) to prove that ELQ-answering in a variant of \( \mathcal{EL} \) is \( \text{coNP-hard} \).

The converse of Theorem 11 fails, i.e., materializability is not a sufficient condition for query answering to be in PTIME (unless PTIME \( = \) NP). Such TBoxes have a materialization for every ABox, but these materializations cannot always be computed in PTIME. In fact, we show later that for any non-uniform CSP, there is a materializable \( \text{ALCF-TBox} \) for which Boolean CQ-answering has the same complexity, up to complementation of the complexity class.

Theorem 11 also allows us to prove Theorem 4.

**Proof of Theorem 4 (sketch).** By Theorem 11, it is sufficient to consider materializable TBoxes when proving Theorem 4. To show, for example, that if CQ-answering w.r.t. \( T \) is in PTIME then PEQ-answering w.r.t. \( T \) is in PTIME, one can first transform a PEQ \( q(\bar{x}) \) into an equivalent union \( \bigcup_{\bar{e} \in \bar{E}} q_{\bar{e}}(\bar{x}) \) of CQs \( q_{\bar{e}}(\bar{x}) \); then, CQ-materializability of \( T \) implies that, for any ABox \( A \), \( \text{cert}_{\mathcal{T}}(q,A) = \bigcup_{\bar{e} \in \bar{E}} \text{cert}_{\mathcal{T}}(q_{\bar{e}},A) \), and the sets \( \text{cert}_{\mathcal{T}}(q_{\bar{e}},A) \) can be computed in PTIME. The remaining reductions are more involved, but based on the same idea.

## 4 Unraveling Tolerance

We develop a condition on TBoxes, called unraveling tolerance, that is sufficient for PTIME CQ-anwering and strictly generalizes syntactic ‘Horn conditions’ such as the ones used to define the well-known DL Horn-\( SHIQ \) which was designed as a maximal DL with PTIME query answering (Hustadt, Motik, and Sattler 2007). Unraveling tolerance is based on an unraveling operation on ABoxes, in the same spirit as the well-known unraveling of an interpretation into a tree interpretation. More precisely, the unraveling \( A^n \) of an ABox \( A \) is the following (possibly infinite) ABox:

- Ind(\( A^n \)) is the set of sequences \( b_{0}r_{0}b_{1}\ldots r_{n-1}b_{n} \) with \( b_{0},\ldots,b_{n} \in \text{Ind}(A) \) and \( r_{0},\ldots,r_{n-1} \in \mathbb{N}_{R} \cup \mathbb{N}_{R}^{\circ} \) such that for all \( i < n \), we have \( r_{i}(b_{i},b_{i+1}) \in A \) and \( (b_{i-1},r_{i-1}^{\setminus}) \neq (b_{i+1},r_{i}) \) when \( i > 0 \);
- for each \( C(b) \in A \) and \( \alpha = b_{0}\cdot\ldots\cdot b_{n} \in \text{Ind}(A^{n}) \) with \( b_{n} = b \), we have \( C(\alpha) \in A^{n} \);
- for each \( \alpha = b_{0}r_{0}\cdot\ldots\cdot r_{n-1}b_{n} \in \text{Ind}(A^{n}) \) with \( n > 0 \), we have \( r_{n-1}(b_{0}\cdot\ldots\cdot b_{n-1},\alpha) \in A^{n} \).

For all \( \alpha = b_{0}\cdot\ldots\cdot b_{n} \in \text{Ind}(A^{n}) \), we write tail(\( \alpha \)) to denote \( b_{n} \). Note that the condition \( (b_{i-1},r_{i-1}^{\setminus}) \neq (b_{i},r_{i}) \) is needed to ensure that functional roles can still be interpreted in a functional way after unraveling.

**Definition 12.** A TBox \( T \) is unraveling tolerant if for all ABoxes \( A \) and ELIQs \( q \), we have that \( (T,A) \models q \) implies \( (T,A^{n}) \models q \).
It is not hard to prove that the converse direction \( (\mathcal{T}, A^u) \models q \) implies \( (\mathcal{T}, A) \models q' \) is true for all ALC\textcircled{FL}-TBoxes.

**Example 13.** (i) The ALC-Box \( \mathcal{T}_1 = \{ A \subseteq \forall r.B \} \) is unraveling tolerant. This can be proved by showing that (i) for any (finite or infinite) ABox \( \mathcal{A} \), the interpretation \( \mathcal{I}_A^+ \) that is obtained from \( \mathcal{I}_A \) by setting \( B_{\mathcal{I}_A}^+ = B_{\mathcal{I}_A} \cup (\exists r.\neg A)_{\mathcal{I}_A} \) is an ELIQ-materialization of \( \mathcal{T}_1 \) and \( \mathcal{A} \); and (ii) \( \mathcal{I}_A^+ \models C(a) \) iff \( \mathcal{I}_A \models C(a) \) for all ELIQs \( C(a) \). The proof of (ii) is based on a simple induction on the structure of the ELIQ-concept \( C. \) As witnessed by the ABox \( \mathcal{A} = \{ r(a, a) \} \) and ELIQ \( B(b) \), the use of inverse roles in the definition of \( A^u \) is crucial here despite the fact that \( \mathcal{T}_1 \) does not use inverse roles.

(ii) A simple example for an ALC-Box that is not unraveling tolerant is \( \mathcal{T}_2 = \{ A \land \exists r. A \subseteq B, \neg A \land \exists r. \neg A \subseteq B \} \). For \( A = \{ r(a, a) \} \), it is easy to see that we have \( (\mathcal{T}_2, A) \models B(a) \) (use a case distinction on the truth value of \( A \) at \( a \)), but \( (\mathcal{T}_2, A^u) \not\models B(a) \).

Before we show that unraveling tolerance indeed implies PTIME query answering, we first demonstrate the generality of this property by relating it to Horn-ALCF\textcircled{I}, the ALC\textcircled{FL}-fragment of Horn-SH\textcircled{IQ}. Different versions of Horn-SH\textcircled{IQ} have been proposed in the literature, giving rise to different versions of Horn-ALCF\textcircled{I} (Eiter et al. 2008; Kazakov 2009). As the original definition from (Hustadt, Motik, and Sattler 2007) based on polarity is rather technical, we prefer to work with the following, more direct definition. A Horn-ALCF\textcircled{I}-Box has the form \( \mathcal{T} = \{ T \subseteq C_T \} \cup \mathcal{F} \), where \( \mathcal{F} \) is a set of functionality assertions and \( C_T \) is built according to the topic rule in\( R, R' ::= \top \mid \bot \mid A \mid \neg A \mid R \land R' \mid L \rightarrow R \mid \forall r. R \mid \forall r. R \mid L \mid L' \mid L \cup L' \mid \exists r. L \)

where \( r \) ranges over \( N_R \cup N_R^- \). By applying some simple transformations, it is not hard to show that every Horn-ALCF\textcircled{I}-Box according to the original, polarity-based definition is equivalent to a Horn-ALCF\textcircled{I}-Box of the form introduced here. Although not important in our context, we note that even a polytime transformation is possible.

**Theorem 14.**

**Every Horn-ALCF\textcircled{I}-Box is unraveling tolerant.**

**Proof.** (hint) Based on a generalization of the argument in Example 13, where the ad hoc materialization \( \mathcal{I}_A^+ \) is replaced by a systematically constructed canonical model \( \mathcal{I}_{\mathcal{T}, \mathcal{A}}^+ \) of \( \mathcal{T} \) and \( \mathcal{A} \). □

Theorem 14 shows that unraveling tolerance and Horn logic are closely related. Yet, the next example shows that there are unraveling tolerant ALC\textcircled{FL}-TBoxes that are not equivalent to any Horn sentence of FO. Since any Horn-ALCF\textcircled{I}-Box is equivalent to such a sentence, it follows that unraveling tolerant ALC\textcircled{FL}-TBoxes strictly generalize Horn-ALCF\textcircled{I}-TBoxes. This increased generality will pay off in Section 5 when we establish a dichotomy result for TBoxes of depth one.

**Example 15.** Take the ALC-Box \( \mathcal{T} = \{ \exists r. (A \land \neg B_1 \land \neg B_2) \subseteq \exists r. (\neg A \land \neg B_1 \land \neg B_2) \}\). One can show as in Example 13 that \( \mathcal{T} \) is unraveling tolerant; here, the materialization is actually \( \mathcal{I}_A^+ \) itself instead of some \( \mathcal{I}_A^+ \), i.e., as far as ELIQ (and even PEQ) answering is concerned, \( \mathcal{T} \) cannot be distinguished from the empty TBox.

It is well-known that FO Horn sentences are preserved under direct products (Chang and Keisler 1990). To show that \( \mathcal{T} \) is not equivalent to any such sentence, it thus suffices to show that \( \mathcal{T} \) is not preserved under direct products. This is simple: let \( \mathcal{I}_1 \) and \( \mathcal{I}_2 \) consist of a single \( r \)-edge between elements \( d \) and \( e \), and let \( \epsilon \in (A \land B_1 \land \neg B_2)^2 \) and \( \epsilon \in (A \land \neg B_1 \land B_2)^2 \); then the direct product \( \mathcal{I} \) of \( \mathcal{I}_1 \) and \( \mathcal{I}_2 \) still has the \( r \)-edge between \( (d, d) \) and \( (e, e) \) and satisfies \( (e, e) \in (A \land \neg B_1 \land \neg B_2)^2 \), thus is not a model of \( \mathcal{T} \).

We now establish the PTIME upper bound for unraveling tolerant TBoxes.

**Theorem 16.** If an ALC\textcircled{FL}-Box \( \mathcal{T} \) is unraveling tolerant, then PEQ-anwering w.r.t. \( \mathcal{T} \) is in PTIME.

**Proof.** (sketch) Let \( \mathcal{T} \) be unraveling tolerant. By Theorem 4, it suffices to show that ELIQ-anwering w.r.t. \( \mathcal{T} \) is in PTIME. Let \( \mathcal{A} \) be an ABox and \( q = C_0(a_0) \) an ELIQ. Let \( cl(T, C_0) \) denote the closure under single negation of the set of subconcepts of \( \mathcal{T} \) and \( C_0 \). \( tp(T, C_0) \) denotes the set of all types (aka Hintikka sets or maximal consistent sets) over \( cl(T, C_0) \). A type assignment is a map \( \text{Ind}(\mathcal{A}) \rightarrow 2^{tp(T,q)} \).

The PTIME algorithm for checking whether \( (\mathcal{T}, \mathcal{A}) \models q \) is based on the computation of a sequence of type assignments \( \pi_0, \pi_1, \ldots \) as follows. For every \( a \in \text{Ind}(\mathcal{A}) \), \( \pi_0(a) \) is the set of types \( t \in tp(T,q) \) such that \( A(a) \) implies \( A \subseteq t \). Then, \( \pi_{i+1}(a) \) is defined as the set of types \( t_a \in \pi_i(a) \) such that for all \( r(a, b) \in A, r \) a role name or the inverse thereof, there is a type \( t_b \in \pi_i(b) \) such that \( t_a \sim_r t_b \), where we write \( t_a \sim_r t_b \) if the following conditions are satisfied: if \( C \in t_b \) then \( \exists r. C \in t_a \) for all \( \exists r. C \in \text{cl}(T,C_0) \), if \( C \in t_a \) then \( \exists r.\neg C \in t_b \) for all \( \exists r.\neg C \in \text{cl}(T,C_0) \) if \( C \in t_b \) and \( r \notin C \in \text{cl}(T,C_0) \) with \( \text{func}(r) \in T \); if \( \exists r.\neg C \in t_a \) then \( C \in t_a \), for all \( \exists r.\neg C \in \text{cl}(T,C_0) \) with \( \text{func}(r) \in T \).

Clearly, the sequence \( \pi_0, \pi_1, \ldots \) stabilizes after at most \( |\mathcal{O}(\mathcal{A})| \) steps and can be computed in time polynomial in \( |\mathcal{A}| \) (since \( |\mathcal{T}| \) and thus \( |tp(T,q)| \) is a constant). Let \( \pi \) be the final type assignment in the sequence. In the long version, we show that \( (\mathcal{T}, \mathcal{A}) \models q \) iff \( C_0 \in t \) for all \( t \in \pi(a_0) \). □

By Lemma 14 and since we actually exhibit a uniform algorithm for query answering w.r.t. unraveling tolerant TBoxes, Theorem 16 also reproves the known PTIME upper bound for CQ-answering in Horn-ALCF\textcircled{I} (Eiter et al. 2008).

By Theorems 11 and 16, unraveling tolerance implies materializability unless PTIME = NP. We note that, based on the disjunction property this implication can also be proved without the side condition.

**Lemma 17.** Every unraveling tolerant ALC\textcircled{FL}-Box is materializable.

The converse of Lemma 17 and, more generally, of Theorem 16 fails. In fact, while unraveling tolerance is a sufficient condition for PTIME query answering, it is not a necessary one. An example is given in Section 6, where the
TBox $T_{C_2}$ associated with the CSP representation of the 2-colorability problem has PTIME query answering but is not unraveling tolerant.

The PTIME algorithm in Theorem 16 resembles the standard arc consistency algorithm for CSPs (Dechter 2003). This link to CSPs can be formalized for $ALCI$-TBoxes using the templates $I_{\tau, q}$ constructed in the proof of Theorems 22 and 24 below: it is known that a CSP can be solved using arc consistency iff it has tree obstructions (Krokhin 2010). Also, one can show that an $ALCI$-TBox $T$ is unraveling tolerant iff all templates $I_{\tau, q}$ from Theorem 24 have tree obstructions. Consequently, for any $ALCI$-TBox $T$, ELIQs can be answered using an arc consistency algorithm iff $T$ is unraveling tolerant.

There are generalizations of both arc consistency and tree obstructions (Bulatov, Krokhin, and Larose 2008) and it should be possible to accordingly generalize Theorem 16 to larger classes of TBoxes by relaxing the property of ABox unraveling such that it produces, for example, ABoxes of bounded treewidth.

5 Dichotomy for Depth One

We establish a dichotomy between PTIME and coNP for TBoxes of depth one, i.e., sets of CIs $C \subseteq D$ such that the maximum nesting depth of the constructors $\exists r. E$ and $\forall r. E$ in $C$ and $D$ is one. All examples given in the present paper up to this point use TBoxes of depth one.

Our main observation is that, when the depth of TBoxes is restricted to one, we can prove a converse of Theorem 17.

Theorem 18. Every materializable $ALCFI$-TBox of depth one is unraveling tolerant.

Proof. (sketch) Let $T$ be a materializable TBox of depth one, $A$ an ABox, and $q$ an ELIQ with $(T, A^u) \not\models q$. We have to show that $(T, A) \not\models q$. It follows from $(T, A^u) \not\models q$ that $A^u$ is consistent w.r.t. $T$ and thus there is a materialization $\bar{I}^u$ for $\bar{T}$ and $A^u$ (even though $A^u$ can be infinite, see long version). We have $\bar{I}^u \not\models q$ and our aim is to convert $\bar{I}^u$ into a model $\bar{I}$ of $\bar{T}$ and $A$ such that $\bar{I} \not\models q$. This is done in two steps.

As a preliminary to the first step, we note that $\bar{I}^u$ can be assumed w.l.o.g. to have forest-shape, i.e., $\bar{I}^u$ can be constructed by selecting a tree-shaped interpretation $\bar{I}_\alpha$ with root $\alpha$ for each $\alpha \in \text{Ind}(A^u)$, then taking the disjoint union of all these interpretations, and finally adding role edges $(\alpha, \beta)$ to $r^x$ whenever $r(\alpha, \beta) \in A^u$. In fact, to achieve the desired shape we can simply unravel $\bar{I}^u$ starting from the elements $\text{Ind}(A^u) \subseteq \Delta^x$ and then use Point 1 of Lemma 8 and the fact that there is an i-simulation from the unraveling of $\bar{I}^u$ to $\bar{I}^u$ to show that the obtained model is still a materialization of $\bar{T}$ and $A$.

Now, step one of the construction is to uniformize $\bar{I}^u$ such that for all $\alpha, \beta \in \text{Ind}(A^u)$ with tail($\alpha$) = tail($\beta$), the tree component $\bar{I}_\alpha$ of $\bar{I}^u$ is isomorphic to the tree component $\bar{I}_\beta$ of $\bar{I}^u$. To achieve this while preserving the property that $\bar{I}^u \not\models q$, we rely on the self-similarity of the ABox $A^u$: for all $\alpha, \beta \in \text{Ind}(A^u)$ with tail($\alpha$) = tail($\beta$), we can find an automorphism on $A^u$ that maps $\alpha$ to $\beta$.

Step two is to construct the desired model $\bar{I}$ of $\bar{T}$ and the original ABox $A$, starting from the uniformized version of $\bar{I}^u$: take the disjoint union of all the tree components $\bar{I}_\alpha$ of $\bar{I}_\alpha$, with $\alpha \in \text{Ind}(A)$ (note that $\text{Ind}(A) \subseteq \text{Ind}(A^u)$), and add $(\alpha, b)$ to $r^x$ whenever $r(\alpha, b) \in A$. Due to the uniformity of $\bar{I}^u$, we can find an i-simulation from $\bar{I}$ to $\bar{I}_\alpha$. Since matches of ELIQs are preserved under i-simulations, $\bar{I}^u \not\models q$ thus implies $\bar{I} \not\models q$. \hfill $\square$

The dichotomy follows: If an $ALCFI$-TBox $T$ of depth one is materializable, then PEQ-answering w.r.t. $T$ is in PTIME by Theorems 18 and 16. Otherwise, ELIQ-answering w.r.t. $T$ is coNP-complete by Theorem 11.

