Available online at www.sciencedirect.com

SciVerse ScienceDirect

Electronic Notes in Theoretical Computer Science

Electronic Notes in Theoretical Computer Science 278 (2011) 71–84

www.elsevier.com/locate/entcs

Query Answering with DBoxes is Hard[1]

Enrico Franconi[2] Yazmín Angélica Ibáñez-García[2]

Free University of Bozen-Bolzano

İnanç Seylan[3]

Universität Bremen

Abstract

Data in description logic knowledge bases is stored in the form of an ABox. ABoxes are often confusing for developers coming from relational databases because an ABox, in contrast to a database instance, provides an incomplete specification. A recently introduced assertional component of a description logic knowledge base is a DBox, which behaves more like a database instance. In this paper, we study the data complexity of query answering in the description logic DL-Lite$_\mathcal{F}$ extended with DBoxes. DL-Lite$_\mathcal{F}$ is a description logic tailored for data intensive applications and the data complexity of query answering in DL-Lite$_\mathcal{F}$ with ABoxes is tractable (in AC^0). Our main result is that this problem becomes coNP-complete with DBoxes. In some expressive description logics, query answering with DBoxes also leads to a higher (combined) complexity than query answering with ABoxes. As a proof of concept, we relate query answering in $\mathcal{ALCFIO}$, i.e., $\mathcal{ALC}$ with $\mathcal{F}$unctional and $\mathcal{I}$nverse roles, and n$\mathcal{O}$minals to the same problem in $\mathcal{ALCFI}$ with DBoxes. The exact complexity of the former is an open problem in the description logic literature. Here we show that query answering in $\mathcal{ALCFIO}$ and $\mathcal{ALCFI}$ with DBoxes are mutually reducible to each other in polynomial time.
All the proofs in this paper are available in the appendix for the reviewers' convenience.

Keywords: Description logics, conjunctive queries, hybrid logic, model theory.

1 Introduction

Description Logics (DLs) constitute a family of logics commonly used in knowledge representation; and they are the logical underpinning of the OWL 2 Web Ontology Language [3]. A standard use case for DL-based systems is to store the facts about the application domain in a *knowledge base* (KB) and then *query* the KB. Traditionally, a DL KB consists of two components: a TBox and an ABox [3]. The TBox is the intensional part of the KB. For example, in a TBox one can assert that a

[1] This work was supported by DFG SFB/TR 8 "Spatial Cognition"

[2] Email: `[franconi|ibanezgarcia]@inf.unibz.it`

[3] Email: `seylan@informatik.uni-bremen.de`

doi:10.1016/j.entcs.2011.10.007

father is a man with at least one child. The ABox on the other hand is for asserting facts about individuals, e.g., John is a father, Mary is a daughter of John, etc. As for querying the KB, a popular query language is conjunctive queries (CQs) which originate from database theory [1].

The ABox of a DL KB resembles a database instance since it talks about individuals. In contrast to a database instance, an ABox provides an incomplete specification for the predicates appearing in it. Because of this incompleteness, when one talks about answers of a query over a DL KB, one uses the notion of *certain answers*, i.e., answers which hold in every model of the KB. To elaborate, consider the DL KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, where $\mathcal{T} = \{\texttt{Employee} \sqsubseteq \exists\texttt{worksFor}, \exists\texttt{worksFor}^{-} \sqsubseteq \texttt{Project}\}$ and
$\mathcal{A} = \{\texttt{Employee}(\texttt{john}), \texttt{Project}(\texttt{prja})\}$. In the TBox $\mathcal{T}$, we assert that every employee works for a project; in the ABox $\mathcal{A}$, we assert that `john` is an employee and `prja` is a project. By the semantics of an ABox, we have that there are some models of $\mathcal{K}$ for which somebody works for `prja` and some models for which nobody works for it. Therefore, the certain answers of the query $\texttt{worksFor}(x, \texttt{prja})$, which asks for all employees working for `prja`, is empty.

This semantical difference between ABoxes and database instances is often confusing for developers with experience in relational database management systems. In order to avoid such problems, *DBoxes* were introduced recently as an alternative assertional component of DL KBs [16]. Syntactically, a DBox looks very similar to an ABox. For example, the set of assertions above also constitute a DBox $\mathcal{D}$. The difference between $\mathcal{A}$ and $\mathcal{D}$ is in the semantics because a DBox is similar to a database instance in that the absence of information is interpreted as negative information. In particular, if we replace $\mathcal{A}$ by $\mathcal{D}$ in $\mathcal{K}$, then the certain answers of the query $\texttt{worksFor}(x, \texttt{prja})$ is `john`. DBoxes are closely related to *nominals* from hybrid logics [2]. In this respect, we think that they provide a nice connection between hybrid logics, DLs, and databases.

For these reasons, it is a natural research topic to study the complexity of query answering in DLs with DBoxes. In this paper, we choose to study this problem for a DL that is oriented towards data intensive applications since DBoxes are the data components of a KB. In particular, we study the data complexity of query answering in $\mathsf{DL\text{-}Lite}_{\mathcal{F}}$ extended with DBoxes. $\mathsf{DL\text{-}Lite}_{\mathcal{F}}$ belongs to the $\mathsf{DL\text{-}Lite}$ family of DLs [6]. The data complexity of query answering in these logics (with ABoxes) is tractable since these problems can be reduced to query answering in relational databases. Our main result is that query answering in $\mathsf{DL\text{-}Lite}_{\mathcal{F}}$ with DBoxes is CONP-complete.

$\mathsf{DL\text{-}Lite}_{\mathcal{F}}$ with DBoxes is closely related to the expressive DL $\mathcal{ALCFIO}$, i.e., $\mathcal{ALC}$ with $\mathcal{F}$unctional and $\mathcal{I}$nverse roles, and n$\mathcal{O}$minals. This is because $\mathsf{DL\text{-}Lite}_{\mathcal{F}}$ with DBoxes contains all these three constructs in a restricted way. Since the exact (combined) complexity of query answering in $\mathcal{ALCFIO}$ is an open problem in the DL literature, our result about $\mathsf{DL\text{-}Lite}_{\mathcal{F}}$ with DBoxes is also interesting in this sense. As another contribution, we identify an expressive DL, namely $\mathcal{ALCFI}$ extended with DBoxes, such that query answering in this logic is polynomially reducible to the same problem in $\mathcal{ALCFIO}$ and vice versa. As a consequence of these reductions, any complexity result about $\mathcal{ALCFI}$ with DBoxes is easily transferable to $\mathcal{ALCFIO}$. Another consequence of this result is that we identify an expressive DL, namely $\mathcal{ALCFI}$, for which query answering with DBoxes is strictly harder (coN2ExpTime-hard [7]) than query answering with ABoxes (2-ExpTime-complete [8]).

The paper is structured in the following way. In Section 2, we review some basic notions from description logics and conjunctive queries. In particular, we define the syntax and semantics of the logics we are interested in, as well as, how conjunctive queries are matched in models. Section 3 introduces the notion of DBoxes as an alternative way of representing extensional knowledge in description logics; together with the notion of query entailment with DBoxes. In Section 4, we establish the main results regarding the data complexity of query entailment in $\mathsf{DL\text{-}Lite}_{\mathcal{F}}$ with DBoxes. Finally, in Section 5 we show that query entailment in $\mathcal{ALCFIO}$ and query entailment in $\mathcal{ALCFI}$ with DBoxes are problems that are polynomially reducible to each other.

2 Preliminaries

2.1 $\mathcal{ALCFIO}$

The language of $\mathcal{ALCFIO}$ contains *concept names* $\mathsf{N}_C = \{A_0, A_1, \ldots, \}$, *role names* $\mathsf{N}_R = \{P_0, P_1, \ldots\}$, and *individual names* $\mathsf{N}_I = \{a_0, a_1, \ldots\}$, such that N_C, N_R and N_I are countably infinite and mutually disjoint sets. Complex *roles* R and *concepts* C of this language are defined as follows:

$$R \;::=\; P \;\mid\; P^-,$$
$$C \;::=\; \top \;\mid\; A \;\mid\; \{a\} \;\mid\; \neg C \;\mid\; C_1 \sqcap C_2 \;\mid\; \exists R.C \mid \leq 1R$$

$\mathcal{ALCFI}$*-concepts* are defined as above, except they exclude *nominals*, i.e., concepts of the form $\{a\}$.

An $\mathcal{ALCFIO}$*-TBox* $\mathcal{T}$ is a finite set of *concept inclusion* axioms (or simply concepts inclusions) of the form:

$$C_1 \sqsubseteq C_2.$$

An $\mathcal{ALCFIO}$*-ABox* is a finite set of assertions of the form:

$$C(a),\;\; P(a, b),$$

where C is a complex concept, $P \in \mathsf{N}_R$, $a, b \in \mathsf{N}_I$. For a role R we set $\mathsf{Inv}(R) := P^-$ if $R = P \in \mathsf{N}_R$, and $\mathsf{Inv}(R) := P$ if $R = P^-$, $P \in \mathsf{N}_R$.

As usual in description logics, the semantics of $\mathcal{ALCFIO}$ is given in terms of *interpretations*. An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, consists of an non empty *domain* $\Delta^{\mathcal{I}}$, and an interpretation function $\cdot^{\mathcal{I}}$ that assigns to each $A \in \mathsf{N}_C$ a subset $A^{\mathcal{I}}$ of

$\Delta^{\mathcal{I}}$, to each $P \in \mathsf{N}_R$ a binary relation $P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ over the domain, and to each $a \in \mathsf{N}_I$ an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. Unless otherwise stated, we do not make the *unique name assumption* (UNA) for individual names, i.e., for all $a, b \in \mathsf{N}_I$ and all interpretations $\mathcal{I}$, if $a \neq b$ then it is not necessarily the case that $a^{\mathcal{I}} \neq b^{\mathcal{I}}$.

Furthermore, $\cdot^{\mathcal{I}}$ is extended to complex $\mathcal{ALCFIO}$-concepts inductively as follows:

$$\begin{aligned}
\top^{\mathcal{I}} &= \Delta^{\mathcal{I}},\\
(P^-)^{\mathcal{I}} &= \{\langle t, s\rangle \mid \langle s, t\rangle \in P^{\mathcal{I}}\},\\
\{a\}^{\mathcal{I}} &= \{a^{\mathcal{I}}\},\\
(\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}},\\
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}},\\
(\exists R.C)^{\mathcal{I}} &= \{s \in \Delta^{\mathcal{I}} \mid \exists t \in \Delta^{\mathcal{I}}.\langle s, t\rangle \in R^{\mathcal{I}} \wedge t \in C^{\mathcal{I}}\},\\
(\leq 1R)^{\mathcal{I}} &= \{s \in \Delta^{\mathcal{I}} \mid \forall t, u \in \Delta^{\mathcal{I}}.\langle s, t\rangle \in R^{\mathcal{I}} \wedge \langle s, u\rangle \in R^{\mathcal{I}} \rightarrow t = u\}.
\end{aligned}$$

The *satisfaction relation* $\models$ is also standard. Given an interpretation $\mathcal{I}$, $\mathcal{I} \models C_1 \sqsubseteq C_2$ iff $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$; $\mathcal{I} \models C(a)$ iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$, and $\mathcal{I} \models P(a, b)$ iff $\langle a^{\mathcal{I}}, b^{\mathcal{I}}\rangle \in P^{\mathcal{I}}$.

A knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is said to be satisfiable (or consistent) if there is an interpretation $\mathcal{I}$ satisfying all members of $\mathcal{T}$ and $\mathcal{A}$. In this case we write $\mathcal{I} \models \mathcal{T}$ (as well as $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$) and say that $\mathcal{I}$ is a *model* of $\mathcal{K}$ (and of $\mathcal{T}$ and $\mathcal{A}$).

For an $\mathcal{ALCFIO}$-concept C, we denote as $\mathsf{sub}(C)$ the set of all its subconcepts, (i.e., subformulae). For an $\mathcal{ALCFIO}$-TBox $\mathcal{T}$, $\mathsf{con}(\mathcal{T})$ denotes the smallest set of concepts such that (i) if $C_1 \sqsubseteq C_2 \in \mathcal{T}$ then $C_1, C_2 \in \mathsf{con}(\mathcal{T})$; and (ii) if $C_1 \in \mathsf{con}(\mathcal{T})$ and $C_2 \in \mathsf{sub}(C_1)$ then $C_2 \in \mathsf{con}(\mathcal{T})$.

2.2 DL-Lite$_{\mathcal{F}}$

Basic DL-Lite$_{\mathcal{F}}$*-concepts* are defined as follows.

$$B ::= \bot \mid A \mid \exists R$$

A DL-Lite$_{\mathcal{F}}$ TBox, is a finite set of concept inclusion axioms of the form:

$$B_1 \sqsubseteq B_2, \qquad B_1 \sqsubseteq \neg B_2, \quad \text{or} \quad (\mathsf{funct}\ R)$$

As usual, the semantics of DL-Lite$_{\mathcal{F}}$-concepts is given in terms of interpretations. As DL-Lite$_{\mathcal{F}}$-concepts are special cases of $\mathcal{ALCFIO}$-concepts, e.g., $\bot$ corresponds to $\neg\top$, and $\exists R$ to $\exists R.\top$, we omit the explicit definition of the semantics here.

The satisfaction relation is defined in the same way as for $\mathcal{ALCFIO}$. We only need to extend it for the (global) *functionality* axioms, i.e., given an interpretation $\mathcal{I}$, $\mathcal{I} \models (\mathsf{funct}\ R)$ if for every $s, t, u \in \Delta^{\mathcal{I}}$, whenever $\langle s, t\rangle \in R^{\mathcal{I}}$ and $\langle s, u\rangle \in R^{\mathcal{I}}$, then $t = u$.

For a DL-Lite$_{\mathcal{F}}$-TBox, $\mathsf{con}(\mathcal{T})$ is the smallest set of concepts such that (i) if $B_1 \sqsubseteq B_2 \in \mathcal{T}$ then $B_1, B_2 \in \mathsf{con}(\mathcal{T})$; and (ii) if $B_1 \sqsubseteq \neg B_2 \in \mathcal{T}$ then $B_1, B_2 \in \mathsf{con}(\mathcal{T})$, i.e., the set of basic concepts occurring in $\mathcal{T}$.