Theorem 19 (Dichotomy). For every $ALCFI$-TBox $T$ of depth one, one of the following is true:

- Q-answering w.r.t. $T$ is in PTIME for any $Q \in \{PEQ, CO, ELIQ\}$.
- Q-answering w.r.t. $T$ is coNP-complete for any $Q \in \{PEQ, CO, ELIQ\}$.

It is interesting to note that, despite the dichotomy, the complexity landscape of depth one TBoxes is still rich. Relevant results can be found in (Calvanese et al. 2006): there are $\mathcal{EL}$-TBoxes for which CQ-answering is PTIME-complete and CQ-answering w.r.t. the $\mathcal{EL}$-TBox $\{\exists r. A \subseteq A\}$, which encodes reachability in directed graphs, is NL-complete. We add that CQ-answering w.r.t. the Horn-$\mathcal{EL}$-box $\{\exists r. A \subseteq A\}$ encodes reachability in undirected graphs and can be shown to be LOGSPACE-complete. Also note that every DL-Lite TBox is of depth one, which gives a whole class of FO-rewritable depth one TBoxes. We conjecture that PTIME, NLOGSPACE, and LOGSPACE-completeness of CQ-answering are decidable for $ALCFI$-TBoxes of depth one. Note that FO-rewritability for materializable $ALCFI$-Tboxes of depth one is decidable. This is shown in (Lutz and Wolter 2011), where FO-rewritability is reduced to boundedness problems for monadic datalog programs and logics over trees (Cosmadakis et al. 1988; Gaifman et al. 1987; Otto, Blumensath, and Weyer 2010). For $ALCI$-TBoxes, a NEXPTIME upper bound can be obtained from the connection to CSPs established in the next section and relevant complexity results on their FO-definability (Larose, Loten, and Tardif 2007).

6 Query Answering in $ALC / ALCI = CSP$

We show that query answering w.r.t. $ALC$ and $ALCI$-TBoxes has the same computational power as non-uniform CSPs in the following sense: (i) for every CSP, there is an $ALC$-TBox such that query answering w.r.t. $T$ is of the same complexity, up to complementation; conversely, (ii) for every $ALCI$-TBox $T$ and ELIQ $q$, there is a CSP that has the same complexity as answering $q$ w.r.t. $T$, up to complementation. This has many interesting consequences, a main one being that the Feder-Vardi conjecture holds if and only if there is a PTIME/coNP-dichotomy for query answering w.r.t. $ALC$-TBoxes (equivalently $ALCI$-TBoxes). All this
is true already for materializable TBoxes. By Theorem 4 and since we carefully choose the appropriate language in each technical result below, it is true for any of our query languages ELIQ, CQ, and PEQ (and ELQ for ALC-­TBoxes).

We begin by introducing non-uniform CSPs. Since every non-uniform CSP is polynomially equivalent to a non-uniform CSP with one binary predicate (Feder and Vardi 1993), we consider CSPs over unary and binary predicates (concept names and role names), only. A signature $\Sigma$ is a finite set of concept and role names. An interpretation $I$ is a $\Sigma$-­interpretation if $\text{Ind}(I) = \emptyset$ and $X^I = \emptyset$ for all $X \in \{N_c \cup N_r\} \setminus \Sigma$. For two finite $\Sigma$-­interpretations $I$ and $I'$, we write $\text{Hom}(I', I)$ if there is a homomorphism from $I'$ to $I$. Any $\Sigma$-­interpretation $I$ gives rise to the following non-­uniform constraint satisfaction problem in signature $\Sigma$, denoted by $\text{CSP}(I)$: given a finite $\Sigma$-­interpretation $I'$, decide whether $\text{Hom}(I', I)$. Numerous algorithmic problems can be given in the form $\text{CSP}(I)$. For example, k-colorability is $\text{CSP}(\mathbb{C}_k)$, where $\mathbb{C}_k$ is an $\{r\}$-­interpretation defined by setting $\mathbb{C}^r_k = \{i, \ldots, k\}$ and $r^{\mathbb{C}_k} = \{(i, j) \mid i \neq j\}$.

We first show how to convert a CSP into a (materializable) ALC-­TBox. For a $\Sigma$-­interpretation $I$, $A_T$ denotes $I$ viewed as an ABox: $A_T = \{\{a \mid a \in \Sigma \cap N_c \land d \in \mathbb{A}^I\} \cup \{\{r(a, b) \mid r \in \Sigma \cap N_r \land (d, e) \in r^I\}\}$. Theorem 20. For every non-uniform constraint satisfaction problem $\text{CSP}(I)$ in signature $\Sigma$, one can compute (in polynomial time) a materializable ALC-­TBox $T_\Sigma$ such that

1. $\text{Hom}(I, J)$ iff $A_J$ is consistent w.r.t. $T_\Sigma$, for all $\Sigma$-­interpretations $I, J$;
2. for any Boolean PEQ $q$, answering $q$ w.r.t. $T_\Sigma$ is polynomially reducible (in fact, FO-­reducible) to the complement of $\text{CSP}(I)$.

Note that $\text{CSP}(I)$ and $T_\Sigma$ ‘have the same complexity’ in the following sense: by Point 1 of Theorem 20, $\text{CSP}(I)$ reduces to consistency of ABoxes w.r.t. $T_\Sigma$; since an ABox $A$ is consistent w.r.t. $T_\Sigma$ iff $\text{Ind}(T_\Sigma, A) \neq A(a)$ with $A$ a fresh concept name and $a \in \text{Ind}(A)$, this also yields a reduction from the complement of $\text{CSP}(I)$ to ELQ-­answering w.r.t. $T_\Sigma$; conversely, Point 2 ensures that (Boolean) PEQ-­answering w.r.t. $T_\Sigma$ reduces to the complement of $\text{CSP}(I)$. All reductions are extremely simple, in polynomial and in fact even FO-­reductions.

Our approach to proving Theorem 20 is to generalize the reduction of k-colorability to query answering w.r.t. ALC-­TBoxes discussed in Examples 1 and 3, where the main challenge is to overcome the observation from Example 3 that PTIME CSPs such as 2-colorability may be translated into CoNP-­hard TBoxes. Note that this is due to the disjunction in the TBox $T_k$ of Example 1, which causes non-­materializability. Our solution is to replace the concept names $A_1, \ldots, A_k$ in $T_k$ with compound concepts that are ‘invisible to the query’, behaving essentially like second-­order variables. Unlike the original depth one TBox $T_k$, the resulting TBox is of depth three.

In detail, fix a constraint satisfaction problem $\text{CSP}(I)$, reserve a concept name $Z_d$ and role names $r_d, s_d$ for any $d \in \mathbb{A}^I$, and set

$$T = \{T \subseteq \exists r_d \cdot T, T \subseteq \exists s_d \cdot Z_d \mid d \in \mathbb{A}^I\}$$

$$H_d = \forall r_d \cdot \exists s_d \cdot \neg Z_d, \quad d \in \mathbb{A}^I$$

The following shows that we can use the concepts $H_d$ as unary predicates to represent the ‘values’ of $\text{CSP}(I)$ (the domain elements of $I$).

Lemma 21. For every ABox $A$ and family of sets $I_d \subseteq \text{Ind}(A), d \in \mathbb{A}^I$, there is a materialization $\bar{T}$ of $T$ and $A$ such that $H_d = I_d$ for all $d \in \mathbb{A}^I$.

Now, the TBox $T_\Sigma$ for the CSP$(I)$ in signature $\Sigma$ from Theorem 20 is $T$ extended with the following CIs:

$$\begin{align*}
T \subseteq \bigcup_{d \in \mathbb{A}^I} H_d \quad &\text{for all } d, e \in \mathbb{A}^I, d \neq e \\
H_d \cap H_e \subseteq \bot \quad &\text{for all } d, e \in \mathbb{A}^I, r \in \Sigma, (d, e) \notin r^I \\
H_d \cap A \subseteq \bot \quad &\text{for all } d \in \mathbb{A}^I, A \in \Sigma, d \notin A^I.
\end{align*}$$

Based on Lemma 21, it can be verified that $T_\Sigma$ satisfies Conditions 1 and 2 of Theorem 20. For Point 2, we show that for all Boolean PEQs $q$ and ABoxes $A$, $(T_\Sigma, A) \models q$ iff $(T, A) \models q$ or not Hom$(T_{\Sigma}', I)$. Here, $T_{\Sigma}'$ is the restriction of $I$ to signature $\Sigma$ and with $\text{Ind}(T_{\Sigma}') = \emptyset$. Moreover, it is not hard to see that $T$ is FO-­rewritable for PEQs, thus $(T, A) \models q$ in $\text{AC}^0$.

We now come to the conversion of an ALCI-­TBox and query $q$ into a CSP. We start with considering Boolean CQs of the form $\exists x.C(x)$, which is not strong enough to obtain the desired dichotomy result, but serves as a warmup that is conceptually clearer than the version for ELIQs that we present afterwards. We use sig$(T)$ to denote the signature of the TBox $T$, and likewise for a CQ $q$.

Theorem 22. Let $T$ be an ALCI-­TBox, $q = \exists x.C(x)$ with $C$ an $\mathcal{EL}$-­concept, and $\Sigma = \text{sig}(T) \cup \text{sig}(q)$. Then one can construct (in time exponential in $|T| + |C|$) a $\Sigma$-­interpretation $T_{\Sigma, q}$ such that for all ABoxes $A$: $(\text{HomDual}) \quad (T, A) \models q \text{ iff not Hom}(T_{\Sigma}', T_{\Sigma, q})$

Proof. (sketch) The interpretation $T_{\Sigma, q}$ can be obtained using a standard type-based construction. We use the sets cl$(T, C)$, tp$(T, C)$, and the relation $\rightsquigarrow$, between types as defined in the proof of Theorem 16. A $T$-­type $t$ that omits $q$ is an element of $\text{tp}(T, C)$ that is satisfiable in a model $J$ of $T$ with $C' = \emptyset$. Then $\Delta^{t, q}$ is the set of all $T$-­types that omit $q$, $t \in \Delta^{t, q}$ iff $A \in t$, for all $A \in \Sigma$, and $(t', t) \in r^{t, q}$ iff $t \rightsquigarrow t'$, for all $r \in \Sigma$. It is shown in the long version that condition (HomDual) is satisfied. Finally, a Pratt-­style type elimination algorithm can be used to construct $T_{\Sigma, q}$ in exponential time (Pratt 1979).

Example 23. Let $T = \{A \subseteq \forall r.B\}$ and define $q = \exists x.B(x)$. Then $T_{\Sigma, q}$ is defined, up to isomorphism, by $\Delta^{t, q} = \{(a, b), c\}, A^{T_{\Sigma, q}} = \{b\}, B^{T_{\Sigma, q}} = \emptyset$, and $r^{T_{\Sigma, q}} = \{(a, a), (a, b), (a, c)\}$. 

\end{document}
For ELIQs, the conversion of a TBox and query into a CSP is similar to the one above, but employs a concept name \( P \) that represents the individual name used in the ELIQ.

**Theorem 24.** Let \( T \) be an ALCI-TBox, \( C(a) \) an ELIQ and \( \Sigma = \text{sig}(T) \cup \text{sig}(C) \cup \{ P \} \), where \( P \) is a fresh concept name. Then one can construct (in time exponential in \(|T| + |C|\)) a \( \Sigma \)-interpretation \( I_{T,q} \) such that for all ABoxes \( A \):

1. \( (T, A) \models C(a) \iff \text{not } \text{Hom}(I_{A,q}^\Sigma, I_{T,q}) \), where \( A' \) is obtained from \( A \) by adding \( P(a) \) and removing all other assertions that use \( P \);
2. \( (T, A) \models \exists x. (P(x) \land C(x)) \iff \text{not } \text{Hom}(I_{A,q}^\Sigma, I_{T,q}) \).

As a consequence of Theorems 20 and 24, we obtain:

**Theorem 25.** There is a dichotomy between PTIME and coNP for CQ-answering w.r.t. ALCI-TBoxes if and only if the Feder-Vardi conjecture is true.

The same is true for ALCI-TBoxes, for ELIQs, and PEQs. For ALC-F-TBoxes, it additionally holds for ELIQs.

**Proof.** Let CSP(\( I \)) be an NP-intermediate CSP, i.e., a CSP that is neither in \( \text{PTIME} \) nor \( \text{NP} \)-hard. Take the TBox \( T' \) from Theorem 20. By Point 1 of that theorem (and the mentioned reduction of ABox consistency to the complement of ELQ-answering), CQ-answering w.r.t. \( T' \) is not in \( \text{PTIME} \). By Point 2, CQ-answering w.r.t. \( T' \) is not \( \text{coNP} \)-hard.

Conversely, let \( T \) be an ALCI-TBox for which CQ-answering w.r.t. \( T \) is neither in \( \text{PTIME} \) nor \( \text{coNP} \)-hard. Then by Theorem 4 and since every ELIQ is a CQ, the same holds for ELIQ-answering w.r.t. \( T \). It follows that there is concrete ELIQ \( q \) such that answering \( q \) w.r.t. \( T \) is \( \text{coNP} \)-intermediate. Then \( I_{T,q} \) be the interpretation constructed in Point 1 of Theorem 24. By Point 1 of that theorem, CSP(\( I \)) is not in \( \text{PTIME} \); by Point 2, it is not \( \text{NP} \)-hard.

The construction in Theorem 24 cannot be generalized from ELIQs to CQs. For simplicity, let us consider Theorem 22 instead. We show that it is impossible to construct an interpretation \( I_{T,q} \) that satisfies (HomDual) for Boolean CQs that are not of the simple form \( \exists x. C(x) \). Indeed, this holds even with the empty TBox. It is thus crucial to use ELIQs even when proving the dichotomy result for CQs and PEQs.

The following states this more formally.

**Theorem 26.** Let \( q \) be a Boolean CQ without individual names, \( \text{sig}(q) = \Sigma \), and \( I_0 \) the empty TBox. Then there is a \( \Sigma \)-interpretation \( I_{q,I_0} \) that satisfies (HomDual) if \( q \) is logically equivalent to a CQ of the form \( \exists x. C(x) \) with \( C \) an \( \Delta_2 \)-concept.

**Proof.** This is a consequence of results on homomorphism dualities (Nesetril and Tardif 2000), the problem of constructing for a given \( \Sigma \)-interpretation \( I \), a \( \Sigma \)-interpretation \( J \) such that the following duality holds for all \( \Sigma \)-interpretations \( J' \):

\[ \text{Hom}(I, J) \iff \text{not } \text{Hom}(J', I) \]

By (Nesetril and Tardif 2000; Nesetril 2009), such an \( J \) exists iff the undirected graph induced by \( I \) is a tree. It remains to observe that for any Boolean CQ \( q \) without individual names and all \( \Sigma \)-interpretations \( J \), we have \( A_J \models q \iff \text{Hom}(I_q, J) \), where \( I_q \) is the interpretation with \( \Delta_2^+ \) the variables in \( q \) and in which \( x \in A_I^2 \) (resp. \( x,y \in r_I^2 \)) if \( A(x) \) (resp. \( r(x,y) \)) is a conjunct of \( q \).

Interestingly, (Nesetril 2009) presents five constructions of \( T \), one of which resembles our type elimination procedure (but, of course, without taking into account TBoxes).

### 7 Non-Dichotomy in \( \text{ALCF} \)

We show that the complexity landscape for \( \text{ALCF} \) query answering is much richer than for \( \text{ALCI} \). In particular, we show that (i) for CQ-answering w.r.t. \( \text{ALCF} \)-TBoxes, there is no dichotomy between \( \text{PTIME} \) and \( \text{coNP} \) unless \( \text{PTIME} = \text{NP} \); and (ii) CQ-answering in \( \text{PTIME} \) is undecidable for \( \text{ALCF} \)-TBoxes, and likewise for \( \text{coNP} \)-hardness, materializability and FO-rewritability. Point (i) is a consequence of the following, much stronger, result.

**Theorem 27.** For every language \( L \) in \( \text{coNP} \), there is an \( \text{ALCF} \)-TBox \( T \) and ELIQ \( \text{rej} \) a concept name, such that the following holds:

1. there exists a polynomial reduction of deciding \( v \in L \) to answering \( \text{rej}(a) \) w.r.t. \( T \);
2. for every ELIQ \( q \), \( q \) w.r.t. \( T \) is polynomially reducible to deciding \( v \in L \).

To show (i), assume that for every \( \text{ALCF} \)-TBox \( T \), CQ answering w.r.t. \( T \) is in \( \text{PTIME} \) or \( \text{coNP} \)-hard. By Ladner’s Theorem, one can take a \( \text{coNP} \)-intermediate language \( L \) and let \( T \) be the TBox from Theorem 27. By Point 1 of the theorem, CQ-answering w.r.t. \( T \) is not in \( \text{PTIME} \). Thus it must be \( \text{coNP} \)-hard. By Theorem 4 and since a dichotomy for CQ-answering w.r.t. \( T \) also implies a dichotomy for ELIQ-answering w.r.t. \( T \), ELIQ-answering w.r.t. \( T \) is also \( \text{coNP} \)-hard. By Point 2 of Theorem 27, this is impossible.

The proof of Theorem 27 combines the ‘hidden’ concepts \( H_d \) from the proof of Theorem 20 with a modification of the TBox constructed in (Baader et al. 2010) to prove the undecidability of query emptiness in \( \text{ALCF} \). Using a similar strategy, we establish the undecidability results announced as Point (ii) above, summarized by the following theorem.