2.3 Conjunctive Queries

Conjunctive queries (CQs) are the most frequently asked queries in relational database systems [1]. These queries are definable by existential positive first-order formulas and are preserved under homomorphisms. CQs are also common in DLs. In this section, we define our notation for CQs. In particular, we view concept and role names as unary and binary predicates, respectively. Since DLs are fragments of first-order logic with at most two variables, it does not make much sense to consider predicates of arity more than two.

Let N_V be a countably infinite set of *variables* which is disjoint from N_I, i.e., the set of individual names. Together, N_V and N_I form the set N_T of *terms*. A *conjunctive query* (CQ) is a first-order formula of the form $\exists \overline{y}.\varphi(\overline{x},\overline{y})$, where

- $\overline{x} = x_1,\ldots,x_n$ and $\overline{y} = y_1,\ldots,y_m$ are vectors of variables and
- φ is a conjunction of *concept atoms* $A(t)$ and *role atoms* $P(t,t')$, where $A \in \mathsf{N}_C$, $P \in \mathsf{N}_R$, and $t,t' \in \mathsf{N}_T$.

The variables in $\overline{x}$ are called *distinguished variables*; and the ones in $\overline{y}$ are *undistinguished*. We call the query k-ary if there are k distinguished variables. For a CQ q, we denote by $\mathsf{terms}(q)$ the set of terms in q.

Let $q = \exists \overline{y}.\varphi(\overline{x},\overline{y})$ be a k-ary CQ and $\mathcal{I}$ an interpretation. A *match* for q in $\mathcal{I}$ is a mapping $\nu : \mathsf{terms}(q) \to \Delta^{\mathcal{I}}$ such that $\nu(a) = a^{\mathcal{I}}$ for all $a \in \mathsf{terms}(q) \cap \mathsf{N}_I$ and all atoms in q are satisfied, i.e.,

- $\nu(t) \in A^{\mathcal{I}}$ for all $A(t) \in q$ and
- $\langle \nu(t), \nu(t')\rangle \in P^{\mathcal{I}}$ for all $P(t,t') \in q$.

If ν is a match for q in $\mathcal{I}$ then we write $\mathcal{I},\nu \models q$. If there is a match for q in $\mathcal{I}$ then we denote this by $\mathcal{I} \models q$.

For a k-tuple of individual names $\overline{a} = a_1,\ldots,a_k$, a match ν for q in $\mathcal{I}$ is called an $\overline{a}$-*match* if $\nu(x_i) = a_i^{\mathcal{I}}$, $i \leq k$. We say that $\overline{a}$ is an *answer* to q in an interpretation $\mathcal{I}$ if there is an $\overline{a}$-match for q in $\mathcal{I}$ and use $\mathsf{ans}(q,\mathcal{I})$ to denote the set of all answers to q in $\mathcal{I}$.

3 DBoxes

In this section, we introduce the notion of knowledge bases with DBoxes. The syntax and semantics of DBoxes is given by the following definition:

Definition 3.1 A *DBox* is a finite set of assertions of the form $A(a)$ and $P(a,b)$, where $A \in \mathsf{N}_C$, $P \in \mathsf{N}_R$, and $a,b \in \mathsf{N}_I$. The set of individual names occurring in a DBox $\mathcal{D}$ is called the *active domain* of $\mathcal{D}$ and it is denoted by $\mathsf{adom}(\mathcal{D})$. The *signature* of a DBox $\mathcal{D}$, denoted as $\mathsf{sig}(\mathcal{D})$, consists of the concepts and role names occurring in $\mathcal{D}$, denoted as $\mathsf{con}(\mathcal{D})$ and $\mathsf{rol}(\mathcal{D})$, respectively.

Let $\mathcal{D}$ be a DBox and $\mathcal{I}$ an interpretation. $\mathcal{I} \models \mathcal{D}$ iff

- $a^{\mathcal{I}} \neq b^{\mathcal{I}}$, for all $a,b \in \mathsf{adom}(\mathcal{D})$ with $a \neq b$;
- $A^{\mathcal{I}} = \{a^{\mathcal{I}} \mid A(a) \in \mathcal{D}\}$, for every $A \in \mathsf{con}(\mathcal{D})$;

- $P^{\mathcal{I}} = \{\langle a^{\mathcal{I}}, b^{\mathcal{I}}\rangle \mid P(a,b) \in \mathcal{D}\}$, for every $P \in \mathsf{rol}(\mathcal{D})$.

Intuitively, the semantics of a DBox $\mathcal{D}$, enforces the UNA for the individual names in $\mathsf{adom}(\mathcal{D})$, and that the extensions of the concepts and roles occurring in $\mathcal{D}$, i.e., $\mathsf{sig}(\mathcal{D})$, are given by the assertions in $\mathcal{D}$, and coincide in every model $\mathcal{I}$ of $\mathcal{D}$.

Let $\mathscr{L}$ be either $\mathsf{DL\text{-}Lite}_{\mathcal{F}}$, $\mathcal{ALCFI}$ or $\mathcal{ALCFIO}$. A $\mathscr{L}$ knowledge base with a DBox ($\mathscr{L}$-KB) $\mathcal{K}$ is a pair $(\mathcal{T}, \mathcal{D})$, where $\mathcal{T}$ is an $\mathscr{L}$-TBox and $\mathcal{D}$ is a DBox. For a $\mathscr{L}$-KB $\mathcal{K} = (\mathcal{T}, \mathcal{D})$, we define the following notions:

- $\mathsf{con}(\mathcal{K}) = \mathsf{con}(\mathcal{T}) \cup \{A \in \mathsf{N}_C \mid A(a) \in \mathcal{D}\} \cup \{\exists R.\top, \exists R^-.\top \mid R(a,b) \in \mathcal{D}\}$;
- $\mathsf{rol}(\mathcal{K})$ the set of roles occurring in $\mathcal{T}$ or $\mathcal{D}$;
- $\mathsf{sig}(\mathcal{K})$ the set of concept names and role names occurring in $\mathcal{T}$ or $\mathcal{D}$;
- $\mathsf{adom}(\mathcal{K})$ the union of $\mathsf{adom}(\mathcal{D})$ and all individuals that appear as nominals, if any, in $\mathcal{T}$.

Let $\mathcal{I}$ be an interpretation and $\mathcal{K} = (\mathcal{T}, \mathcal{D})$. We have that $\mathcal{I} \models \mathcal{K}$ iff $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{D}$. We say that $\mathcal{K}$ *is satisfiable* if there is some interpretation such that $\mathcal{I} \models \mathcal{K}$.

A *certain answer* of a k-ary CQ q with respect to the $\mathscr{L}$-KB $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ is a k-tuple $\overline{a} \in \mathsf{adom}(\mathcal{D})^k$ such that $\overline{a} \in \mathsf{ans}(q, \mathcal{I})$ for all models $\mathcal{I}$ of $\mathcal{K}$. The set of certain answers to q over $\mathcal{K}$ will be denoted by $\mathsf{cert}(q, \mathcal{K})$. Moreover, $\mathcal{K} \models q$ iff for all interpretations $\mathcal{I}$, $\mathcal{I} \models \mathcal{K}$ implies $\mathcal{I} \models q$.

The *CQ answering problem* can be formulated as follows: given an $\mathscr{L}$-KB $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ and a CQ q, to compute $\mathsf{cert}(q, \mathcal{K})$. The *CQ entailment* problem is given an $\mathscr{L}$-KB $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ and a boolean CQ q, i.e., a CQ without any distinguished variables, to decide whether $\mathcal{K} \models q$.

Observe that for a k-ary CQ $q = \varphi(\overline{x})$, we have $\mathsf{cert}(q, \mathcal{K}) = \{\overline{a} \in \mathsf{adom}(\mathcal{D})^k \mid \mathcal{K} \models \varphi(\overline{a})\}$, where $\varphi(\overline{a})$ denotes the substitution of $\overline{x}$ by $\overline{a}$ in φ and is a boolean CQ. Since CQ answering can be reduced to CQ entailment in this way and that CQ entailment is a decision problem, in the rest of this paper we will study CQ entailment.

4 Data Complexity of CQ Entailment in $\mathsf{DL\text{-}Lite}_{\mathcal{F}}$

In this section, we study the data complexity of query entailment in $\mathsf{DL\text{-}Lite}_{\mathcal{F}}$ with DBoxes. Data complexity is a common measure of complexity in databases [1]. When considering data complexity, the only input considered is the database instance, while the query is assumed to be fixed.

Data complexity of query answering in DLs (w.r.t. KBs with ABoxes) is well-studied [5,12]. In this setting, the only input is the ABox, while the TBox and the query are regarded as fixed. As stated in the previously, we are interested in studying the data complexity of query answering in $\mathsf{DL\text{-}Lite}_{\mathcal{F}}$ with DBoxes. In this setting, we consider the DBox as the input, and again the TBox and the query are regarded as fixed.

Our main result in this section is Theorem 4.16. We show that query entailment in $\mathsf{DL\text{-}Lite}_{\mathcal{F}}$ is harder when we consider KBs with DBoxes. It is known that query entailment in DL-Lite$_{\mathcal{F}}$ with ABoxes is in AC^0, for data complexity [6]. However, as we show in Lemma 4.1, the problem becomes coNP-hard for data complexity, when DBoxes are considered. Moreover, we show that this complexity bound is tight (Theorem 4.16). In particular, we show a match with the data complexity of CQ entailment in expressive DLs such as $\mathcal{SHIQ}$, $\mathcal{SHOI}$, and $\mathcal{SHOQ}$ [12].

Lemma 4.1 *CQ entailment in* DL-Lite$_{\mathcal{F}}$ *with DBoxes is* coNP*-hard for data complexity.*

Proof. The proof is by a reduction of the 3-*colorability* problem for undirected graphs to non-entailment of a CQ w.r.t. a DL-Lite$_{\mathcal{F}}$-KB with DBox. An undirected graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ with node set $\mathcal{V}$ and edge set $\mathcal{E}$, is said to be 3-*colorable* if each node in $\mathcal{V}$ can be assigned exactly one of three colors, in such a way that no two adjacent nodes are assigned the same color. 3-colorability is the problem of deciding whether a given graph is 3-colorable. It is well-known that 3-colorability is NP-complete [13].

Given an (finite) undirected graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, let $\mathcal{K}_{\mathcal{G}} = (\mathcal{T}, \mathcal{D}_{\mathcal{G}})$, where

$$\mathcal{D}_{\mathcal{G}} = \{V(a_v) \mid v \in \mathcal{V}\} \cup \{E(a_v, a_{v'}) \mid \langle v, v' \rangle \in \mathcal{E}\} \cup \{C(r), C(g), C(b)\}$$

and $\mathcal{T} = \{V \sqsubseteq \exists R, \exists R^- \sqsubseteq C\}$, and let $q = \exists x, y, z[E(x, y) \wedge R(x, z) \wedge R(y, z)]$. In the definition above, we have for all $v, v' \in \mathcal{V}$, $v \neq v'$ implies $a_v \neq a_{v'}$, and the semantics of the DBox allows us to fix the extension of C in every model of $\mathcal{K}_{\mathcal{G}}$, thus expressing the fact that there are exactly three colors used for coloring. The role name R basically corresponds to the `hasColor` relation, and the meaning of V and E are self-explanatory. Observe that $\mathcal{T}$ and q does not depend on $\mathcal{G}$, which is essential for the correctness of the reduction for data complexity. It can now be shown that $\mathcal{G}$ is 3-colorable iff $\mathcal{K}_{\mathcal{G}} \not\models q$. □

For the upper bound, one can trivially embed DL-Lite$_{\mathcal{F}}$ with DBoxes to $\mathcal{ALCFIO}$. However, the data complexity of query answering in $\mathcal{ALCFIO}$ is unknown. Also note that DL-Lite$_{\mathcal{F}}$ with DBoxes can not be trivially embedded in $\mathcal{SHIQ}$, $\mathcal{SHOI}$, and $\mathcal{SHOQ}$ since each of these logics lack one of the constructs of DL-Lite$_{\mathcal{F}}$ with DBoxes. For these reasons, we will establish a weak *forest model property* for our logic in order to show the upper bound. These models consist of several trees and the roots of these trees may be arbitrarily connected to each other. Moreover, there may be back edges from non-root nodes to root nodes. We observe that it is not enough to take only active domains elements as the roots of these trees. This is because the interaction between DBox assertions, inverse roles, and functionality assertions may enforce the existence of elements in the domain that act like active domain elements although they are not. These elements are called *new nominals* [14]. In order to devise a decision procedure, we have to establish a bound on the number of new nominals, and hence the number of trees in our models. This is the same problem that one faces for CQ entailment in $\mathcal{ALCFIO}$. Note that in high contrast to $\mathcal{ALCFIO}$, the 'light' nature of DL-Lite$_{\mathcal{F}}$ allows us to establish a polynomial upper bound on the size of new nominals.

The key observation for establishing a bound on the number of new nominals is that they belong to concepts whose instances are bounded in every model of the KB. Clearly, DBox concepts have a bounded extension in every model. However, TBox axioms may enforce some other concepts, not occurring in the DBox, to have a bounded extension as well. We formalize this in the following definition.

Definition 4.2 Let $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ be a $\mathsf{DL\text{-}Lite}_{\mathcal{F}}$-KB with a DBox. Then $\mathsf{Bcon}(\mathcal{K})$ is inductively defined as follows:

- $\mathsf{Bcon}(\mathcal{K})^0 = \{B \in \mathsf{con}(\mathcal{K}) \mid B \in \mathsf{con}(\mathcal{D})\} \cup \{\exists P, \exists P^- \in \mathsf{con}(\mathcal{K}) \mid P \in \mathsf{rol}(\mathcal{D})\}$
- $\mathsf{Bcon}(\mathcal{K})^{i+1} = \{B_1 \in \mathsf{con}(\mathcal{K}) \mid B_1 \sqsubseteq B_2 \in \mathcal{T}$ and $B_2 \in \mathsf{Bcon}(\mathcal{K})^i\} \cup \{\exists \mathsf{Inv}(R) \in \mathsf{con}(\mathcal{T}) \mid (\mathsf{funct}\ R) \in \mathcal{T}$ and $\exists R \in \mathsf{Bcon}(\mathcal{K})^i\}$

As there are only finitely many concepts in $\mathsf{con}(\mathcal{K})$, there exists j, such that $\mathsf{Bcon}(\mathcal{K})^j = \mathsf{Bcon}(\mathcal{K})^{j+1}$. We set $\mathsf{Bcon}(\mathcal{K}) := \mathsf{Bcon}(\mathcal{K})^j$.