**Theorem 28.** For \( \text{ALCF} \)-TBoxes \( T \), the following problems are undecidable (Points 1 and 2 are subject to the side condition that \( \text{PTIME} \neq \text{NP} \)):

1. CQ-answering w.r.t. \( T \) is in \( \text{PTIME} \);
2. CQ answering w.r.t. \( T \) is \( \text{coNP} \)-hard;
3. \( T \) is materializable.

### 8 Conclusions

Much work remains to be done to fully accomplish the general research goal formulated in the introduction. In particular, lower complexity classes, FO-rewritability, and generalizations of unraveling tolerance/arc consistency should be investigated. It would also be interesting to identify additional ways to exploit the new CSP connection and to explore the applicability of our method to other DLs and to fragments of first-order logic such as the guarded fragment.

**Acknowledgments.** C. Lutz was supported by the DFG SFB/TR 8 “Spatial Cognition”.
References


A Proofs for Section 2
In this section, we prove Theorem 4. Note that in the proofs of Theorems 9 and 11 we do not use Theorem 4. Thus, we can (and will) employ them in the proof below. We formulate Theorem 4 again.

Theorem 4 For all ALCFI-TBoxes $T$, the following are equivalent:
1. CQ-answering w.r.t. $T$ is in $PTIME$ iff PEQ-answering w.r.t. $T$ is in $PTIME$ iff ELIQ-answering w.r.t. $T$ is in $PTIME$;
2. $T$ is FO-rewritable for CQ iff it is FO-rewritable for PEQ iff it is FO-rewritable for ELIQ.

If $T$ is an ALCFI-TBoxes, then we can replace ELIQ in Points 1 and 2 with ELQ.

We start the proof with the observation that the implications
- If PEQ-answering w.r.t. $T$ is in $PTIME$, then CQ-answering w.r.t. $T$ is in $PTIME$;
- If CQ answering w.r.t. $T$ is in $PTIME$, then ELIQ-answering w.r.t. $T$ is in $PTIME$;
- If $T$ is FO-rewritable for PEQ, then $T$ is FO-rewritable for CQ;
- If $T$ is FO-rewritable for CQ, then $T$ is FO-rewritable for ELIQ

are trivial, by the obvious inclusions between the sets of queries considered. For the proofs of the other directions we can assume that $T$ is materializable: otherwise, by Theorems 9 and 11, ELIQ-answering w.r.t. $T$ is coNP-hard and the implications are trivial.

For materializable $T$, the implications
- If CQ-answering w.r.t. $T$ is in $PTIME$, then PEQ-answering w.r.t. $T$ is in $PTIME$;
- If $T$ is FO-rewritable for CQ, then $T$ is FO-rewritable for PEQ;

are obvious since the evaluation of a disjunction in an interpretation reduces to evaluating all its disjuncts. Thus, it remains to show the following two implications:
1. If ELIQ answering w.r.t. $T$ is in $PTIME$, then CQ-answering w.r.t. $T$ is in $PTIME$;
2. If $T$ is FO-rewritable for ELIQ, then $T$ is FO-rewritable for CQ.

To show these implications, we introduce some notation and a lemma. For a sequence $\vec{r} = r_1 \cdots r_n$ of roles, we set $\exists r.C = \exists r_1 \cdots \exists r_n.C$. In an interpretation $I$, the distance $\text{dist}_I(d_1, d_2)$ between $d_1, d_2 \in \Delta^I$ is the minimal $n$ such that there are $d_1 = e_0, \ldots, e_n = d_2$ and roles $r_1, \ldots, r_n$ with $(d_i, d_{i+1}) \in r_i^I$ for $i < n$.

Lemma 29. Let $C$ be an ELI$^1$-concept and assume that $(T, A) \models \exists v.C(v)$. If $T$ is materializable, then there exists a sequence of roles $\vec{r} = r_1 \cdots r_n$ of length $n \leq 2^{2^{2^{(|T|+|C|)}}} \times 2^{|T||C|} + 1$ such that there exists $a \in \text{Ind}(A)$ with $(T, A) \models \exists \vec{r}.C(a)$. 

---

Proof. Let $\mathcal{I}$ be a PEQ-materialization of $\mathcal{T}$ and $\mathcal{A}$. We may assume that $\mathcal{I}$ is i-unfolded. From $(\mathcal{T}, \mathcal{A}) \models \exists v. C(v)$, we obtain $C^\mathcal{I} \neq \emptyset$. Choose $d \in C^\mathcal{I}$ and $a \in \text{Ind}(\mathcal{A})$ such that $\pi := \text{dist}_\mathcal{I}(d, a^\mathcal{I})$ is minimal. (We assume, for simplicity, that there is only one such $d$. The argument is easily generalized.) Assume $n > 2^{2(|\mathcal{T}|+|C|)} \times 2|\mathcal{T}|C| + 1$.

Let $a^\mathcal{I} = d_0, \ldots, d_n = d$ and $(d_i, d_{i+1}) \in r^\mathcal{I}$ for $i < n$. Let $\text{sub}(\mathcal{T}, C)$ denote the closure under single negation of the set of subconcepts of concepts in $\mathcal{T}$ and $C$. Set

$$t^\mathcal{I}(e) = \{ D \in \text{sub}(\mathcal{T}, C) \mid e \in D^\mathcal{I} \}.$$  

As $n > 2^{2(|\mathcal{T}|+|C|)} \times 2|\mathcal{T}|C| + 1$, there exist $d_i$ and $d_{i+j}$ with $j > 1$ and $i + j < n$ such that

$$t^\mathcal{I}(d_i) = t^\mathcal{I}(d_{i+j}), \quad t^\mathcal{I}(d_{i+1}) = t^\mathcal{I}(d_{i+j+1}), \quad r_{i+1} = r_{i+j+1}.$$  

Now replace in $\mathcal{I}$ the interpretation induced by the subtree generated by $d_{i+j+1}$ by the interpretation induced by the subtree generated by $d_{i+1}$ and denote the resulting interpretation by $\mathcal{J}$. $\mathcal{J}$ is still a model of $(\mathcal{T}, \mathcal{A})$. But now $\mathcal{J} \not\models \exists \vec{y}\phi(\vec{a})$. We have a contradiction since $a^\mathcal{I} \in (\exists \vec{y}\phi(\vec{a}))^\mathcal{I}$ and therefore, since $\mathcal{I}$ is a minimal model of $(\mathcal{T}, \mathcal{A})$, $(\mathcal{T}, \mathcal{A}) \models (\exists \vec{y}\phi(\vec{a}))^\mathcal{I}$. $
$
Let $q(\vec{x}) = \exists \vec{y}.\phi(\vec{x}; \vec{y})$ be a CQ with $\vec{x} = x_1, \ldots, x_n$ and $\vec{y} = y_1, \ldots, y_m$. We regard $\phi$ as a set of atoms. A splitting $S = (Y, \sim, f)$ of $q(\vec{x})$ consists of a subset $Y$ of $\vec{y}$, an equivalence relation $\sim$ on $\text{Ind}(q) \cup \vec{x} \cup Y$ and a mapping $f : \{ u_\sim \mid u \in \text{Ind}(q) \cup \vec{x} \cup Y \} \rightarrow 2^{\vec{x} \cup Y}$

(we denote by $u_\sim$ the equivalence class of $u$ w.r.t. $\sim$) such that

- for every $y \in \vec{y} \setminus Y$ there exists $u \in f(u_\sim)$;
- $f(u_\sim) \cap f(v_\sim) = \emptyset$ whenever $u_\sim \neq v_\sim$;
- if $r(t, t') \in \phi$ or $r(t', t) \in \phi$ and $t \in f(u_\sim)$, then $t' \in u_\sim$ or $t' \in f(u_\sim)$.

Let $U_S$ denote the set of all equivalence classes $u_\sim$ w.r.t. $\sim$. Thus, if $(Y, \sim, f)$ is a splitting of $q(\vec{x})$, we can form

- $\forall \vec{y}^\prime$ consisting of all $A(t)$ with $t \in \text{Ind}(q) \cup \vec{x} \cup Y$ and all $r(t, t')$ with $t, t' \in \text{Ind}(q) \cup \vec{x} \cup Y$;
- for every $u_\sim \in U_S$, $\forall \vec{y}^\prime$ consisting of all $A(t)$ and $r(t, t')$ with $t, t' \in u_\sim \cup f(u_\sim)$.

Intuitively, splittings describe potential assignments $\pi$ for the variables in $\vec{x}, \vec{y}$ in an unfolded CQ-materialization $\mathcal{I}$ of $(\mathcal{T}, \mathcal{A})$ in which

- all $v \in u_\sim$ receive the same value $\pi(v)$ and this value is in $\text{Ind}(\mathcal{A})$;
- all $y \in f(u_\sim)$ receive values $\pi(y)$ in the “anonymous” tree generated by $\pi(u)$.

Using Lemma 29 (for those $y$ that are not reachable in $\phi$ from any member of $\text{Ind}(\mathcal{A}) \cup \vec{x} \cup Y$) one can easily construct, for every $u_\sim \in U_S$ a disjunction $D_u = \bigvee_{y \in f(u_\sim)} C_y$ of $\mathcal{E}\mathcal{L}$-concepts such that for all CQ-materializations $\mathcal{I}$ of some $(\mathcal{T}, \mathcal{A})$ and all $a \in \text{Ind}(\mathcal{A})$, $(1.)$ implies $(2.)$ and $(2.)$ implies $(3.)$, where

1. there exists an assignment $\pi$ in $\mathcal{I}$ with

- $\pi(u) = \pi(u') = a^\mathcal{I}$ for all $u' \in u_\sim$;
- $\pi(x)$ in the anonymous subtree generated by $a^\mathcal{I}$ for all $x \in f(u_\sim)$;
- $\mathcal{I} \models \forall \phi_u$.

2. $a^\mathcal{I} \in D_u^\mathcal{I}$.

3. there exists an assignment $\pi$ in $\mathcal{I}$ with

- $\pi(u) = \pi(u') = a^\mathcal{I}$ for all $u' \in u_\sim$;
- $\mathcal{I} \models \forall \phi_u$.

For every splitting $S = (Y, \sim, f)$ of $\phi(\vec{x})$, set

$$\chi_S = \phi_Y \land \bigwedge_{u_\sim \in U_S} \bigwedge_{t, t' \in u_\sim} (t = t') \land \bigwedge_{u_\sim \in U_S} D_u.$$  

To prove the implication (2.), assume that $\mathcal{T}$ is FO-writable for ELIQ. By materializability, $\mathcal{T}$ is FO-writable for unions of ELIQs. For every $u_\sim \in U_S$, let $\chi_u$ be a FOQ with

$$\mathcal{I}_A \models \chi_u[a] \Leftrightarrow (\mathcal{T}, \mathcal{A}) \models D_u(a).$$  

for all $a \in \text{Ind}(\mathcal{A})$. Let $\chi_u^\ast$ be the FOQ resulting from $\chi_u$ by replacing every $D_u$ with $\chi_u$. Then it is readily checked that

$$\mathcal{I}_A \models \bigvee_{S \in \text{splits of } q(\vec{x})} \exists \vec{a}.\chi_u^\ast[\vec{a}] \Leftrightarrow (\mathcal{T}, \mathcal{A}) \models q(\vec{a})$$  

for all $\vec{a} \in \text{Ind}(\mathcal{A})$. Thus, $\mathcal{T}$ is FO-writable for CQ.

We come to implication (1.). Assume that ELIQ-answering w.r.t. $\mathcal{T}$ is in PTIME. By materializability, unions of ELIQs can be answered w.r.t. $\mathcal{T}$ in PTIME. We can evaluate a CQ $q(\vec{x})$ in polynomial time as follows: to decide whether $(\mathcal{T}, \mathcal{A}) \models q(\vec{a})$ for a given $\vec{a} \in \text{Ind}(\mathcal{A})$, go through all splittings $S = (Y, \sim, f)$ of $q(\vec{x})$ and all assignments $\pi(y) \in \text{Ind}(\mathcal{A})$ for $y \in Y$ and check

$$\mathcal{I}_A \models \phi_Y \land \bigwedge_{u_\sim \in U_S} \bigwedge_{t, t' \in u_\sim} (t = t')[\vec{a}]$$  

and

$$(\mathcal{T}, \mathcal{A}) \models \bigvee_{u_\sim \in U_S} D_u(\pi(u)).$$  

If both hold for at least one pair $S, \pi$, then $(\mathcal{T}, \mathcal{A}) \models q(\vec{a})$; otherwise $(\mathcal{T}, \mathcal{A}) \not\models q(\vec{a})$. Both conditions can be checked in polynomial time.

B Proofs for Section 3

We introduce some notions and notations. For any interpretation $\mathcal{I}$, we define its i-unfolding $\mathcal{I}^*$. The domain $\Delta^{\mathcal{I}^*}$ of $\mathcal{I}^*$ consists of all words $d_0q_1 \cdots r_n d_n$ with $n \geq 0$, $d_i \in \Delta^{\mathcal{I}^*}$, and $r_i$ (possibly inverse) roles such that

- there exists $a \in \text{Ind}(\mathcal{A})$ with $d_0 = a^{\mathcal{I}^*}$;
- for $0 < i \leq n$ there does not exists $a \in \text{Ind}(\mathcal{A})$ such that $d_i = a^{\mathcal{I}^*}$;
- for $0 \leq i < n$: $(d_i, d_{i+1}) \in r_{i+1}^\mathcal{I}$ and if $r_i = r_{i+1}$, then $d_{i-1} \neq d_{i+1}$.

For $d_0 \cdots d_n \in \Delta^{\mathcal{I}^*}$, we set $\text{tail}(d_0 \cdots d_n) = d_n$. Now set
• for all $A \in N_C$:
$$A^\ast = \{ w \in \Delta^\ast | \text{tail}(w) \in A^\ast \}$$

• for all $r \in N_R$:
$$r^\ast = \{ \langle \sigma, \sigma r d \rangle, \sigma, \sigma r d \in \Delta^\ast \cup \{ \sigma r^{-1} d, \sigma \} | \sigma, \sigma r^{-1} d \in \Delta^\ast \}$$

• $\text{Ind}(I^\ast) = \text{Ind}(I)$ and $a^\ast = a^\ast$, for all $a \in \text{Ind}(I)$.

We call an interpretation $I$ i-unfolded if it is isomorphic to its own i-unfolding. Clearly, every i-unfolding $I^\ast$ of an interpretation $I$ is i-unfolded.

For $\text{ALCFI}$-TBoxes it is not required to unfold along inverse roles. Thus, we define the domain $\Delta^\ast$ of the unfolding $I^\ast$ of $I$ as the set of all words $d_0 r_1 \ldots r_n d_n$ with $n \geq 0$, $d_i \in \Delta^\ast$, and $r_i$ role names. The definition of the interpretation of concept, role and individual names remains the same (but can be simplified). We call an interpretation $I$ unfolded if it is isomorphic to its own unfolding. Every unfolding $I^\ast$ of an interpretation $I$ is unfolded.

**Lemma 30.** Let $I$ be an interpretation.

• $f(w) := \text{tail}(w), w \in \Delta^\ast$, is a homomorphism from $I^\ast$ to $I$;

• $f(w) := \text{tail}(w), w \in \Delta^\ast$, is a homomorphism from $I^\ast$ to $I$;

• for any interpretation $J$, if there is an i-simulation between $I$ and $J$, then there is a homomorphism from $I^\ast$ to $J$;

• for any interpretation $J$, if there is a simulation between $I$ and $J$, then there is a homomorphism from $I^\ast$ to $J$;

• If $I$ is a model of $(T, A)$ with $T$ an $\text{ALCFI}$-TBox, then $I^\ast$ is a model of $(T, A)$;

• If $I$ is a model of $(T, A)$ with $T$ an $\text{ALCFI}$-TBox, then $I^\ast$ is a model of $(T, A)$.

We formulate Lemma 8 again.

**Lemma 8** Let $T$ be an $\text{ALCFI}$-TBox and $A$ and ABox. A model $I$ of $T$, $A$ is

1. an ELIQ-materialization of $T$ and $A$ iff it is i-sim-initial in $\text{Mod}(T, A)$;

2. a PEQ-materialization of $T$ and $A$ iff it is a CQ-materialization of $T$ and $A$ iff it is hom-initial in $\text{Mod}(T, A)$.

If $T$ is a $\text{ALCFI}$-TBox, then $I$ is an ELQ-materialization of $T$ and $A$ if and only if it is i-sim-initial in $\text{Mod}(T, A)$.

**Proof.** We apply Lemma 30.

1. Consider the direction from left to right only. Let $I$ be an ELIQ-materialization and $J \in \text{Mod}(T, A)$. Assume first that $J$ has finite outdegree. For $a \in \text{Ind}(A)$, let $I^a$ and $J^a$ denote the interpretations obtained from $I$ and $J$ by setting $\text{Ind}(I^a) = \{ a \}$ and $\text{Ind}(J^a) = \{ a \}$, respectively. Thus, the only difference is that only the individual name $a$ is interpreted. Using the condition that $J$ has finite outdegree, one can readily show $(1) \Rightarrow (2)$, where

1. for all $\text{ELIQ}$-concepts $C$: if $I \models C(a)$, then $J \models C(a)$;

2. there is an i-simulation $S_a$ between $I^a$ and $J^a$.

Now, condition $(1)$ holds for all $a \in \text{Ind}(I)$ since $I$ is an ELIQ-materialization of $T$ and $A$. Thus $\bigcup_{a \in \text{Ind}(A)} S_a$ is an i-simulation between $I$ and $J$, as required.

Now assume that $J$ does not have finite outdegree. Construct the i-unfolding $J^\ast$ of $J$. From $J^\ast$ we obtain an interpretation $J^b$ of bounded outdegree by selective filtration as follows: let $S_0 = \text{Ind}(A)$ and assume $S_n$ has been defined. Then define $S_{n+1}$ as the union of $S_n$ and, for every $d \in S_n$ and $\exists r. D \in \text{sub}(T)$ with $d \in (\exists r. D)^J$, some witness $d' \in D^J$ with $(d, d') \in r^J$ and $d' \in D^J$ if no such $d'$ exists already in $S_n$. Let $S = \bigcup_{n \geq 0} S_n$. Let $J^b$ be the restriction of $J^\ast$ to $S$. The outdegree of $J^b$ is bounded by $|\text{sub}(T)| + |A|$, and, therefore, finite. By construction, $J^b \in \text{Mod}(T, A)$. Since there is a homomorphism from $J^\ast$ to $J$, its restriction to the domain of $\Delta^J$ is a homomorphism from $J^b$ to $J$. Since $J^b$ has finite outdegree, we find an i-simulation $D$ between $I$ and $J^b$. The composition of $D$ and $f$ is the required i-simulation between $I$ and $J$.

(2) Let $I$ be countable and generated. It is straightforward to check that if $I$ is hom-initial, then it is a PEQ-materialization; obviously, if $I$ is a PEQ-materialization, then it is a CQ-materialization. Thus, it remains to show that if $I$ is a CQ-materialization, then it is hom-initial. Assume that $I$ is a CQ-materialization and $J \in \text{Mod}(T, A)$. We assume $J$ has finite outdegree (the infinite outdegree case can be reduced to the finite outdegree case as in $(1)$). First, for every finite subset $X$ of $I$ we obtain from the condition that $I$ is a CQ-materialization that there is a homomorphism $h_X$ from the subinterpretation $I|_X$ of $I$ induced by $X$ into $J$. Since $I$ is countable, we can take an enumeration $a_1, \ldots$ of $\Delta^I \setminus \text{Ind}(A)$. Let $X_0, X_1, \ldots$ be a sequence of finite subsets of $\Delta^I$ such that

• $X_0 = \{ a \mid a \in \text{Ind}(A) \}$

• $X_n \subseteq X_{n+1}$ for $n \geq 0$;

• $X_n \models \text{Ind}(A) \cup \{ d_1, \ldots, d_n \}$, for $n \geq 0$;

• for all $d \in X_n$ there exists a path in $X_n$ from some $a \in \text{Ind}(A)$ to $d$.

Condition $4$ can be satisfied since $I$ is generated. Let $h_{X_n}$ be a homomorphism from $I|_{X_n}$ to $J$, for $n \geq 0$. We define the required homomorphism as the limit of a sequence of homomorphisms $f_0, f_1, \ldots$. Let $f_0 = h_{X_0}$ and assume $f_n$ has been defined. We assume that there is an infinite set $X \subseteq N$ such that for all $m \in X$: if $d \in \text{dom}(f_n)$, then $f_n(d) = h_{X_m}(d)$ for all $m \in X$. Let $i$ be minimal such that $d_i$ is not in the domain of $f_n$ (if no such $i$ exists, we are done). Let $X_m$ be the minimal $m$ such that $d_i \in X_m$ and $X_m \subseteq X$. Let $k$ be the length of the shortest path from some $a^\ast$ with $a \in \text{Ind}(A)$ to $d_i$ in $X_m$. This is a path in any $X_n$ with $n \geq m$. Thus, for all $X_n$, $n \geq m$, there exists a path of length $\leq k$ from $h_X(d_i)$ to some $a^\ast$ in $J$. Since $J$ has finite outdegree, there exists an infinite subset $X'$ of $X$ such that $h_X(d_i) = h_Y(d_i)$ for all $X, Y \subseteq X'$. We now set $f_{n+1} = f_n \cup \{ (d_i, e) \}$, where $h_X(d) = e$ for all $X \in X'$. This finishes the construction of $f_{n+1}$.
We set \( f = \bigcup_{n \geq 0} f_n \). By definition, \( f \) is a homomorphism, as required.

The claim for ALCF-TBoxes is proved in the same way as (1).

We show that the generatedness condition cannot be dropped: consider the TBox

\[
\mathcal{T} = \{ A \subseteq \exists r, A, B \subseteq A \}
\]

It is readily seen that \( \mathcal{T} \) is PEQ-materializable. Let \( \mathcal{A} = \{ B(a) \} \). The interpretation \( \mathcal{I} \) with

- \( \Delta^I = \{ a \} \cup \{ 1, 2, \ldots \} \);
- \( A^I = \Delta^I \);
- \( B^I = \{ a \} \);
- \( \rho^I = \{ (a, 1) \} \cup \{ (n, n + 1) \mid n \geq 1 \} \)

is hom-initial in \( \text{Mod}(\mathcal{T}, \mathcal{A}) \). However, the interpretation \( \mathcal{I}' \) defined as the disjoint union of \( \mathcal{I} \) and the interpretation \( \mathcal{J} \) with

- \( \Delta^J = \{ \ldots, -2, -1, 0, 1, 2, \ldots \} \);
- \( \rho^J = \{ (n, n + 1) \mid n \in \Delta^J \} \);
- \( A^J = \Delta^J \);
- \( B^J = \emptyset \)

is a PEQ-materialization of \( \mathcal{T} \) and \( \mathcal{A} \), but it is not hom-initial as there is no homomorphism from \( \mathcal{J} \) to \( \mathcal{I} \).

**Proof of Theorem 9** We apply Lemmas 8 and 30. For Points 1 and 2, assume that \( I \) is i-initial in \( \text{Mod}(\mathcal{T}, \mathcal{A}) \).

We have to show that there exists a model \( \mathcal{I}' \) of \( \mathcal{T} \) and \( \mathcal{A} \) that is hom-initial in \( \text{Mod}(\mathcal{T}, \mathcal{A}) \). To construct \( \mathcal{I}' \), let \( I_0 \) be an at most countable and generated subinterpretation of \( I \) in \( \text{Mod}(\mathcal{T}, \mathcal{A}) \). For example, one can take an elementary subinterpretation of \( I \) and then restrict its domain to the points reachable from \( \{ a^I \mid a \in \text{Ind}(\mathcal{A}) \} \) in \( I \). Clearly, \( I_0 \) is still i-initial in \( \text{Mod}(\mathcal{T}, \mathcal{A}) \).

Now, the unfolding \( I_0 \) of \( I_0 \) is hom-initial in \( \text{Mod}(\mathcal{T}, \mathcal{A}) \). The final Point of Theorem 9 is proved similarly.

**Proof of Theorem 11**

The proof is by reduction of 2+2-SAT, a variant of propositional satisfiability that was first introduced by Schaefer as a tool for establishing lower bounds for the data complexity of query answering in a DL context (Schaefer 1993). A 2+2 clause is of the form \( (p_1 \vee p_2 \vee \neg n_1 \vee \neg n_2) \), where each of \( p_1, p_2, n_1, n_2 \) is a propositional letter or a truth constant 0, 1. A 2+2 formula is a finite conjunction of 2+2 clauses. Now, 2+2-SAT is the problem of deciding whether a given 2+2 formula is satisfiable. It is shown (in Schaefer 1993) that 2+2-SAT is NP-complete.

**Theorem 11.** If an ALCFI-TBox \( \mathcal{T} \) (ALCF-TBox \( \mathcal{T} \)) is not materializable, then ELIQ-answering (ELQ-answering) is coNP-hard w.r.t. \( \mathcal{T} \).

**Proof.** We first show that if an ALCFI-T is not materializable, then Boolean UELIQ-answering w.r.t. \( \mathcal{T} \) is coNP-hard, where a Boolean UELIQ is a disjunction \( q_1 \vee \cdots \vee q_k \), with each \( q_i \) a Boolean ELIQ. We then sketch the modifications necessary to lift the result to Boolean ELIQ-answering w.r.t. \( \mathcal{T} \).

Since \( \mathcal{T} \) is not materializable, by Theorem 9 it does not have the disjunction property. Thus, there is an ABox \( \mathcal{A}_v \) and ELIQs \( C_0(a_0), \ldots, C_k(a_k) \) such that \( \mathcal{T}, \mathcal{A}_v \models C_0(a_0) \vee \cdots \vee C_k(a_k) \), but \( \mathcal{T}, \mathcal{A}_v \not\models C_i(a_i) \) for all \( i \leq k \).

We use \( \mathcal{A}_v \) and the sequence \( C_0(a_0), \ldots, C_k(a_k) \) to generate truth values for variables in the input 2+2 formula.

Let \( \varphi = c_0 \land \cdots \land c_n \) be a 2+2 formula in propositional letters \( q_0, \ldots, q_m \), and let \( c_i = p_{i,1} \lor p_{i,2} \lor \neg n_{i,1} \lor \neg n_{i,2} \) for all \( i \leq n \). Our aim is to define an ABox \( \mathcal{A}_v \) and a Boolean UELIQ \( q \) such that \( \varphi \) is unsatisfiable iff \( \mathcal{T}, \mathcal{A}_v \models q \). To start, we represent the formula \( \varphi \) in the ABox \( \mathcal{A}_v \) as follows:

- the individual name \( f \) represents the formula \( \varphi \);
- the individual names \( c_0, \ldots, c_n \) represent the clauses of \( \varphi \);
- the assertions \( c(f, c_0), \ldots, c(f, c_n) \), associate \( f \) with its clauses, where \( c \) is a role name that does not occur in \( \mathcal{T} \);
- the individual names \( q_0, \ldots, q_m \) represent variables, and the individual names 0, 1 represent truth constants;
- the assertions

\[
\bigcup_{i \leq n} \{ p_{i,1}(c_{i,1}), p_{i,2}(c_{i,2}), n_{i,1}(c_{i,1}, n_{i,1}), n_{i,2}(c_{i,1}, n_{i,2}) \}
\]

associate each clause with the four variables/truth constants that occur in it, where \( p_{1,2}, n_{1,2} \) are role names that do not occur in \( \mathcal{T} \).

We further extend \( \mathcal{A}_v \) to enforce a truth value for each of the variables \( q_i \). To this end, add to \( \mathcal{A}_v \) copies \( A_0, \ldots, A_m \) of \( \mathcal{A}_v \) obtained by renaming individual names such that \( \text{Ind}(A_j) \cap \text{Ind}(A_j') = \emptyset \) whenever \( i \neq j \). As a notational convention, let \( a_j' \) be the name used for the individual name \( a_j \) in \( \mathcal{A}_j \), for all \( i \leq m \) and \( j \leq k \) (note that \( a_j \) comes from the ELIQ \( C_j(a_j) \) in the sequence fixed above).

Intuitively, the copy \( \mathcal{A}_j \) of \( \mathcal{A} \) is used to generate a truth value for the variable \( q_j \), where we want to interpret \( q_j \) as true if the ELIQ \( C_0(a_0) \) is satisfied and as false if any of the ELIQs \( C_j(a_j) \), \( 0 < j \leq k \), is satisfied. To actually relate each individual name \( q_i \) to the associated ABox \( \mathcal{A}_i \), we use role names \( r_0, \ldots, r_k \) that do not occur in \( \mathcal{T} \). More specifically, we extend \( \mathcal{A}_v \) as follows:

- link variables \( q_i \) to the ABoxes \( \mathcal{A}_i \) by adding assertions \( r_j(q_i, a_j') \) for all \( i \leq m \) and \( j \leq k \); thus, truth of \( q_i \) means that \( \exists r_0, C_0(q_i) \) is satisfied and falsity means that \( \exists r_j, C_j(q_i) \) is satisfied for some \( j \) with \( 0 < j \leq k \);
- to ensure that 0 and 1 have the expected truth values, add a copy of \( C_0 \) viewed as an ABox with root 1′ and a copy of \( C_2 \) viewed as an ABox with root 0′; add \( r_0(1, 1') \) and \( r_1(0, 0') \).
Consider the query
\[ q_0 = 3c.(\exists p_1, ff \land \exists p_2, ff \land \exists n_1, tt \land \exists n_2, tt) \]
which describes the existence of a clause with only false literals and thus captures falsity of \( \varphi \), where \( tt \) is an abbreviation for \( \exists r_0, C_0 \) and \( ff \) an abbreviation for the ELIQ-concept \( \exists r_1, C_1 \cup \cdots \cup \exists r_k, C_k \). It is straightforward to show that \( \varphi \) is unsatisfiable iff \( \langle A, T \rangle \models q_0 \). To obtain the desired UELIQ \( q \), it remains to take \( q \) and distribute disjunction to the outside.

We now show how to improve the result from UELIQ-answering to ELIQ-answering. Our aim is to change the \( \exists r \) occur in \( \exists r \) and thus captures falsity of \( \varphi \). The basic idea is that each \( \exists r \) is satisfied, for all \( i \), so that for \( 1 \leq j \leq k \),

- \( \exists r, C_i(b_j^i) \) is satisfied for all \( l = 1, \ldots, j-1, j+1, \ldots, k \)

and

- the assertion \( r_j(b_j^i, a_j^i) \) is in \( A_r \).

Thus, \( (\exists r_1, C_1 \cap \cdots \cap \exists r_k, C_k)(b_j^i) \) is satisfied iff \( C_j(a_j^i) \) is satisfied, for all \( j \), with the following:

1. \( h(q_i, b_i^1, \ldots, h(q_i, b_i^k) \) for all \( i \), with the following:
   - \( r_j(b_j^i, a_j^i, r_1(b_j^1, d_1), \ldots, r_{j-1}(b_j^{j-1}, d_{j-1}), \ldots, r_k(b_k^j, d_k) \) for all \( i \) and \( 1 \leq j \leq k \).

This finishes the modified construction. Again, it is not hard to prove correctness.

It remains to note that, when \( T \) is an \( ALCFI \)-TBox, then the above construction of \( q \) yields an ELQ instead of an ELIQ.

\[ \square \]

C Proofs for Section 4

Lemma 14.
Every Horn-\( ALCFI \)-TBox is unraveling tolerant.

Proof. We give a characterization of the entailment of ELIQs in the presence of Horn-\( ALCFI \)-TBoxes which is in the spirit of the rule-based (sometimes also called consequence-driven) algorithms commonly used for Horn description logics such as \( EL^+ \) and Horn-\( SHIQ \), see e.g. (Baader, Brandt, and Lutz 2005; Kazakov 2009; Krötzsch 2010).

In the characterization, we use extended ABoxes, i.e., finite sets of assertions \( C(a) \) with \( C \) a potentially compound concept and \( r(a, b) \). An \( ELIQ \)-concept is a concept that is formed according to the second syntax rule in the definition of Horn-\( ALCFI \). For an extended ABox \( A' \) and an assertion \( C(a) \), \( C \) an \( ELIQ \)-concept, we write \( A' \vdash C(a) \) if \( A' \) syntactically entails \( C(a) \), formally:

- \( A' \vdash \top(a) \) is unconditionally true;
- \( A' \vdash \bot(a) \) if \( \bot(b) \in A' \) for some \( b \in \text{Ind}(A) \);
- \( A' \vdash A(a) \) if \( A(a) \in A' \);
- \( A' \vdash C \cap D(a) \) if \( A' \vdash C(a) \) and \( A' \vdash D(a) \);
- \( A' \vdash C \sqcup D(a) \) if \( A' \vdash C(a) \) or \( A' \vdash D(a) \);
- \( A' \vdash \exists r.C(a) \) if there is an \( r(a, b) \in A' \) such that \( A' \vdash C(b) \).

Now for the characterization. Let \( T = \{ \top \subseteq C_T \} \) be a Horn-\( ALCFI \)-TBox and \( A \) a potentially infinite ABox (so that we can also apply the construction to unravelings of ABoxes). We produce a sequence of extended ABoxes \( A_0, A_1, \ldots \), starting with \( A_0 = A \cup \{ T(\alpha_T) \} \), where \( \alpha_T \) is a fresh individual which, intuitively, is a representative for all individual names that do not occur in \( A \). In what follows, we use additional individual names of the form \( ar_1C_1 \cdots C_k \) with \( a \in \text{Ind}(A_0), r_1, \ldots, r_k \) roles that occur in \( T \), and \( C_1, \ldots, C_k \in \text{sub}(T) \). We assume that \( N_0 \) contains such names as needed and use the symbol \( a \) also to refer to individual names of this compound form. Each extended ABox \( A_{i+1} \) is obtained from \( A_i \) by applying the following rules:

R1. if \( a \in \text{Ind}(A_i) \), then add \( C_T(a) \).
R2. if \( C \cap D(a) \in A_i \), then add \( C(a) \) and \( D(a) \);
R3. if \( C \rightarrow D(a) \in A_i \) and \( A_i \vdash C(a) \), then add \( D(a) \);
R4. if \( \exists r.C(a) \in A_i \) and \( \text{func}(r) \notin T \), then add \( r(a, arC) \) and \( C(arC) \);
R5. if \( \exists r.C(a) \in A_i \) and \( \text{func}(r) \in T \), and \( r(a, b) \in A_i \), then add \( C(b) \);
R6. if \( \exists r.C(a) \in A_i \) and \( \text{func}(r) \notin T \), and there is no \( r(a, b) \in A_i \), then add \( r(a, arC) \) and \( C(arC) \);
R7. if \( \forall r.C(a) \in A_i \) and \( r(a, b) \in A_i \), then add \( C(b) \).