Lemma 4.3 *Let $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ be a $\mathsf{DL\text{-}Lite}_{\mathcal{F}}$-KB with a DBox. We have that*

- $\sharp\mathsf{Bcon}(\mathcal{K}) \leq |\mathcal{K}|$, *and*
- *for every model $\mathcal{I}$ of $\mathcal{K}$ and for all $B \in \mathsf{Bcon}(\mathcal{K})$, $\sharp(B^{\mathcal{I}}) \leq \sharp\mathsf{adom}(\mathcal{D})$.*

In order to establish a forest model property for $\mathsf{DL\text{-}Lite}_{\mathcal{F}}$ with DBoxes, we will work on structures based on graphs instead of interpretations. We call these structures $\mathcal{K}$-*graphs* because of their intimate connection with a given KB $\mathcal{K}$. Such structures are commonly used in the literature and have many names, e.g., Hintikka structure [15], model graph [9], or even tableau [10]. Modulo some differences, building blocks of these structures are sets of finite concepts each of which is a subset of a relevant concept closure.

Definition 4.4 Let $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ be KB. A $\mathcal{K}$-*graph* is a tuple $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$, where $(\mathcal{V}, \mathcal{E})$ is a directed graph with $\mathsf{adom}(\mathcal{D}) \subseteq \mathcal{V}$, and $\mathcal{L}$ is a function associating with every $v \in \mathcal{V}$ a subset of $\mathsf{con}(\mathcal{K})$ and with every $\langle v, v' \rangle \in \mathcal{E}$ a subset of $\mathsf{rol}(\mathcal{K})$. The set of *nominals* of $\mathcal{M}$ is the set of nodes $\mathsf{nom}(\mathcal{M}) = \{v \in \mathcal{V} \mid \mathcal{L}(v) \cap \mathsf{Bcon}(\mathcal{K}) \neq \emptyset\}$. We use the notation $R^{\mathcal{M}}(v, v')$ to express that

- $\langle v, v' \rangle \in \mathcal{E}$ and $R \in \mathcal{L}(v, v')$, or
- $\langle v', v \rangle \in \mathcal{E}$ and $\mathsf{Inv}(R) \in \mathcal{L}(v', v)$.

We are interested in certain $\mathcal{K}$-graphs that satisfy additional properties.

Definition 4.5 Let $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ be a $\mathsf{DL\text{-}Lite}_{\mathcal{F}}$-KB with a DBox and let $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$ be a $\mathcal{K}$-graph. We say $\mathcal{M}$ is a $\mathcal{K}$-*graph quasimodel* if for all $v, v' \in \mathcal{V}$, $\mathcal{M}$ satisfies the following conditions:

($\mathsf{P}^A_{\mathcal{D}}$) for all $A \in \mathsf{con}(\mathcal{D})$, $A \in \mathcal{L}(v)$ iff $v = a$ and $A(a) \in \mathcal{D}$, for some $a \in \mathsf{adom}(\mathcal{D})$;

($\mathsf{P}^R_{\mathcal{D}}$) for all $R \in \mathsf{rol}(\mathcal{D})$, $R^{\mathcal{M}}(v, v')$ iff $v = a$, $v' = b$, $R(a, b) \in \mathcal{D}$, for some $a, b \in \mathsf{adom}(\mathcal{D})$;

($\mathsf{P}^+_{\sqsubseteq}$) for all $B_1 \sqsubseteq B_2 \in \mathcal{T}$, if $B_1 \in \mathcal{L}(v)$ then $B_2 \in \mathcal{L}(v)$;

($\mathsf{P}^-_{\sqsubseteq}$) for all $B_1 \sqsubseteq \neg B_2 \in \mathcal{T}$, if $B_1 \in \mathcal{L}(v)$ then $B_2 \notin \mathcal{L}(v)$;

($\mathsf{P}^{\Leftarrow}_{\exists}$) for all $\exists R \in \mathsf{con}(\mathcal{K})$, if there is some $v' \in \mathcal{V}$ with $R^{\mathcal{M}}(v, v')$ then $\exists R \in \mathcal{L}(v)$;

($\mathsf{P}_{\leq}$) for all $(\mathsf{funct}\ R) \in \mathcal{T}$, there is at most one $v' \in \mathcal{V}$ with $R^{\mathcal{M}}(v, v')$.

$\mathcal{M}$ is a called a *model* if in addition to the properties above, it satisfies the following property.

($\mathsf{P}_{\exists}^{\Rightarrow}$) for all $\exists R \in \mathsf{con}(\mathcal{K})$, if $\exists R \in \mathcal{L}(v)$ then there is some $v' \in \mathcal{V}$ with $R^{\mathcal{M}}(v, v')$.

We write $\mathcal{M} \vdash \mathcal{K}$ to denote that $\mathcal{M}$ is a $\mathcal{K}$-graph model.

Query matches in $\mathcal{K}$-graphs are very similar to query matches in interpretations.

Definition 4.6 A *match* ν for a CQ q in a $\mathcal{K}$-graph $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$ is a total function $\nu : \mathsf{terms}(q) \to \mathcal{V}$ such that $\nu(a) = a$ for each individual name $a \in \mathsf{terms}(q)$. We write $\mathcal{M}, \nu \vdash q$ if for every $A(t) \in q$, $A \in \mathcal{L}(\nu(t))$; and for every $R(t, t') \in q$, $R^{\mathcal{M}}(\nu(t), \nu(t'))$.

The following lemma is a consequence of the definitions above, and establishes that $\mathcal{K}$-graph models capture faithfully the semantics of $\mathsf{DL\text{-}Lite}_{\mathcal{F}}$-KBs with DBoxes.

Lemma 4.7 *Let $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ be a $\mathsf{DL\text{-}Lite}_{\mathcal{F}}$-KB with a DBox and q be a CQ. Then $\mathcal{K} \not\models q$ if and only if there is some $\mathcal{K}$-graph model $\mathcal{M}$ such that $\mathcal{M} \not\vdash q$.*

Intuitively, a $\mathcal{K}$-graph $\mathcal{M}$ is called a $\mathcal{K}$-forest if the structure resulting from removing all edges going to nominals of $\mathcal{M}$ is a forest, i.e., a set of disjoint trees.

Definition 4.8 Let $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$ be a $\mathcal{K}$-graph. The graph $\mathcal{G}_f = (\mathcal{V}, \mathcal{E}_f)$, where

$$\mathcal{E}_f = \mathcal{E} \setminus \{\langle v, v' \rangle \in \mathcal{E} \mid v' \in \mathsf{nom}(\mathcal{M})\}$$

is called the *f-pruning* of $\mathcal{M}$. We call $\mathcal{M}$ a *$\mathcal{K}$-forest* if its f-pruning is a forest. The *roots* of a $\mathcal{K}$-forest $\mathcal{M}$ is the roots of its f-pruning and it is denoted by $\mathsf{roots}(\mathcal{M})$. The *branching degree* of a $\mathcal{K}$-forest $\mathcal{M}$ is the branching degree of its f-pruning and it is denoted by $\mathsf{bdegree}(\mathcal{M})$.