We call \( A_c = \bigcup_{i \geq 0} A_i \) the completion of the original ABox \( A \). Note that \( A_c \) may be infinite even if \( A \) is finite, and that none of the above rules is applicable in \( A_c \). In the following, we write ‘\( A_c \vdash \bot(a) \) instead of ‘\( A_c \vdash \bot(a) \) for some \( a \in N_0 ^c \).

Claim 1. For all ELIQs \( C(a) \), we have
1. \( (T, A) \models C(a) \) iff \( A_c \vdash C(a) \) or \( A_c \vdash \bot(a) \);
2. \( (T, A) \models C(a) \) iff \( A_c \vdash C(a_T) \) or \( A_c \vdash \bot \) whenever \( a \in N_0 \setminus \text{Ind}(A) \).

We only sketch the proof. For the “if” directions, the central observation is that for any model \( I \) of \( T \) and \( A \), we can construct a homomorphism \( h \) from \( A_c \) to \( I \), i.e., \( h \) is a map from \( \text{Ind}(A_c) \) to \( \Delta_T ^I \) such that the following conditions are satisfied:

(a) \( h(a) = a \) for all \( a \in \text{Ind}(A) \);
(b) if \( C(a) \in A_c \), then \( h_i(a) \in C_T^I \);
(c) if \( r(a, b) \in A_c \), then \( (h_i(a), h_i(b)) \in r_T^I \).
More specifically, we inductively construct homomorphisms $h_i$ from $A_i$ to $I$, that satisfy Conditions (a) to (c) above with $A_i$ replaced by $A_i$, and such that $h_0 \leq h_1 \leq \cdots$. Then $h = \bigcup_{i \geq 0} h_i$ is the required homomorphism from $A_c$ to $I$.

Let $C(a)$ be an ELIQ. If $A_c \vdash \bot$, the existence of a homomorphism $h$ from $A_c$ into any model $I$ of $\mathcal{T}$ and $\mathcal{A}$ implies that $A$ is inconsistent w.r.t. $\mathcal{T}$, whence $(\mathcal{T}, A) \models C(a)$. If $A_c \vdash C(a)$, then preservation of ELIQs under homomorphisms also yields $(\mathcal{T}, A) \models C(a)$. For Point 2, assume $A_c \vdash C(\alpha T)$. We can construct the above homomorphisms $h$ such that $h(\alpha T) = a$. Thus, we again obtain $(\mathcal{T}, A) \models C(a)$.

For the “only if” direction of Point 1, we have to show that if $A_c \not\models C(a)$, where $C(a)$ is an ELIQ, and $A_c \not\vdash \bot$, then $(\mathcal{T}, A) \not\models C(a)$ (and similarly for Point 2). Define an interpretation $I$ as follows:

$$
\Delta^I = \text{Ind}(A_c)
$$

$$
A^I = \{ a \mid A(a) \in A_c \} \quad \text{for all } A \in N_c
$$

$$
r^I = \{ r(a, b) \mid r(a, b) \in A_c \} \quad \text{for all } r \in N_r
$$

$$
\alpha^I = a \quad \text{for all } a \in \text{Ind}(A)
$$

$$
\alpha^I = a^I \quad \text{for all } a \in N_i \setminus \text{Ind}(A)
$$

It can be shown that $I$ is a model of $A_c$ (thus $A$) and $\mathcal{T}$ and that $A_c \not\models C(a)$ implies $I \not\models C(a)$. Thus $(\mathcal{T}, A) \not\models C(a)$ as required.

We now consider the application of the above completion construction to both the original ABox $A$ and its unraveling $A^\nu$. Recall that individuals in $A^\nu$ are of the form $a_0 \tau_0 a_1 \cdots \tau_{n-1} a_n$, thus individuals in $A_c^\nu$ are of the form $a_0 \tau_0 a_1 \cdots \tau_{n-1} a_n \cdot s_1 C_1 \cdots s_k C_k$. For $\alpha \in \text{Ind}(A_c)$ and $\beta \in \text{Ind}(A_c^\nu)$, we write $\alpha \sim \beta$ if

$$
\alpha = a_n s_1 C_1 \cdots s_k C_k \quad \text{and} \quad
\beta = a_0 \tau_0 a_1 \cdots \tau_{n-1} a_n s_1 C_1 \cdots s_k C_k
$$

for some $a_0, \ldots, a_n, \tau_0, \ldots, \tau_{n-1}, s_1, \ldots, s_k, C_1, \ldots, C_k$. This includes the case where $k = 0$, i.e., the $s_1 C_1 \cdots s_k C_k$ component is empty in both $\alpha$ and $\beta$. The following claim can be shown by induction on $i$.

Claim 2. For all $\alpha \in \text{Ind}(A_c)$ and $\beta \in \text{Ind}(A_c^\nu)$ with $\alpha \sim \beta$, we have

1. $A_i \vdash C(\alpha)$ iff $A_i^\nu \vdash C(\beta)$ for all $\mathcal{ELC}$-concepts $C$;
2. $C(\alpha) \in A_i$ iff $C(\beta) \in A_i^\nu$ for all $C \in \text{sub}(\mathcal{T})$.

From Claims 1 and 2, we obtain that $A$ and $A^\nu$ entail exactly the same ELIQs. It follows that $\mathcal{T}$ is unraveling tolerant.

Lemma 17. Every unraveling tolerant $\mathcal{ALCFI}$-TBox is materializable.

Proof. We show the contrapositive using a proof strategy that is very similar to the second step in the proof of Theorem 11. Thus, take an $\mathcal{ALCFI}$-TBox $\mathcal{T}$ that is not materializable. By Theorem 9, $\mathcal{T}$ does not have the disjunction property. Thus, there is an ABox $\mathcal{A}_v$ and ELIQs $C_0(a_0), \ldots, C_k(a_k)$ such that $(\mathcal{T}, \mathcal{A}_v) \not\models C_0(a_0) \vee \cdots \vee C_k(a_k)$, but $(\mathcal{T}, \mathcal{A}_v) \models C_i(a_i)$ for all $i \leq k$. Let $A_i$ be $C_i$ viewed as a tree-shaped ABox with root $a_i$, for all $i \leq k$. Assume w.l.o.g. that none of the ABoxes $\mathcal{A}_v, A_0, \ldots, A_k$ share any individual names. Consider the ABox

$$
\mathcal{A} = \mathcal{A}_v \cup \mathcal{A}_0 \cup \cdots \cup \mathcal{A}_k \cup \{ (r(b, b_0), \ldots, r(b, b_k)) \cup \{ r_0(b_0, b_0), \ldots, r_{j-1}(b_j, b_{j-1}), r_{j+1}(b_j, b_{j+1}), \ldots, r_k(b_k, a_k) \}
$$

where $b$ is a fresh individual name and $r, r_0, \ldots, r_k$ do not occur in $\mathcal{T}$, and the ELIQ

$$
q = \exists r. (\exists r_0.C_0 \cap \cdots \cap \exists r_k.C_k)(b).
$$

Then we have

Claim. $(\mathcal{T}, \mathcal{A}) \models q$, but $(\mathcal{T}, \mathcal{A}^\nu) \not\models q$.

Proof. “$(\mathcal{T}, \mathcal{A}) \models q$”. Take a model $I$ of $\mathcal{T}$ and $\mathcal{A}$. By construction of $\mathcal{A}$, we have $b_i^I \in (\exists r_0.C_0)^I \forall i \neq j$. Due to the inclusion of $\mathcal{A}_v$, and since $(\mathcal{T}, \mathcal{A}_v) \models C_0(a_0) \vee \cdots \vee C_k(a_k)$, we find one $b_i$ such that $b_i^I \in (\exists r_i.C_i)^I$. Consequently, $I \models q$.

“$(\mathcal{T}, \mathcal{A}^\nu) \not\models q$” (sketch). Consider the elements $br(b_i, a_i)$ in $\mathcal{A}^\nu$. Each such element is the root of a copy of the unraveling $A^\nu$ of $A$, restricted to those individuals in $A_v$ that are reachable from $a_i$. Since $(\mathcal{T}, \mathcal{A}_v) \not\models C_0(a_0)$, we find a model $I_i$ of $\mathcal{T}$ and $\mathcal{A}_v$ with $b_i^I \notin (\exists r_i.C_i)^I$. By unraveling $\mathcal{I}$, we obtain a model $I^\nu_i$ of $\mathcal{T}$ and $\mathcal{A}^\nu$ with $b_i^{I^\nu_i} \notin (\exists r_i.C_i)^I$. By combining the models $I_0^\nu, \ldots, I_k^\nu$, one can craft a model $I$ of $\mathcal{T}$ and $\mathcal{A}^\nu$ such that $br(b_i, a_i) \notin C_i$ for all $i \leq k$. Consequently, $I \not\models q$.

It follows that $\mathcal{T}$ is not unraveling tolerant.

Theorem 16. If an $\mathcal{ALCFI}$-TBox $\mathcal{T}$ is unraveling tolerant, then PEQ-answering w.r.t. $\mathcal{T}$ is in PTIME.

To prove Theorem 16, let $\mathcal{T} = \{ T \subseteq C_T \}$ be an unraveling tolerant TBox, where we assume w.l.o.g. that $C_T$ is built from the constructors $\neg, \cap$, and $\exists r.C$, only. By Theorem 4, it suffices to show that ELIQ-answering w.r.t. $\mathcal{T}$ is in PTIME. Thus, let $q = C_0(a_0)$ be an ELIQ. We use $cl(\mathcal{T}, q)$ to denote the set of subconcepts of $\mathcal{T}$ and $q$, closed under single negation. For an interpretation $I$ and $d \in \Delta^I$, we use $t^I_{\mathcal{T}, q}(d)$ to denote the set of concepts $C \in cl(\mathcal{T}, q)$ such that $C \in d^I$. A $\mathcal{T}$, $q$-type is a subset $t \subseteq cl(\mathcal{T}, q)$ such that for some model $\mathcal{I}$ of $\mathcal{T}$, we have $t = t^I_{\mathcal{T}, q}(d)$. We use $tp(\mathcal{T}, q)$ to denote the set of all $\mathcal{T}, q$-types. For $t, t' \in tp(\mathcal{T}, q)$ and $r$ a role, we write $t \rightsquigarrow_r t'$ if the following conditions are satisfied:

- if $C \subseteq t'$ then $\exists r.C \subseteq t$, for all $\exists r.C \in cl(\mathcal{T}, q)$;
- if $C \subseteq t$ then $\exists r.C \subseteq t'$, for all $\exists r.C \in cl(\mathcal{T}, q)$;
- $\exists r.C \subseteq t$ iff $C \subseteq t'$, for all $\exists r.C \in cl(\mathcal{T}, q)$ with func($r$) $\subseteq \mathcal{T}$;
- $\exists r.C \subseteq t'$ iff $C \subseteq t$, for all $\exists r.C \in cl(\mathcal{T}, q)$ with func($r$) $\subseteq \mathcal{T}$. 
A type assignment is a map \( \text{Ind}(\mathcal{A}) \to 2^{\mathbb{N}(\mathcal{T}, q)} \). The \( \text{PTIME} \) algorithm for checking, given an ABox \( \mathcal{A} \), whether \((\mathcal{T}, \mathcal{A}) \models q\) is based on the computation of a sequence of type assignments \( \pi_0, \pi_1, \ldots \) as follows. For every \( a \in \text{Ind}(\mathcal{A}), \pi_0(a) \) is the set of all types \( t \in \text{tp}(\mathcal{T}, q) \) such that \( A(a) \in \mathcal{A} \) implies \( A \in t \). Then, \( \pi_{i+1}(a) \) is defined as the set of all \( t \in \pi_i(a) \) such that for all \( r(a, b) \in A \), \( r \) a role name or the inverse thereof, there is a type \( t_b \in \pi_i(b) \) such that \( t_a \triangleright_r t_b \).

Clearly, the sequence \( \pi_0, \pi_1, \ldots \) will stabilize after at most \( O(|\mathcal{A}|) \) steps and can be computed in time polynomial in \( |\mathcal{A}| \) (since \( |\mathcal{T}| \) and thus \( |\text{tp}(\mathcal{T}, q)| \) is a constant). Let \( \pi \) be the final type assignment in the sequence. The following yields Theorem 16.

**Lemma 31.** \((\mathcal{T}, \mathcal{A}) \models q \iff C_0 \in t \forall t \in \pi(a_0)\).

**Proof.** By unrolling tolerance, we have \((\mathcal{T}, \mathcal{A}) \models q \iff (\mathcal{T}, \mathcal{A}^u) \models q \iff \mathcal{I} \models q \iff \text{tp}(\mathcal{T}, q,d) = t\).

It thus suffices to show that for all \( t \in \text{tp}(\mathcal{T}, q) \), we have \( t \in \pi(a_0) \) iff there is a model \( \mathcal{I} \) of \( \mathcal{T} \) and \( \mathcal{A}^u \) with \( \text{tp}_{\mathcal{T}, q}(a_0^\alpha) = t \).

\(\Rightarrow\). Let \( \mathcal{I} \) be a model of \( \mathcal{T} \) and \( \mathcal{A}^{u} \) with \( \text{tp}_{\mathcal{T}, q}(a_0^\alpha) = t \). It is not hard to show by induction on \( i \) that for all \( i \geq 0 \) and all \( a_0 \cdots a_k \in \text{Ind}(\mathcal{A}^u) \), we have \( t_{\mathcal{T}, q}(a_0^\alpha) \in \pi_i(a_k) \). In particular, this implies that \( t_{\mathcal{T}, q}(a_0^\alpha) \in \pi(a_0^\alpha) \).

\(\Leftarrow\). Let \( t \in \pi(a_0) \). We build a model \( \mathcal{I} \) of \( \mathcal{T} \) and \( \mathcal{A}^{u} \) such that \( t_{\mathcal{T}, q}(a_0^\alpha) = t \). As first, construct a map \( \lambda : \text{Ind}(\mathcal{A}^u) \to \text{tp}(\mathcal{T}, q) \) such that for all \( a_0 \cdots a_k \in \text{Ind}(\mathcal{A}^u) \), we have \( \lambda(a_0 \cdots a_k) \in \pi(a_k) \). Start with setting \( \lambda(a_0) = t \).

Then exhaustively apply the following steps, where \( r \) is a role name:

- If \( \lambda(a_0 \cdots a_k) \) is defined, \( r(a_k, a_{k+1}) \in A \), and \( \lambda(a_0 \cdots a_k r a_{k+1}) \) is undefined, then by definition of the sequence \( \pi_0, \pi_1, \ldots \) and since \( \lambda(a_0 \cdots a_k) \in \pi(a_k) \), there is a type \( t' \in \pi(a_{k+1}) \) such that \( \lambda(a_0 \cdots a_k) \triangleright_r t' \).

- If \( \lambda(a_0 \cdots a_k) \) is defined, \( r(a_{k+1}, a_k) \in A \), and \( \lambda(a_0 \cdots a_k r^{-1} a_{k+1}) \) is undefined, then by definition of the sequence \( \pi_0, \pi_1, \ldots \) and since \( \lambda(a_0 \cdots a_k) \in \pi(a_k) \), there is a type \( t' \in \pi(a_{k+1}) \) such that \( \lambda(a_0 \cdots a_k) \triangleright_r t' \).

By definition of types, for each \( \alpha \in \text{Ind}(\mathcal{A}^u) \) we find a tree-shaped model \( \mathcal{I}_\alpha \) of \( \mathcal{T} \) and \( \mathcal{A}^u \) and a \( d_\alpha \in \Delta^{\mathcal{T}, \mathcal{I}} \) such that \( t_{\mathcal{T}, q}(d_\alpha^\alpha) = \lambda(\alpha) \).

Assume w.l.o.g. that the domains of all these models \( \Delta^{\mathcal{T}, \mathcal{I}} \) are disjoint. Define a new interpretation \( \mathcal{I}' \) as follows:

- \( (\mathcal{T}, \mathcal{A}) \models q \iff \exists \mathcal{I}' \in \Xi \models \text{Hom}(\mathcal{J}', \mathcal{J}) \)

**Theorem 32.** Let \( \mathcal{T} \) be a ALC\text{-}TBox. Then \( \mathcal{T} \) is unrolling tolerant iff all \( \mathcal{T}, \mathcal{A} \) models of the ELIQ have tree obstructions.

**Proof.** We use the notation from Theorem 24. The main observation is that if there is a homomorphism from a tree interpretation \( \mathcal{I} \) to an ABox (regarded as an interpretation), then there is a homomorphism from \( \mathcal{I} \) to the unrolling of the ABox (regarded as an interpretation). We now give the details.

Assume \( \mathcal{T} \) is unrolling tolerant. Let \( q = C(a) \) be an ELIQ. Let \( \Sigma = \text{sig}(\mathcal{T}) \cup \text{sig}(C) \cup \{P\} \). For every ABox \( \mathcal{A} \), we have \( (\mathcal{T}, \mathcal{A}) \models C(a) \iff (\mathcal{T}, \mathcal{A}^u) \models C(a) \).

By compactness, for every \( \alpha \) with \( (\mathcal{T}, \mathcal{A}) \models C(a) \), there exists a finite \( \mathcal{A}_f \subseteq \mathcal{A}^u \) such that \( (\mathcal{T}, \mathcal{A}_f) \models C(a) \). From \( \mathcal{A}_f \) we obtain \( \mathcal{A}_f^\Lambda \) by adding \( P(a) \) to \( \mathcal{A}_f \) and removing all other occurrences of \( P \). Now let \( \Xi_q \) denote the set of all \( \Delta_y \mathcal{A}_f \).

We show that \( \Xi_q \) satisfies the conditions for tree obstructions for the template \( \mathcal{T}_q \) as follows:

- Assume \( \text{Hom}(\mathcal{J}, \mathcal{I}_q) \). Let \( \mathcal{A} \) be an ABox with \( \mathcal{J} = \mathcal{I}_q \). Then \( (\mathcal{T}, \mathcal{A}) \models \exists \mathcal{P}(x) \land C(x) \). By materializability, there exists \( a \in \text{Ind}(\mathcal{A}) \) with \( P(a) \in \mathcal{A} \) and \( (\mathcal{T}, \mathcal{A}) \models C(a) \). Hence \( \mathcal{I}_q \mathcal{A} \mathcal{P}_E \in \Xi_q \) and clearly \( \text{Hom}(\mathcal{J}_q, \mathcal{I}_q) \), as required.

Conversely, assume \( \text{Hom}(\mathcal{I}_q, \mathcal{J}_q) \) for some \( \mathcal{I}_q \mathcal{A} \mathcal{P}_E \in \Xi_q \). We have \( (\mathcal{T}, \mathcal{A}) \models C(a) \). Hence not \( \text{Hom}(\mathcal{I}_q, \mathcal{I}_q) \). But then \( \text{Hom}(\mathcal{J}, \mathcal{I}_q) \), as required.
Claim 1. For all $\alpha, \beta \in \text{Ind}(A^u)$ with $\text{tail}(\alpha) = \text{tail}(\beta)$ and all $\mathcal{ALCFI}$-concepts $C$, we have $A^u \models C(\alpha)$ iff $A^u \models C(\beta)$.

Assume to the contrary that there are $\alpha, \beta \in \text{Ind}(A^u)$ with $\text{tail}(\alpha) = \text{tail}(\beta)$, $A^u \models C(\alpha)$, and $A^u \not\models C(\beta)$. Then there is a model $\mathcal{I}$ of $A^u$ and $\mathcal{T}$ such that $\mathcal{I} \not\models C(\alpha)$, in contradiction to $A^u \models C(\alpha)$.

Define a map $\iota : \text{Ind}(A^u) \to \text{Ind}(A^u)$ such that $\text{tail}(\iota(\gamma)) = \text{tail}(\gamma)$ for all $\gamma \in \text{Ind}(A^u)$ as follows:

1. Start with setting $\iota(\alpha) = \beta$;
2. if $\iota(\gamma)$ is defined, $\gamma = a_0r_0a_1 \cdots a_{n-1}r_{n-1}a_n$,
   $\iota(\alpha \cdot a_{n-1})$ is undefined, and $\iota(\gamma)$ is of the form $b_0s_0b_1 \cdots b_{m-1}s_{m-1}b_m$ with $s_{m-1} = r_{n-1}$ and $b_{m-1} = a_{n-1}$, then set $\iota(\alpha \cdot a_{n-1}) = b_0 \cdots b_{m-1};$
3. if $\iota(\gamma)$ is defined, $\gamma = a_0r_0a_1 \cdots a_{n-1}r_{n-1}a_n$,
   $\iota(\alpha \cdot a_{n-1})$ is undefined, and $\iota(\gamma)$ is not of the form $b_0s_0b_1 \cdots b_{m-1}s_{m-1}b_m$ with $s_{m-1} = r_{n-1}$ and $b_{m-1} = a_{n-1}$, then $\iota(\gamma)r_{n-1}a_{n-1} \in \text{Ind}(A^u)$ and we use it as the value of $\iota(\alpha \cdot a_{n-1});$
4. if $\iota(\gamma)$ is defined, $\gamma \alpha \in \text{Ind}(A^u)$, $\iota(\gamma \alpha)$ is undefined, and $\iota(\gamma)$ is of the form $a_0 \cdots a_{n-1}r_{n-1}a_n$ with $r_{n-1} = r^\alpha$ and $a_{n-1} = a$, then set $\iota(\gamma \alpha) = a_0 \cdots a_{n-1};$
5. if $\iota(\gamma)$ is defined, $\gamma \alpha \in \text{Ind}(A^u)$, $\iota(\gamma \alpha)$ is undefined, and $\iota(\gamma)$ is of the form $a_0 \cdots a_{n-1}r_{n-1}a_n$ with $r_{n-1} = r^\alpha$ and $a_{n-1} = a$, then $\iota(\gamma \alpha) \in \text{Ind}(A^u)$ and we use it as the value of $\iota(\gamma \alpha);$ 
6. if the value of $\iota(\gamma)$ is undefined after exhaustive application of the above rules, set $\iota(\gamma) = \gamma$.

It can be verified that $\iota$ is an ABox automorphism, i.e., for all $\gamma \in \text{Ind}(A^u)$, $A \models C_{\mathcal{N}}$, and $r \in R_{\mathcal{N}}$, we have

- $A(\gamma) \in A^u$ if $A(\iota(\gamma)) \in A^u$;
- $r(\gamma, \gamma') \in A^u$ if $r(\iota(\gamma), \iota(\gamma')) \in A^u$.

Let the interpretation $\mathcal{J}$ be defined as $\mathcal{I}$, but put $\gamma^J = \iota(\gamma)^J$ for all $\gamma \in \text{Ind}(A^u)$. $\mathcal{J}$ is a model of $A$ since $\iota$ is an ABox automorphism and a model of $\mathcal{T}$ since $\mathcal{I}$ is. Moreover, $\mathcal{I} \not\models C(\beta)$ implies $\beta^J \not\in C^J$, which implies $\beta^J \not\in C^J$ by definition of $\mathcal{J}$. Since $\alpha^J = \beta^J$, we have $\mathcal{J} \not\models C(\alpha)$ as required. This finishes the proof of Claim 1.

Using Claim 1, we exhibit some self-similarity also inside $I^u$. However, we cannot use $\mathcal{ALCFI}$-concepts here since entailment by $A^u$ agrees with truth in $I^u$ only for ELIQs, but not for $\mathcal{ALCFI}$-instance queries. We thus concentrate on $\mathcal{ELL}$-concepts and $\mathcal{BL}$-concepts, where the latter are constructed only from concept names and the Boolean operators.

**Claim 2.** For all $\alpha, \beta \in \text{Ind}(A^u)$ with $\text{tail}(\alpha) = \text{tail}(\beta)$, we have

1. $\alpha \in C^u$ iff $\beta \in C^u$ for all $\mathcal{ELL}$-concepts $C$ and
2. $\alpha \in C^u$ iff $\beta \in C^u$ for all $\mathcal{BL}$-concepts $C$.

Point 1 is an immediate consequence of Claim 1 and the fact that $I^u$ is an ELIQ-materialization of $A^u$. For Point 2, note that Point 1 yields $\alpha \in I^u_{\mathcal{B}}$ iff $\beta \in I^u_{\mathcal{B}}$ for all concept names $A$. Point 2 then follows by a straightforward induction on the structure of $C$.

Now for the announced uniformization of $I^u$. What we want to achieve is that for all $\alpha, \beta \in \text{Ind}(A^u)$, $\text{tail}(\alpha) = \text{tail}(\beta)$ implies $I_{\alpha} = I_{\beta}$ (recall that $I_{\alpha}$ is the tree component of $I^u$ rooted at $\alpha$, and likewise for $I_{\beta}$). Construct the interpretation $J^u$ as follows:
• for each $\alpha \in \text{Ind}(A^u)$ with $\text{tail}(\alpha) = a$, take a copy $J_\alpha$ of $I_\alpha$ with root $\alpha$;

• then $J^u$ is the disjoint union of all interpretations $J_\alpha$, $\alpha \in \text{Ind}(A^u)$, extended with a role edge $(\alpha, \beta) \in r^{J^u}$ whenever $r(\alpha, \beta) \in A^u$.

It is straightforward to verify that $J^u$ is a model of $A^u$: all role assertions are satisfied by construction of $J^u$; moreover, $A(\alpha) \in A^u$ implies $A(a) \in A^u$ where $a = \text{tail}(\alpha)$, thus $\alpha \in A^u$; by construction of $J^u$, this yields $\alpha \in A^\Delta$ as required.

Next, we show that $J^u$ is a model of $T$. Let $f : \Delta^J \to \Delta^u$ be a mapping that assigns to each domain element of $J^u$ the original element in $T^u$ of which it is a copy.

**Claim 3.** For every $d \in \Delta^J$ and $\mathcal{ALCFL}$-concept $C$ of depth one, we have $d \in C^\Delta$ iff $f(d) \in C^T$.

The proof of claim 3 is by induction on the structure of $C$. We assume w.l.o.g. that $C$ is built only from the constructors $\neg, \cap$, and $\exists r.C$. The base case, where $C$ is a concept name, is an immediate consequence of the definition of $I$. The case where $C = \neg D$ and $C = D_1 \cap D_2$ is routine. Thus we concentrate on the case where $C = \exists r.D$, where $r \in N_R \cup N_R'$.

First let $d \in C^\Delta$. Then there is a $(d, e, d') \in r^{J^u}$ with $e \in D^{J^u}$. First assume that the edge $(d, e)$ was added to $r^{J^u}$ because $d = \alpha$ and $e = \beta$ for some $\alpha, \beta \in \text{Ind}(A^u)$ with $r(\alpha, \beta) \in A^u$. Let $\text{tail}(\alpha) = a$ and $\text{tail}(\beta) = b$. Then we have $f(\alpha) = a$ and $f(\beta) = b$. By construction of $A^u$, $r(\alpha, \beta) \in A^u$ implies that $\beta = arb$ or $\alpha = br-\alpha$. In both cases we have $r(a, b) \in A$, thus $r(a, arb) \in A^u$, thus $(a, arb) \in r^{J^u}$. Since $\beta = e \in D^\Delta$, $\text{IH}$ yields that $b \in D^\Delta$. Since $C$ is of depth one, $D$ is a $\mathcal{BL}$-concept. By Point 2 of Claim 2, $arb \in D^{J^u}$ and we are done. Now assume that there is an $\alpha \in \text{Ind}(A^u)$ such that $(d, e) \in J_\alpha$. By construction of $J^u$, we then have $(f(d), f(e)) \in r^{J^u}$ and $\text{IH}$ yields $f(e) \in D^\Delta$.

Now let $f(d) \in C^T$. Then there is a $(d, e, d') \in r^{J^u}$ with $e \in D^{J^u}$. First assume that $f(d) = \alpha$ and $e = \beta$ for some $\alpha, \beta \in \text{Ind}(A^u)$ with $r(\alpha, \beta) \in A^u$. Since $f(d) \in \text{Ind}(A^u)$, we must have $d = \gamma \in \text{Ind}(A^u)$ and $f(d) = a \in \text{Ind}(A^u)$ with $\text{tail}(\gamma) = a$. By construction of $A^u$, $r(\alpha, \beta) \in A^u$ implies that $\beta = arb$, thus $r(a, b) \in A$, thus $r(\gamma, \beta) \in A^u$ with $(i) \delta = \gamma rb$ or $(ii) \gamma = \delta r^{-1} a$ and $\text{tail}(\delta) = b$. Since $arb \in D^{J^u}$, Point 2 of Claim 2 yields $b \in D^\Delta$. Since $\text{tail}(\delta) = b$ implies $f(\delta) = b$, $\text{IH}$ yields $\delta \in D^\Delta$ and we are done. Now assume that there is an $\alpha \in \text{Ind}(A^u)$ such that $(f(d), e) \in I_\alpha$. By construction of $J^u$, $f(d)$ being in $I_\alpha$ implies that $\alpha = a$ for some $\alpha \in \text{Ind}(A)$ and that there is an $\alpha' \in \text{Ind}(A^u)$ such that $d$ is in $J_{\alpha'}$ and $\text{tail}(\alpha') = a$. Again by construction of $J^u$, we thus find an $e' \in J_{\alpha'}$, with $f(e') = e$ and $(d, e') \in r^{J^u} \subseteq r^{J^u}$. $\text{IH}$ yields $e' \in D^\Delta$.

This finishes the proof of Claim 3. We can now show that $J^u$ is a model of $T$. First, $J^u$ satisfies all CLs in $T$ since $J^u$ does and by Claim 3. It remains to show that $I$ satisfies all functionality assertions in $T$. Thus, let $\text{func}(r) \in T$. We show that each $d \in \Delta^T$ has at most one $r$-successor in $J^u$. Distinguish two cases:

• $d \notin \text{Ind}(A^u)$. Then $d$ has at most one $r$-successor since $\Delta^I$ satisfies $\text{func}(r)$ and by construction of $J^u$.

• $d = a \in \text{Ind}(A^u)$. Let $\text{tail}(\alpha) = a$. By construction of $J^u$ and $A^u$, $\alpha$ has the same number of $r$-successors in $J^u$ as in $I^u$. Since $I^u$ satisfies $\text{func}(r)$, $\alpha$ can have at most one $r$-successor in $J^u$.

The final condition that $J^u$ should satisfy is that $J^u \not\models q = C_0(a_0)$. Assume to the contrary that $J^u \models q$. We view $q$ as a tree-shaped CQ whose root is the individual name $a_0$ and whose non-root nodes are variables, thus $J^u \models q$ means that there is a match $\pi$ of $q$ in $J^u$, i.e., a mapping $\pi : \text{term}(q) \to \Delta^J$ such that $\pi(a_0) = a_0$, $A(t) \in q$ implies $\pi(t) \in A^\Delta$, and $r(t, t') \in q$ implies $(\pi(t), \pi(t')) \in r^{J^u}$. We prove that this implies the existence of a match $\tau$ for $q$ in $I^u$, which yields a contradiction to $I^u \not\models q$.

We start the construction of $\tau$ by setting $\tau(t) = \pi(t)$ for all $t \in \text{term}(q)$ with $\pi(t) \in \text{Ind}(A^u)$. It remains to define $\tau(x)$ for all variables $x \in \text{term}(q)$ such that $\pi(x) \neq \alpha$ for all $\alpha \in \text{Ind}(A^u)$. This is done by applying the following construction, for each $t \in \text{term}(q)$ such that $\pi(t) = \alpha \in \text{Ind}(A^u)$.

Recall that $J_\alpha$ is the tree interpretation rooted at $\alpha$ in $J^u$. Let $V$ be the set of all variables $x \in \text{term}(q)$ such that there is a sequence $r_1(t_1, t_2), \ldots, r_{\pi^{-1}(a_0)}(t_{\pi^{-1}(a_0)}, t_{\pi^{-1}(a_0)}) \in q$, $r_i \in N_R \cup N_R'$, such that $t_1 = t$, $t_n = x$, and $\pi(t_i) \in \Delta^\Delta \setminus \{\alpha\}$ for $2 \leq i \leq n$. We define $\tau(x)$ for all $x \in V$ simultaneously.

To this end, let $J^\Delta_{\alpha}$ be the restriction of $J_\alpha$ to those elements that are in $V$. It is not hard to verify that $J^\Delta_{\alpha}$ is a finite tree and an initial piece of the potentially infinite tree $J_\alpha$. Let $C_V$ be an $\mathcal{ELI}$-concept that describes $J^\Delta_{\alpha}$ up to homomorphisms, i.e., for any interpretation $I$ and $d \in \Delta^\Delta$ we have $d \in C^I_V$ iff $J^\Delta_{\alpha}$ can be embedded into $I$ with a $\text{sig}(q)$-homomorphism (a homomorphism that ignores all symbols which do not occur in $q$) $h$ such that $h(\alpha) = d$. Let $\text{tail}(\alpha) = a$. By construction of $J^u$, the tree component $I_\alpha$ of $I^u$ is identical to $J_\alpha$ and thus has $J^\Delta_{\alpha}$ as an initial piece, which implies $d \in C^I_V$, and consequently there is a homomorphism $h$ that embeds $J^\Delta_{\alpha}$ into $I^u$ such that $h(\alpha) = a$. To define the match $\tau$ for the variables in $V$, compose $\tau$ with $h$.

It can be verified that the overall mapping $\tau$ obtain in this way is a match for $q$ in $I$.

This finishes the construction and analysis of the uniform model $J^u$. It remains to convert $J^u$ into a model $I$ of $T$ and the original ABox $A$ such that $I \not\models q$.

• take the disjoint union of the components $J_\alpha$ of $J^u$, for each $\alpha \in \text{Ind}(A)$;

• set $a^T = a$ for all $a \in \text{Ind}(A)$;

• add the edge $(a, b) \in r^{J^u}$ whenever $r(a, b) \in A$.

It is straightforward to verify that $I$ is a model of $A$: all role assertions are satisfied by construction of $I$; moreover, $A(a) \in A$ implies $A(a) \in A^u$, whence $a \in A^\Delta$ which in turn implies $a \in A^T$ by construction of $I$. To show that $I$ is a model of $T$, we first note that $J^u$ is self-similar in a way that parallels Claim 1.

**Claim 4.** For all $\alpha, \beta \in \text{Ind}(A^u)$ with $\text{tail}(\alpha) = \text{tail}(\beta)$ and
all $ALCFI$-concepts $C$, we have $\alpha \in C \setminus ^*r_n \iff \beta \in C \setminus ^*r_n$.

Proof sketch. The proof parallels the one of Claim 1. This time, we define an automorphism $\iota$ on the model $J^n$ instead of on the ABox $A^n$. For the elements $\text{Ind}(A^n) \subseteq \Delta J^n$, the construction of $\iota$ is exactly as in the proof of Claim 1. We can then extend the initial $\iota$ to all non-ABox-elements of $J^n$ exploiting the uniformity of this interpretation. Details are left to the reader.

Next, we show the following.

Claim 5. For every $d \in \Delta T$ and $ALCFI$-concept $C$, we have $d \in C \setminus ^*r_n$ iff $d \in C \setminus ^*r_e$.