The $\mathcal{K}$-forests we are interested in have as their roots exactly the nominals and their tree parts are uniform.

Definition 4.9 Let $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$ be a $\mathcal{K}$-forest. Then $\mathcal{M}$ is called *uniform* if

(U1) for all $\langle v, v' \rangle \in \mathcal{E}$, if $v' \in \mathcal{V} \setminus \mathsf{nom}(\mathcal{M})$ then for some $\exists R \in \mathcal{L}(v)$, $\mathcal{L}(v, v') = \{R\}$ and $\mathcal{L}(v') = \{B \in \mathsf{con}(\mathcal{T}) \mid \mathcal{T} \models \exists \mathsf{Inv}(R) \sqsubseteq B\}$;

(U2) $\mathsf{bdegree}(\mathcal{M}) \leq |\mathcal{T}|$ and $\mathsf{roots}(\mathcal{M}) = \mathsf{nom}(\mathcal{M})$;

(U3) for all $\langle v, v' \rangle \in \mathcal{E}$, if $v \in \mathcal{V} \setminus \mathsf{nom}(\mathcal{M})$, $v' \in \mathsf{nom}(\mathcal{M})$, and $R \in \mathcal{L}(v, v')$ then $(\mathsf{funct}\ \mathsf{Inv}(R)) \notin \mathcal{T}$.

It is enough to consider only uniform forest-models of a KB (with polynomially many roots) for deciding CQ entailment. We establish this in the following.

Theorem 4.10 *Let $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ be a $\mathsf{DL\text{-}Lite}_{\mathcal{F}}$-KB and let q be a CQ. Then $\mathcal{K} \not\models q$ if and only if there exists a uniform $\mathcal{K}$-forest model $\mathcal{M}$ with $\mathcal{M} \not\vdash q$.*

Thus, we can decide CQ non-entailment by finding a $\mathcal{K}$-forest model $\mathcal{M}$ with $\mathcal{M} \not\models q$. At this point, we are faced with the problem that we can not simply construct a $\mathcal{K}$-forest model $\mathcal{M}$ and check whether $\mathcal{M} \not\models q$ since $\mathcal{M}$ can be infinite. However, as we will show, it is possible to represent $\mathcal{K}$-forest models in a finite way. Here the crucial observation is that in a sufficiently big tree in a uniform $\mathcal{K}$-forest model, there is a bounded number of subtrees up to isomorphism.

Definition 4.11 Let $\mathcal{M}_1 = (\mathcal{V}_1, \mathcal{E}_1, \mathcal{L}_1)$ and $\mathcal{M}_2 = (\mathcal{V}_2, \mathcal{E}_2, \mathcal{L}_2)$ be two $\mathcal{K}$-graphs. Then $\mathcal{M}_1$ and $\mathcal{M}_2$ are called *isomorphic*, written $\mathcal{M}_1 \cong \mathcal{M}_2$, if and only if there is a bijection $\beta : \mathcal{V}_1 \to \mathcal{V}_2$ such that:

- for all $a \in \mathsf{adom}(\mathcal{D})$ and $v \in \mathcal{V}_1$, $v = a$ iff $\beta(v) = a$;
- for all $v \in \mathcal{V}_1$, $\mathcal{L}_1(v) = \mathcal{L}_2(\beta(v))$;
- for all $v, v' \in \mathcal{V}_1$, $\langle v, v' \rangle \in \mathcal{E}_1$ iff $\langle \beta(v), \beta(v') \rangle \in \mathcal{E}_2$;
- for all $\langle v, v' \rangle \in \mathcal{E}_1$, $\mathcal{L}_1(v, v') = \mathcal{L}_2(\beta(v), \beta(v'))$;

If we want to specify the bijection explicitly, we use the notation $\mathcal{M}_1 \cong_\beta \mathcal{M}_2$.

Definition 4.12 Let $n \in \mathbb{N}$ be a fixed natural number and $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$ be a $\mathcal{K}$-forest. An *n-tree* in $\mathcal{M}$ is the restriction of $\mathcal{M}$ to $\{v\} \cup \mathsf{desc}^n_{\mathcal{G}_f}(v) \cup \mathsf{nom}(\mathcal{M})$, for some $v \in \mathcal{V} \in \backslash\mathsf{nom}(\mathcal{M})$. Here $\mathsf{desc}^n_{\mathcal{G}_f}(v)$ denotes all nodes v' of the subtree of $\mathcal{G}_f$ rooted at v such that the distance between v and v' is at most n.

Now we define the notion of *blocking* which is a standard technique for devising tableau-based decision procedures in DLs. Our definitions are based on the ones in [12] with slight variations in the notation.

Definition 4.13 Let $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$ be a $\mathcal{K}$-forest. We say that an n-tree $\mathcal{M}'$ in $\mathcal{M}$ *has a witness* if there is an n-tree $\mathcal{M}''$ in $\mathcal{M}$ such that

- $\mathcal{M}' \cong \mathcal{M}''$,
- $\mathcal{V}' \cap \mathcal{V}'' = \emptyset$,
- $v' = \mathsf{desc}_{\mathcal{G}_f}(v)$, where v' and v are roots of $\mathcal{M}'$ and $\mathcal{M}''$, respectively and $\mathsf{desc}_{\mathcal{G}_f}(v)$ denotes the set of all descendants of v on the subtree of $\mathcal{G}_f$ rooted at v.

In this case, $\mathcal{M}''$ is called a witness of $\mathcal{M}'$.

A $v \in \mathcal{V}$ is *directly n-blocked* if $v \notin \mathsf{nom}(\mathcal{M})$ and there is an n-tree $\mathcal{M}'$ in $\mathcal{M}$ such that v is a leaf of $\mathcal{M}'$ and $\mathcal{M}'$ has a unique witness. A $v \in \mathcal{V}$ is *indirectly n-blocked* if $v \in \mathsf{desc}_{\mathcal{G}_f}(v')$ for some directly n-blocked node v'. Finally $v \in \mathcal{V}$ is n-blocked if it is directly or indirectly n-blocked.

The class of structures we define next is the finite representation of $\mathcal{K}$-forest models that we are looking for.

Definition 4.14 A uniform $\mathcal{K}$-forest $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$ is called an $\mathcal{K}^n$*-forest model* if

- $\mathcal{V}$ contains no indirectly n-blocked node;
- $\mathcal{M}$ is a quasimodel, which satisfies the following property:
  ($\mathsf{P}^n_\exists$) for all $\exists R \in \mathsf{con}(\mathcal{K})$ and $v \in \mathcal{V}$ that is not n-blocked, if $\exists R \in \mathcal{L}(v)$ then there is some $v' \in \mathcal{V}$ with $R^{\mathcal{M}}(v, v')$.