The proof of Claim 5 is by induction on the structure of $C$. Again, the only interesting case is $C = \exists r.D$, where $r \in N_R \cup N_{\text{r}}$.

First assume $d \in C \setminus ^*r_n$. Then there is a $(d,e) \in r^{-}\text{a}$ with $e \in D^\text{a}$. First assume that the edge $(d,e)$ was added to $r^{-}\text{a}$ because $d = a$ and $e = \beta$ for some $\alpha, \beta \in \text{Ind}(A^n)$ and $r(\alpha, \beta) \in A^n$. Since $d \in \Delta T$, we must have $d = a = \alpha \in \text{Ind}(A)$. Let $\text{tail}(\beta) = b$. By construction of $A^n$, $r(\alpha, \beta) \in A^n$ thus yields $r(a,b) \in A$ and hence $(a,b) \in r^2$. We are done since Claim 4 and $\beta \in D^\text{a} \setminus \text{IH}$ yields $b \in D^\text{a} \setminus \text{IH}$, which implies $b \in D^2$.

Now let $d \in C \setminus ^*r_e$. Then there is a $(d,e) \in r^2$ with $e \in D^\text{a}$. First assume that $d = a$ and $e = b$ with $a, b \in \text{Ind}(A)$ and $r(a,b) \in A$. By construction of $A^n$, this implies that $r(a,arb) \in A^n$. Thus $(a,arb) \in r^{-}\text{a}$ and we are done since $\text{IH}$ yields $b \in D^\text{a}$ by Claim 4. Now assume that there is an $a \in \text{Ind}(A)$ such that $(d,e) \in r^{-}\text{a}$. Then the construction of $I$ yields $(d,e) \in r^{-}\text{a}$ and we are done since $\text{IH}$ yields $e \in D^\text{a}$.

By Claim 4, $I$ satisfies all CIs in $T$. To show that $I$ is a model of $T$, it remains to show that $I$ satisfies all functionality assertions in $T$. Thus, let $\text{func}(r) \in T$. We show that each $d \in \Delta J^n$ has at most one $r$-successor in $J^n$. Distinguishing two cases:

- $d \not\in \text{Ind}(A)$. Then $d$ has at most one $r$-successor since $J^n$ satisfies $\text{func}(r)$ and by construction of $I$.
- $d = a \in \text{Ind}(A)$. By construction of $I$ and $A^n$, $a$ has the same number of $r$-successors in $I$ as in $J^n$. Since $J^n$ satisfies $\text{func}(r)$, $a$ can have at most one $r$-successor in $I$.

It remains to show that $I \models q$. Assume to the contrary of what is to be shown that $I \models q$. Let $S \subseteq \Delta T \times \Delta J^n$ be the set of pairs $(d,e)$ such that for some $a \in \text{Ind}(A)$ and $\alpha \in \text{Ind}(A^n)$ with $\text{tail}(\alpha) = a$, $d \in \Delta J^n$ is the element in the tree interpretation $J_n$ that corresponds to $e \in \Delta J^n$ in the isomorphism tree interpretation $I_n$. Using the definition of $A^n$ and $\iota$, it can be verified that $S$ is an $i$-simulation from $I$ to $J^n$. We only prove that when $(a,b) \in r^2$ with $a, b \in \text{Ind}(A)$ and $(a, \alpha) \in S$, then there is a $\beta$ with $(\alpha, \beta) \in r^2$ and $(b, \beta) \in S$. To start, note that, by definition of $S$, we have $\alpha \in \text{Ind}(A^n)$ and $\text{tail}(\alpha) = a$. From $(a,b) \in r^2$, we obtain $r(a,b) \in A$ and thus by construction of $A^n$ there is a $\beta \in \text{Ind}(A^n)$ with $r(\alpha, \beta) \in A^n$ and $\text{tail}(\beta) = b$. From $r(\alpha, \beta) \in A^n$, we obtain $(\alpha, \beta) \in r^{-}\text{a}$. From $\text{tail}(\beta) = b$, it follows that $(b, \beta) \in S$ as required.

Since matches of ELIQs are preserved under $i$-simulations and $(a_0, a_0) \in S$, $I \models q$ implies $J^n \models q$, which is a contradiction.

\section{Proofs for Section 6}

\subsection{Proof of Lemma 21}

We show Lemma 21 for singleton sets $\Delta T$. The extension to arbitrary interpretations is straightforward. Thus, let $Z$ be a concept name and $z_0, z_1$ role names. Let $T = \{T \subseteq \exists z_0, T, T \subseteq \exists z_1, \exists z_2, Z\}$. It follows that $\Delta T = I \models q$ implies $J^n \models q$, which is a contradiction.

\subsection{Proof of Lemma 34}

For any ABox $A$ and set $I \subseteq \text{Ind}(A)$, one can construct a model $I$ of $(T, A)$ such that $H^2 = I$ and $I$ is hom-initial in $\text{Mod}(T, A)$.

Proof. Assume $A$ and $I \subseteq \text{Ind}(A)$ are given. Denote by $I_{\theta}$ the interpretation based on a binary tree in which every node has one $z_0$-son and one $z_1$-son, and every node reachable with $z_1$ satisfies $Z$. More precisely, the domain $\Delta T_{\theta}$ of $I_{\theta}$ is the set of words over $\{0, 1\}$, $(\sigma, \sigma) \in z_{00}^\theta$ for all $\sigma \in \Delta T_{\theta}$, $(\sigma, \sigma^1) \in z_{10}^\theta$ for all $\sigma \in \Delta T_{\theta}$, and $z_{10}^\theta = \{\sigma | \sigma \in \Delta T_{\theta}\}$. Now, hook mutually disjoint copies of $I_{\theta}$ to each $e \in \text{Ind}(A)$ (i.e., we identify the root of the copy of $I_{\theta}$ with $a^\theta$). The resulting interpretation, call it $I_{\theta}$, satisfies $T$ and $H^2 = \emptyset$. To satisfy the condition $H^2 = I$, we add for all $a \in I$ and $d$ with $(a,arb) \in \Delta J_n$ a new $d'$ to $I_{\theta}$ with $(d, d') \in z_{1}^\theta$ and $d' \not\in Z_{\theta}$. Also, hook a copy of $I_{\theta}$ to $d'$. The resulting interpretation, $I$, satisfies $T$ and we have $H^2 = I$. Now let $J$ be a model of $(T, A)$. To construct a homomorphism $f$, we set $f(a^2) = a^\theta$ for all $a \in \text{Ind}(A)$. Suppose $d \not\in a^2$ for any $a \in \text{Ind}(A)$ and $(d, d') \in z_{1}^\theta$. Set $f(d) = e$. (Observe that $d \not\in Z^\theta$.) If $(d', d) \in z_{00}^\theta$, by $T \subseteq \exists z_0, T$, we find $e$ with $(f(d'), e) \in z_{00}^\theta$. Set $f(d) = e$. One can show that the resulting function $f$ is a homomorphism.

\subsection{Proof of Theorem 24}

Assume $T$ and $C(a)$ are given. Similarly to Theorem 22, the interpretation $I_{\theta, q}$ can be obtained using a standard type-based construction. We use the sets $c(T, C)$, $tp(T, C)$, and the relation $\rightarrow$, between types as defined in the proof of Theorem 16. We define $\Delta T_{\theta, q}$ as the set of all $t \in tp(T, C)$ that are satisfiable w.r.t. $T$ and let $t \in A^\theta T_{\theta, q}$ iff $A \in t$, for all $A \in \Sigma$, and $(t^T, t') \in r^\theta T_{\theta, q}$ iff $t \rightarrow t'$, for all $r \in \Sigma$. Finally, $P^\theta T_{\theta, q} = \{t \in \Delta T_{\theta, q} \mid C \not\in t\}$. We now show:

1. $(T, A) \models C(a)$ iff not $\text{Hom}(I_{\theta, q}^A, I_{\theta, q}^A)$, where $A'$ results from $A$ by adding $P(a)$ to $A$ and removing all other assertions using $P$ from $A$;

2. not $\text{Hom}(I_{\theta, q}^A, I_{\theta, q}^A) \iff (T, A) \models \exists v. (P(v) \land C(v))$.

We start by proving (1).

\”$\Rightarrow$\”. Assume $\text{Hom}(I_{\theta, q}^A, I_{\theta, q}^A)$. Let $h : I_{\theta, q}^A \rightarrow I$ be a witness homomorphism. For each $b \in \text{Ind}(A)$, let $I_{b}^A$ be a copy of $I_{\theta, q}^A$ (with isomorphism $h_b : I_b \rightarrow I$). Hook each
\(I_b\) to \(A'\) by identifying \(b\) with \(b(h)\). The resulting interpretation, \(\mathcal{H}\), is the disjoint union of all \(I_b, b \in \text{Ind}(A)\) together with \((a,b) \in r^H\) whenever \(r(a,b) \in A\) and \(r \in \Sigma\). It is readily checked that

- \(\bigcup_{b \in \text{Ind}(A)} h_b\) is a \(\{P\}\)-bisimulation (two-way!) between \(\mathcal{H}\) and \(J\).

Thus, for all subconcepts \(D\) of \(T\) and \(C\) and all \(b \in \text{Ind}(A)\): \(b \in C^H\) iff \(b(h) \in C^{gT^*}\). We obtain that \(\mathcal{H}\) is a model of \(T\) and \(A\). Moreover, \(a \notin C^H\) since \(h(a) \notin C^{gT^*}\) and the latter follows because otherwise \(h(a) \notin P^{gT^*}\) and \(P(a) \in A'\) which would contradict that \(b\) is a homomorphism. Thus, \((T,A) \not\models C(a)\).

\("\subseteq\)". Assume \((T,A) \not\models C(a)\). Take a witness interpretation \(J\). The type \(t(d)\) of \(d \in \Delta^J\) is the set of (negated) subconcepts \(D\) of \(C\) and \(T\) such that \(d \in D^J\). The mapping \(h : a \mapsto t(a^J)\), for \(a \in \text{Ind}(A)\), is a homomorphism from \(\mathcal{I}_b^J\) to \(J\). We only consider preservation of \(P\). Assume \(P(b) \in A'\). Then \(a = b\). We have \(C \subseteq \text{Ind}(A)\). Thus \(C \notin h(a)\). Hence \(h(a) \notin P^{gT^*}\).

Consider \((2)\). The proof is similar.

\("\supseteq\)". Assume \((T,A) \not\models \exists v . (P(v) \land C(v))\). Take a witness interpretation \(J\). The type \(t(d)\) of \(d \in \Delta^J\) is the set of (negated) subconcepts \(D\) of \(C\) and \(T\) such that \(d \in D^J\). The mapping \(h : a \mapsto t(a^J)\), for \(a \in \text{Ind}(A)\), is a homomorphism from \(\mathcal{I}_b^J\) to \(J\). We only consider preservation of \(P\). Assume \(P(b) \in A'\). Then, since \((T,A) \not\models \exists v . (P(v) \land C(v))\), \(C \not\subseteq h(b)\). Hence \(h(a) \notin P^{gT^*}\).

\("\supseteq\)". Assume \(\text{Hom}(\mathcal{I}_b^J, \mathcal{I}_J)\). Let \(h : \mathcal{I}_b^J \rightarrow \mathcal{I}\) be a witness homomorphism. For each \(b \in \text{Ind}(A)\), let \(I_b\) be a copy of \(I\) (with isomorphism \(h_b : I_b \rightarrow I\)). Hook each \(I_b\) to \(A\) by identifying \(b\) with \(h(b)\). The resulting interpretation, \(\mathcal{H}\), is the disjoint union of all \(I_b, b \in \text{Ind}(A)\) together with \((a,b) \in r^H\) whenever \(r\) \(a,b\) \(\in A\) and \(r \in \Sigma\). For all concepts \(X\) that do not occur in \(T\) or \(C\) (including, in particular, \(P\)), we set \(X^H = \{b \in \text{Ind}(A) \mid X(b) \in A\}\). It is readily checked that

- \(\bigcup_{b \in \text{Ind}(A)} h_b\) is a \(\{P\}\)-bisimulation (two-way!) between \(\mathcal{H}\) and \(J\).

Thus, for all subconcepts \(D\) of \(T\) and \(C\) and all \(b \in \text{Ind}(A)\): \(b \in C^H\) iff \(h(b) \in C^{gT^*}\). Thus, \(\mathcal{H}\) is a model of \(T\) and \(A\). Moreover, \(P^H \cap C^H = \emptyset\); if \(d \in P^H\), then \(d = b^2\) for some \(b \in \text{Ind}(A)\) with \(P(b) \in A\). Thus, \(h(b) \notin P^{gT^*}\). But then \(h(b) \notin C^{gT^*}\). Therefore \(b \notin C\), as required.

It follows that \((T,A) \not\models \exists v . (P(v) \land C(v))\), as required.

**F Proofs for Section 7**

To formulate the result for FO-rewritability, we introduce a slightly modified form of FO-rewritability that takes into account only those ABoxes that are consistent w.r.t. the TBox.

**Definition 35.** Let \(T\) be a \(\text{ALCF}^\star\)-TBox. Let \(Q \in \{\text{CQ, PEQ, ELIQ, ELQ}\}\). We say that \(T\) is FO-rewritable for \(Q\) for consistent ABoxes iff for every \(q(\bar{x}) \in Q\) one can effectively construct a FOQ \(q'(\bar{x})\) such that for every ABox \(A\) that is consistent w.r.t. \(T\), \(\text{cert}_T(q,A) = \{\bar{a} \mid I_A \models q'(\bar{a})\}\).

Using similar modifications of Definition 2, one can define the obvious notions of \(Q\)-answering w.r.t. \(T\) being in PTIME for consistent ABoxes and \(Q\)-answering w.r.t. \(T\) being coNP-hard for consistent ABoxes. Theorem 4 still holds for these modified notions. For simplicity, we state the following result for CQs only.

We first prove an extended version of the undecidability result (Theorem 28) and then modify the TBoxes constructed in its proof to show the non-dichotomy result (Theorem 27). The modified version of Theorem 28 is as follows:

**Theorem 36.** For \(\text{ALCF}^\star\)-TBoxes \(T\), the following problems are undecidable (Points 1 and 2 are subject to the side condition that PTIME \(\neq\) NP):
1. \(\text{CQ}-\text{answering w.r.t. } T\) is in PTIME (with and w/o restriction to consistent ABoxes);
2. \(\text{CQ answering w.r.t. } T\) is coNP-hard; (with and w/o restriction to consistent ABoxes);
3. \(T\) is materializable;
4. \(T\) is FO-rewritable for \(Q\) for consistent ABoxes;

The proofs employ TBoxes that have been introduced in (Baader et al. 2010) to prove the undecidability of the following emptiness problem: given an \(\text{ALCF}^\star\)-TBox \(T\), a signature \(\Sigma\) with \(\Sigma \subseteq \text{sig}(T)\) and a concept name \(A\), does there exist a \(\Sigma\)-ABox \(A\) such that \(\text{is}\) consistent w.r.t. \(T\) and \((T,A) \models \exists v . A(v)\)? Note that this problem is of interest only for \(A \not\models \Sigma\) because otherwise one could clearly take the ABox \(\{A(a)\}\).

We start by defining the TBoxes \(T_{\mathcal{P}}\) constructed in (Baader et al. 2010). An instance of the finite rectangle tiling problem (FRTP) is given by a triple \(\mathcal{P} = (\Sigma, H, V)\) with \(\Sigma\) a non-empty, finite set of tile types including an initial tile \(T_{\text{init}}\) to be placed on the lower left corner and a final tile \(T_{\text{final}}\) to be placed on the upper right corner, \(H \subseteq \Sigma \times \Sigma\) a horizontal matching relation, and \(V \subseteq \Sigma \times \Sigma\) a vertical matching relation. A tiling for \((\Sigma, H, V)\) is a map \(f : \{0,\ldots,n\} \times \{0,\ldots,m\} \rightarrow \Sigma\) such that \(m \geq 0\) and \(f(0,0) = T_{\text{init}}, f(n,m) = T_{\text{final}}, f(i,j), f(i+1,j), f(i,j+1) \in H\) for all \(i < n\) and \(f(i,j), f(i,j+1) \in V\) for all \(i < m\).

It is undecidable whether an instance \(\mathcal{P}\) of the FRTP has a tiling. For simplicity, in the following we fix a set \(\Sigma = \{T_1,\ldots, T_p\}\) of tile types and consider instances of the FRTP over \(\Sigma\) only. It is easy to see that the tiling problem is still undecidable if \(\Sigma\) is sufficiently large.

Now let \(\Sigma = \{T_1,\ldots, T_p, x, y, x^-, y^-\}\) be a signature consisting of a set \(T_1,\ldots, T_p\) of concept names (identical to the tile types) and role names \(x, y, x^-, y^-\) (we are not assuming that \(x^-\) and \(y^-\) are interpreted as the inverse of \(x\) and \(y\), respectively). In (Baader et al. 2010), with any \(\mathcal{Q} = (\Sigma, H, V)\) one associates the \(\text{ALCF}^\star\)-TBox \(T_{\mathcal{Q}}\) containing

\[\mathcal{F} = \{\text{func}(x), \text{func}(y), \text{func}(x^-), \text{func}(y^-)\}\]

and CIs using additional concept names \(U, R, L, D, A, Y, I_x, I_y, C, Z_{c,1}, Z_{c,2}, Z_{x,1}, Z_{x,2}, Z_{y,1}\).

\(x\) and \(y\) are used to build the rectangle. \(U\) and \(R\) mark
We note that the final five inclusions (and the concept names inclusions are not required in the present proof, but are used here to fix the left and lower border of the rectangle. Those final inclusions are not used in (Baader et al. 2010). We use them here to fix the left and lower border of the rectangle. Those inclusions are not required in the present proof, but are used in the non-dichotomy proof below.

Call an ABox $A$ a $\Psi$-ABox (with initial node $a$) iff there is a tiling $f$ for $\Psi$ with domain $\{0, \ldots, n\} \times \{0, \ldots, m\}$ and a bijection $f_{\Psi} : \{0, \ldots, n\} \times \{0, \ldots, m\} \rightarrow \text{Ind}(A)$ with $f_{\Psi}(0, 0) = a$ such that

- $T_{\text{Init}}(f_{\Psi}(0, 0)) \in A$;
- $T_{\text{Final}}(f_{\Psi}(n, m)) \in A$;
- $T_i(f_{\Psi}(k, j)) \in A$ iff $f_i(k, j)$;
- $x(b_1, b_2) \in A$ iff $x^-(b_2, b_1) \in A$ iff $(b_1, b_2) = (f_{\Psi}(k, j), f_{\Psi}(k + 1, j))$;
- $y(b_1, b_2) \in A$ iff $y^-(b_2, b_1) \in A$ iff $(b_1, b_2) = (f_{\Psi}(k, j), f_{\Psi}(k, j + 1))$.

The following is shown in (Baader et al. 2010) (the proof is easily extended to cover the additional concepts for the lower and left border):

**Lemma 37.** For every $\Sigma$-ABox $A$ that is consistent w.r.t. $T_{\Psi}$, the following conditions are equivalent:

- $\models (T_{\Psi}, A)$;
- $A = A_0 \cup A_1$ for a $\Psi$-ABox $A_0$ and $a$, possibly empty, ABox $A_1$ with $\text{Ind}(A_0) \cap \text{Ind}(A_1) = \emptyset$.

Observe that the concept name $A$ used in the CQ occurs only once in the TBox, on the right-hand side of a CI. The CI for $C$ enforces confluence, i.e., $C$ is entailed at an individual $a$ if there is an individual $b$ that is both an $x$-$y$-successor and a $y$-$x$-successor of $a$. This is so because, intuitively, $B_i$ is universally quantified: if confluence fails, we can interpret $Z_{e,1}$ and $Z_{e,2}$ in a way such that neither of the two conjuncts in the pre-condition of the CI for $C$ is satisfied. In a similar manner, the CI for $I_e$ (resp. $I_y$) is used to ensure that $x^-$ (resp. $y^-$) acts as the inverse of $x$ (resp. $y$) at all points in the rectangle, which means that $x$ (resp. $y$) is inverse functional within the rectangle. The following characterization of tilings follows directly from Lemma 37.

**Lemma 38.** $\Psi$ admits a tiling iff there is a $\Sigma$-ABox $A$ that is consistent with $T_{\Psi}$ and such that $T_{\Psi}, A \models \exists v.A(v)$.

Set $\Sigma = \text{sig}(T_{\Psi}) \setminus \Sigma$. To construct the TBoxes we use for the reduction, replace within the TBoxes $T_{\Psi}$ all $B \in \Sigma$ by the concepts $H_B = \forall y. \exists y.B \land y.B$ and add $T_Z = \{ T \subseteq \exists y.B, T \subseteq \exists y.B \mid B \in \Sigma \}$ to $T_{\Psi}$. Also, add the inclusion $H_A \subseteq B_1 \cup B_2$, where $B_1, B_2$ are fresh concept names, to $T_{\Psi}$. Denote the resulting TBox by $T_{\Psi}'$.

For any ABox $A$, we denote by $A^{\Sigma'}$ the subset of $A$ consisting of all assertions in $A$ that use only symbols from $\Sigma$.

**Lemma 39.** For any ABox $A$, $T_{\Psi}', A \models \exists v.H_A(v)$ iff $T_{\Psi}', A^{\Sigma'} \notmodels \exists v.H_A(v)$.

**Proof.** The direction from right to left is trivial. Conversely, suppose $(T_{\Psi}', A^{\Sigma'}) \notmodels \exists v.H_A(v)$. Take a model $I$ of $(T_{\Psi}', A^{\Sigma'})$ such that $A^{\Sigma'} = \emptyset$. Since there are no existential restrictions on the right hand side of CIs, we can assume that $\Delta^{\Sigma'} = \{ a^{\Sigma'} \mid a \in \text{Ind}(A) \}$. Now set, for $B \in \Sigma$, $I_B = \{ a \in \text{Ind}(A) \mid a^{\Sigma} \in B^{\Sigma} \}$. Using Lemma 21, we can find a model $I$ of $(T_{\Psi}', A)$ refuting $\exists v.H_A(v)$.

**Lemma 40.** Assume $\Psi$ does not admit a tiling. Then $T_{\Psi}'$ is FO-rewritable for consistent ABoxes. Hence $T_{\Psi}'$ is materializable and CQ-answering w.r.t. $T_{\Psi}'$ is in PTIME.

**Proof.** If $\Psi$ does not admit a tiling, then $(T_{\Psi}, A^{\Sigma'}) \notmodels \exists v.H_A(v)$, for any ABox $A$ such that $A$ is consistent w.r.t. $T_{\Psi}$, by Lemma 38. Thus, $(T_{\Psi}', A) \notmodels \exists v.H_A(v)$ for any ABox $A$ such that $A$ is consistent w.r.t. $T_{\Psi}'$, by Lemma 39. But now one can show for any ABox $A$ that is consistent w.r.t. $T_{\Psi}'$ and any CQ $q$,

$$(T_{\Psi}', A) \models q \iff (T_{\Psi}, A) \models q$$

$T_{\Psi}$ is FO-rewritable. Thus, $T_{\Psi}'$ is FO-rewritable for consistent ABoxes.

**Lemma 41.** Assume $\Psi$ admits a tiling. Then $T_{\Psi}'$ is not materializable. Thus, $T_{\Psi}'$ is not FO-rewritable for consistent ABoxes and CQ-answering w.r.t. $T$ is CONP-hard.
Proof. Let $A$ be a $\Sigma$-ABox such that $(T_{\mathfrak{q}}, A) \models \exists v.A(v)$ and $A$ is consistent w.r.t. $T_{\mathfrak{q}}$. Then $(T_{\mathfrak{q}}^\\exists, A) \models \exists v.(B_1(v) \vee B_2(v))$ and $A$ is consistent w.r.t. $T_{\mathfrak{q}}^\\exists$. It is readily checked that $(T_{\mathfrak{q}}^\\exists, A) \not\models \exists v.B_1(v)$ and $(T_{\mathfrak{q}}^\\exists, A) \not\models \exists v.B_2(v)$. Thus, $T_{\mathfrak{q}}^\\exists$ is not materializable. \hfill \Box

From Lemmas 40 and 41, we obtain Points 3 and 4 of Theorem 36 as well as Points 1 and 2 for consistent ABoxes. Thus, to prove Theorem 36 it remains to show the following lemma.

Lemma 42. Consistency of ABoxes w.r.t. $T_{\mathfrak{q}}^\\exists$ can be decided in polynomial time (in the size of the ABox).

Proof. Assume $A$ is given. Form $A^\Sigma$ and apply the following rules exhaustively:

- add $I_x(a)$ to $A^\Sigma$ if there exists $b$ with $x(a, b), x^-(b, a) \in A$;
- add $I_y(a)$ to $A^\Sigma$ if there exists $b$ with $y(a, b), y^-(b, a) \in A$;
- add $C(a)$ to $A^\Sigma$ if there exist $a_1, a_2, b$ with $x(a, a_1), y(a_2, a), x(a_2, b), x(a, b) \in A$.

Denote the resulting ABox by $A^\Sigma$. Now remove the three inclusion schemata involving the Boolean combinations $B$ from $T_{\mathfrak{q}}^\\exists$ and denote by $T$ the resulting TBox. One can show that $(T_{\mathfrak{q}}^\\exists, A^\Sigma)$ is consistent iff $(T, A^\Sigma)$ is consistent. The consistency of the latter can be checked in polynomial time since $T$ is a Horn-$\textit{ALCF}$-TBox.

We now come to the proof of Theorem 27.

Theorem 27 For every language $L$ in coNP there exists a $\textit{ALCF}$-TBox $T$ and query $\text{rej}(a)$, $\text{rej}$ a concept name, such that the following holds:

- there exists a polynomial reduction of deciding $v \in L$ to answering $\text{rej}(a)$ w.r.t. $T$;
- for every Boolean ELIQ $q$, answering $q$ w.r.t. $T$ is polynomially reducible to deciding $v \in L$.

Consider a non-deterministic TM $M = (Q, \Sigma, \Delta, q_0, q_0, q_r)$ with $Q$ a finite set of states, $\Sigma$ a finite alphabet, $q_0 \in Q$ a starting state, $\Delta \subseteq Q \times \Sigma \times Q \times \Sigma \times \{L, R\}$ the a transition relation, and $q_a, q_r \in Q$ the accepting and rejecting states. We assume that for any input $v \in \Sigma^*$, $M$ halts after exactly $|v|^k$ steps in the accepting or rejecting state and that it uses exactly $n^k$ cells for the computation. Denote by $L(M)$ the language accepted by $M$ and assume that $L = \Sigma^* \setminus L(M)$.

The ABoxes we use to simulate input words $w \in \Sigma^*$ are $m_1 \times m_2$ grids in which $T_{\text{init}}$ is written in the lower left corner followed by the word $w$, $T_{\text{final}}$ is written in the upper right corner, and $B$ (for blank) is written everywhere else. In our construction of $T$ we first build a TBox that "checks" whether the input ABox is of this form.

To define this part of the TBox, we re-use the above TBox $T_{\mathfrak{q}}$, where $\mathfrak{q} = (\Sigma, H, V)$ with $\Sigma = \{B, T_{\text{final}}, T_{\text{init}}\} \cup \Sigma$ and $H$ consisting of all pairs in $\Sigma \times \Sigma$ except

- $(B, \sigma)$ for $\sigma \in \Sigma$,
- $(\sigma, T_{\text{final}})$ for $\sigma \in \Sigma$.

For any $n, m \geq 2$, and any word $v \in L^*$ there is exactly one tiling $f$ for $\mathfrak{q}$. That tiling places $T_{\text{init}}$ in the lower left corner followed by the word $v$, $T_{\text{final}}$ in the upper right corner, and $B$ is written everywhere else. Thus, every $\mathfrak{q}$-ABox $A$ (with initial node $a$) is isomorphic to some $n \times m$-grid with a word $T_{\text{init}}v (v \in L^*)$ written in the lower left corner. We call this ABox the $\textit{grid}$-ABox for the $n \times m$-rectangle with word $v$. Set

$$T_{\text{grid}} := T_{\mathfrak{q}}, \quad T_{\text{grid}\Sigma} := T_{\mathfrak{q}}^\exists \setminus \{H_A \subseteq B_1 \cup B_2\}.$$ 

Recall that $T_{\text{grid}\Sigma}$ contains the inclusions $T_Z$ for "second-order variables".

To encode the computation of the TM $M$ we use the following set $Z_M$ of inclusions. Intuitively, assume that a grid-ABox with initial node $a$ for the $n \times m$-rectangle with word $v$ is given. Then $(T_{\text{grid}\Sigma}, A) \models H_A(a)$. We introduce a concept name $H_{\text{grid}}$ denoting all individual names in $A$:

$$H_A \models H_{\text{grid}}, \quad H_{\text{grid}} \equiv \forall v.H_{\text{grid}}$$

for all $r \in \{x, y, x^-, y^-, \}$. The remaining inclusions are all relativized to $H_{\text{grid}}$. The remaining inclusions use

- concept names $q \in Q$ that indicate the state of the TM in the computation;
- concept names $\sigma \in \Sigma$ for the input word;
- concept names $A_\sigma, \sigma \in \Sigma$, for symbols written during the computation (and as copies of the symbols of the input word);
- a concept name $H$ for the head of the TM.

We simulate the instructions of $M$ by taking for $(q, \sigma, q') \in Q \times \Sigma \times Q$:

$$H_{\text{grid}} \cap H \cap q \cap A_{\sigma} \equiv \bigcup_{(q, \sigma, q', \sigma', r) \in \Delta} \exists y. (A_{\sigma'} \cap q' \cap \neg H \cap \forall x. \neg H \cap \exists x^{-}.H) \cup$$

$$\bigcup_{(q, \sigma, q', r) \in \Delta} \exists y. (A_{\sigma'} \cap q' \cap \neg H \cap \forall x^{-}.H \cap \exists x.H)$$

We state that cells can only change where $H$ is:

$$H_{\text{grid}} \cap \neg H \cap A_{\sigma} \subseteq \forall y. A_{\sigma}, \quad H_{\text{grid}} \cap \neg H \cap \neg A_{\sigma} \subseteq \forall y, \neg A_{\sigma}$$

We state that $H$ cannot be introduced without a corresponding computation step:

$$H_{\text{grid}} \cap \neg H \cap \forall x^{-}.H \cap \forall x.H \subseteq \forall y, \neg H$$

We state that, when $M$ starts, it is in state $q_0$ and that the head is at the first cell:

$$T_{\text{init}} \cap H_{\text{grid}} \subseteq q_0, \quad T_{\text{init}} \cap H_{\text{grid}} \equiv \exists x.H \cap \forall y^{-}.\bot \cap H_{\text{grid}}.$$
We state that every state \( q \) is uniform over each step of the computation:

\[ q \cap H_{\text{grid}} \subseteq \forall x. q \cap \forall x. q. \]

We state that \( A_\sigma \) is true where \( \sigma \) from the input word is true:

\[
H_{\text{grid}} \cap \sigma \equiv H_{\text{grid}} \cap \forall y. \bot \cap A_\sigma,
\]

for \( \sigma \in \Sigma \). We close with

\[
H_{\text{grid}} \cap A_\sigma \cap A_{\sigma'} \subseteq \bot,
\]

\[
H_{\text{grid}} \cap q \cap q' \subseteq \bot,
\]

for \( \sigma \neq \sigma' \) and \( q \neq q' \), and the assertion that \( \text{rej} \) is true everywhere in the ABox if the machine reaches the rejecting state:

\[
H_{\text{grid}} \cap q_r \subseteq \text{rej},
\]

\[
H_{\text{grid}} \cap \text{rej} \subseteq \forall r. \text{rej}
\]

for \( r \in \{x, y, x^-, y^-\} \). This finishes the definition of \( Z_M \).

As before, we replace every concept name

\[
B \in X := Q \cup \{A_\sigma \mid \sigma \in \Sigma\} \cup \{H_{\text{grid}}, H\}
\]

by \( H_B = \forall r_B. \exists s_B. -Z_B \) and add

\[
\mathcal{T}_{Z,1} = \{T \subseteq \exists r_B. \top, T \subseteq \exists s_B. Z_B \mid B \in X\}
\]

to \( Z_M \) and denote the resulting TBox by \( Z_{SO} \). We set \( T_{SO} = T_{grid} \cup Z_{SO} \). Note that the only “real” concept names in \( T_{SO} \) are \( \top \) and \( \text{rej} \). The following lemma is straightforward now and proves Part 1 of Theorem 27.

**Lemma 43.** If \( A \) is the grid-ABox for the \( m_1 \times m_2 \)-rectangle with word \( v \) and \( m_1, m_2 \geq n^k \) for \( n = |v| \), then \( (T_{SO}^A, A) \models \text{rej}(a) \iff v \notin L(M) \).

By Lemma 43, to check \( v \notin L(M) \), it sufficient to construct the grid-ABox for the \( n^k \times n^k \)-rectangle with word \( v \) and then decide \( (T_{SO}^A, A) \models \text{rej}(a) \). Thus, we have shown that there exists a polynomial reduction of deciding \( v \in L \) to answering \( \text{rej}(a) \) w.r.t. \( T_{SO}^A \).

We now show that for every ELIQ \( C(f) \), answering \( C(f) \) w.r.t. \( T_{SO}^A \) can be polynomially reduced to deciding \( v \in L \). Assume \( C(f) \) is given. Consider an ABox \( A \).

Claim 1. It can be checked in polytime (in the size of \( A \)) whether \( A \) is consistent w.r.t. \( T_{SO}^A \).

Observe that \( A \) is not consistent w.r.t. \( T_{SO}^A \) iff

- \( A \) contains a grid-ABox for a \( m_1 \times m_2 \)-rectangle with word \( v \) and \( m_1 < n^k \) or \( m_2 < n^k \) for \( n = |v| \); or
- \( A \) is not consistent w.r.t. \( T_{grid} \).

The first condition can clearly be checked in polytime and the latter is in PTIME by Lemma 42.

Now, if \( A \) is consistent w.r.t. \( T_{SO}^A \), then one of the following two cases applies:

- \( f \) is in a grid-ABox for the \( m_1 \times m_2 \)-rectangle with word \( v \) and \( m_1, m_2 \geq n^k \) for \( n = |v| \) (there can be other disjoint components). In that case \( (T_{SO}^A, A) \models C(f) \) iff \( (T_V, A') \models C(f) \), where
  - \( A' \) is defined by setting \( A' = A \cup \{\text{rej}(b) \mid b \in \text{Ind}(A)\} \)
    - if \( v \notin L(M) \); and \( A' := A \) otherwise.
  - \( T_V = T_Z \cup T_{Z,1} \).

Both conditions can be checked in polytime.

- \( f \) is not in a grid-ABox for the \( m_1 \times m_2 \)-rectangle with word \( v \). In that case \( (T_{SO}^A, A) \models C(f) \) iff \( (T_Z, A) \models C(f) \). The latter condition can be checked in polytime.