Theorem 4.15 *Let $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ be a* $\mathsf{DL\text{-}Lite}_{\mathcal{F}}$*-KB with a DBoxes and let q be a CQ. Then $\mathcal{K} \not\models q$ if and only if there is some $\mathcal{K}^{|q|}$-forest model $\mathcal{M}$ with $\mathcal{M} \not\vdash q$.*

Proof. [Sketch] Let $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ be a $\mathsf{DL\text{-}Lite}_{\mathcal{F}}$-KB and let q be a CQ with $|q| = n$. We assume that q is connected. A query q is *connected* if, for all $t, t' \in \mathsf{terms}(q)$, there exists a sequence $t_1, \ldots, t_m$ such that $t_1 = t$, $t_m = t'$, and for all $i \in \{1, \ldots, m-1\}$, there exists a role name R such that $R(t_i, t_{i+1})$ or $R(t_{i+1}, t_i)$ is a conjunct of q. This assumption is w.l.o.g. since entailment of q can be decided by checking the entailment of each connected component of q (viewing q as an undirected graph) separately [14].
($\Rightarrow$) This is the easy direction of the proof. Suppose $\mathcal{K} \not\models q$. Then by Theorem 4.10, there is some uniform $\mathcal{K}$-forest model $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$ with $\mathcal{M} \not\vdash q$. We use an inductive construction. As the base case, we set $\mathcal{M}_0 = \mathcal{M}$. Now for the step from $\mathcal{M}_i$ to $\mathcal{M}_{i+1}$, first we choose a node $v \in \mathcal{V}_i$ that is indirectly n-blocked. Then we define $\mathcal{M}_{i+1}$ as the restriction of $\mathcal{M}_i$ to nodes $\mathcal{V}_i \setminus (\{v\} \cup \mathsf{desc}_{(\mathcal{G}_i)_f}(v))$, i.e., we 'chop off' the tree rooted at v. We have that $|\mathcal{M}_{i+1}| < |\mathcal{M}_i|$ and $\mathcal{M}_{i+1}$ lacks the tree rooted at v. Let $\mathcal{M}' = (\mathcal{V}', \mathcal{E}', \mathcal{L}')$ be the $\mathcal{K}$-forest obtained at the limit of this construction. It is easy to see that for every $v \in \mathcal{V}'$, v is either not n-blocked or v is directly n-blocked. Because of this and the fact that $\mathcal{M}$ is a uniform $\mathcal{K}$-forest model, we have that $\mathcal{M}'$ is a uniform $\mathcal{K}^n$-forest model. Moreover, we have $\mathcal{M}' \not\vdash q$ since $\mathcal{M}_0 \not\vdash q$ and for each step i, $\mathcal{M}_{i+1}$ is a strict substructure of $\mathcal{M}_i$.
($\Leftarrow$) Suppose $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$ is a uniform $\mathcal{K}^n$-forest model with $\mathcal{M} \not\vdash q$. The proof is analogous to the one given in [14] (Lemma 42). First we *unravel* $\mathcal{M}$ into a $\mathcal{K}$-forest. Unravelling is a standard construction in modal logics [4]; but in the presence of DBoxes (nominals), inverse roles, and functionality, one has to be more careful. This is because the standard construction can easily lead to the violation of functionality assertions. However, the uniformity of $\mathcal{M}$, more precisely (U3), ensures that this does not happen.

Let $\mathcal{M}'$ be the unravelling of $\mathcal{M}$. It follows by the properties of $\mathcal{M}$ that $\mathcal{M}'$ is a uniform $\mathcal{K}$-forest model. One has to show that $\mathcal{M}' \not\vdash q$. The proof is then by contradiction. Suppose $\mathcal{M}' \vdash q$. By using the connectedness of q and $\mathcal{M}' \vdash q$, we can ensure to find a match ν for the query in $\mathcal{M}$, which then leads to a contradiction.$\Box$

Every uniform $\mathcal{K}^{|q|}$-forest model $\mathcal{M}$ has a finite size. This is easy to see because there are finitely many $|q|$-trees in $\mathcal{M}$ that are distinct up to isomorphism. For bounds on the size of $\mathcal{K}^{|q|}$-forest models, the reader is referred to [11,12,14]. Here the interesting observation is that the size of $|q|$-trees in $\mathcal{M}$ depend on the size of the $\mathcal{T}$ and q. This can be explained as follows. Since the branching degree of $\mathcal{M}$ is bounded by $|\mathcal{T}|$, the branching degree of a $|q|$-tree in $\mathcal{M}$ is also bounded by $|\mathcal{T}|$. Moreover, the height of a $|q|$-tree is bounded by $|q|$. This means, if we take $|\mathcal{T}|$ and $|q|$ as constant, which is what we will do for data complexity, we have that the size of each tree in $\mathcal{M}$ is constant.

Our algorithm for deciding the non entailment of a boolean CQ q from a KB $\mathcal{K}$ is as follows. We first guess a $\mathcal{K}^{|q|}$-forest $\mathcal{M}$ with $\mathsf{bdegree}(\mathcal{M}) \leq |\mathcal{T}|$ and $\mathsf{roots}(\mathcal{M}) \leq |\mathcal{D}| + |\mathcal{D}| \cdot |\mathcal{T}|$. Since by assumption, $|\mathcal{T}|$ and $|q|$ are constant, the size of $\mathcal{M}$ is polynomial in $|\mathcal{D}|$. Verifying if $\mathcal{M}$ is a uniform $\mathcal{K}^{|q|}$-forest model can be done in polynomial time. If $\mathcal{M}$ is not a $\mathcal{K}^{|q|}$-forest model then we return "no". Otherwise, we verify if q matches against this structure. Since $|q|$ is constant, this can be done in time polynomial in $|\mathcal{D}|$. Hence, we have a non-deterministic algorithm that runs in PTime and decides the non-entailment of q from $\mathcal{K}$. This immediately yields a coNP upper bound for deciding CQ entailment from a KB. Then by Lemma 4.1, we obtain the following theorem.

Theorem 4.16 *CQ entailment in* $\mathsf{DL\text{-}Lite}_{\mathcal{F}}$ *with DBoxes is* coNP*-complete for data complexity.*

5 Relating $\mathcal{ALCFIO}$ to $\mathcal{ALCFID}$

The exact (combined) complexity of CQ entailment in $\mathcal{ALCFIO}$ (and its extensions above) is a major open problem in DLs: it is known to be decidable [14] (without any upper complexity bound) and coN2ExpTime-hard [7]. In this section, we prove Theorem 5.1, which gives a new perspective to this problem in terms of DBoxes. We believe that this may be useful for tackling the problem.

Theorem 5.1 *CQ entailment in* $\mathcal{ALCFI}$ *with DBoxes is reducible to CQ entailment in* $\mathcal{ALCFIO}$ *with ABoxes and vice versa.*

We start by reducing reasoning with DBoxes to reasoning with nominals in a rather straightforward way. Let $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ be a $\mathcal{ALCFI}$-KB, $A^{\mathcal{D}} = \{a \in \mathsf{adom}(\mathcal{D}) \mid A(a) \in \mathcal{D}$, and $\exists P^{\mathcal{D}} = \{a \in \mathsf{adom}(\mathcal{D}) \mid \exists a' \in \mathsf{adom}(\mathcal{D}), P(a, a') \in \mathcal{D}\}$. $\tau_{\mathcal{O}}(\mathcal{K}) = (\mathcal{T} \cup \mathcal{T}', \mathcal{A})$ is the $\mathcal{ALCFIO}$-KB such that $\mathcal{A} = \mathcal{D}$ and $\mathcal{T}'$ consist of the following sets of inclusion axioms:

$$\begin{aligned}
\mathcal{T}_{\mathbf{UNA}} &= \{\{a\} \sqsubseteq \neg\{b\} \mid a, b \in \mathsf{adom}(\mathcal{D}), a \neq b\};\\
\mathcal{T}_{\mathsf{con}(\mathcal{D})} &= \big\{A \equiv \{a_1\} \sqcup \ldots \sqcup \{a_n\} \mid A \in \mathsf{con}(\mathcal{D}), a_i \in A^{\mathcal{D}}, 1 \leq i \leq n\big\};\\
\mathcal{T}_{\mathsf{rol}(\mathcal{D})} &= \{\{a\} \sqsubseteq \exists P.\{b_1\} \sqcap \ldots \sqcap \exists P.\{b_n\} \mid P \in \mathsf{rol}(\mathcal{D}), P(a, b_i) \in \mathcal{D}, 1 \leq i \leq n\};\\
&\quad \cup \{\{a\} \sqsubseteq \forall P.(\{b_1\} \sqcup \ldots \sqcup \{b_n\}) \mid P \in \mathsf{rol}(\mathcal{D}), P(a, b_i) \in \mathcal{D}, 1 \leq i \leq n\};\\
&\quad \cup \Big\{\exists P.\top \sqsubseteq \{a_1\} \sqcup \ldots \sqcup \{a_n\} \mid P \in \mathsf{rol}(\mathcal{D}), a_i \in (\exists P)^{\mathcal{D}}, 1 \leq i \leq n\Big\}.
\end{aligned}$$

In the definition above, $\mathcal{T}_{\mathrm{UNA}}$ ensures that UNA for DBox individuals is preserved, $\mathcal{T}_{\mathsf{con}(\mathcal{D})}$ fixes the extension of concept names appearing in $\mathcal{D}$, and $\mathcal{T}_{\mathsf{rol}(\mathcal{D})}$ fixes the extension of a role names appearing in $\mathcal{D}$.

Lemma 5.2 *Let* $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ *be a* $\mathcal{ALCFI}$*-KB and* q *be a CQ. Then* $\mathcal{K} \models q$ *if and only if* $\tau_{\mathcal{O}}(\mathcal{K}) \models q$.

Note that $\mathcal{T}'$ is a TBox in $\mathcal{ALCO}$. Thus, 5.2 holds for $\mathcal{ALC}$ with DBoxes and any of its extensions we consider.

The reduction on the other direction is a little bit more intricate since UNA is not typically made in $\mathcal{ALCFIO}$; but we have it in $\mathcal{ALCFI}$, if we consider DBoxes. We shall show that, CQ entailment in $\mathcal{ALCFIO}$ without UNA can be reduced to query entailment w.r.t. an $\mathcal{ALCFI}$-KB with DBox $\mathcal{D}$, where $\sharp\mathsf{adom}(\mathcal{D}) = 1$. Note that in such KBs the UNA is not relevant.

Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{ALCFIO}$-KB, and q a CQ. We construct the KB $\tau_{\mathcal{D}}(\mathcal{K})$, and the query $\delta(q)$, such that one single individual occurs in $\tau_{\mathcal{D}}(\mathcal{K})$. For each $a \in \mathsf{adom}(\mathcal{K})$, introduce a new concept name A_a, and a new role name R_a. We also use a fresh individual name $o \in \mathsf{N}_I$ such that $o \notin \mathsf{adom}(\mathcal{K}) \cup \mathsf{adom}(q)$, and a concept name A_o. For each $C \in \mathsf{con}(\mathcal{K})$, denote by $\delta(C)$ the concept obtained from C by replacing every nominal $\{a\}$ occurring in C with A_a. We extend δ to $\mathcal{T}$ as follows: $\delta(\mathcal{T}) = \{\delta(C) \sqsubseteq \delta(D) \mid C \sqsubseteq D \in \mathcal{T}\}$.

Set $\mathcal{D} = \{A_o(o)\}$. Then, $\tau_{\mathcal{D}}(\mathcal{K}) = (\delta(\mathcal{T}) \cup \mathcal{T}', \mathcal{D})$, where $\mathcal{T}'$ consists of the following axioms for each $a \in \mathsf{adom}(\mathcal{K})$.:

- $A_a \sqsubseteq \exists R_a^-.A_o$,
- $A_o \sqsubseteq \leq 1 R_a$,
- $A_o \sqsubseteq \exists R_a.A_a$

Furthermore, for each $C(a) \in \mathcal{A}$, and each $P(a, b) \in \mathcal{A}$, we have the following axioms:

- $A_o \sqsubseteq \exists R_a.\delta(C)$,
- $A_o \sqsubseteq \exists R_a.\exists P.A_b$.

Finally, $\delta(q)$ is the CQ obtained from q by replacing every occurrence of $a \in \mathsf{adom}(\mathcal{K})$ in q with a new variable x_a, and appending the atom $A_a(x_a)$ to q.

Intuitively, A_a acts as the nominal $\{a\}$. With the axioms above we connect the instances of concept names A_a around the only instance of A_o with the functional role R_a. This guarantees that there is at most one instance of A_a in every model of $\tau_{\mathcal{D}}(\mathcal{K})$. The following lemma shows the correctness of the reduction.

Lemma 5.3 *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a $\mathcal{ALCFIO}$-KB and q be a CQ. Then $\mathcal{K} \models q$ if and only if $\tau_{\mathcal{D}}(\mathcal{K}) \models \delta(q)$.*

6 Discussion

In this paper, we characterized the data complexity of query entailment in $\mathsf{DL\text{-}Lite}_{\mathcal{F}}$ with DBoxes (Theorem 4.16). The exact combined complexity of this problem remains open. It would also be interesting to study these problems for other logics in the $\mathsf{DL\text{-}Lite}$ family. The coNP lower bound argument (Lemma 4.1) goes through even for a very simple logic in this family. Then the challenge is to identify a fragment of $\mathsf{DL\text{-}Lite}$ for which query answering with DBoxes is tractable. In the case of expressive DLs, we showed that the combined complexity of query entailment in $\mathcal{ALCFIO}$ is complete for the same complexity class as query entailment in $\mathcal{ALCFI}$ with DBoxes (Theorem 5.1). Our reductions do not work for the case of data complexity since they encode the data into the TBox, which is supposed to be fixed. It is an interesting question if such a characterization also exists for data complexity.

References

[1] Abiteboul, S., R. Hull and V. Vianu, “Foundations of Databases,” Addison-Wesley, 1995.

[2] Areces, C. and B. ten Cate, *Hybrid logics*, in: P. Blackburn, J. van Benthem and F. Wolter, editors, *Handbook of Modal Logic*, Elsevier, 2007.

[3] Baader, F., D. Calvanese, D. L. McGuinness, D. Nardi and P. F. Patel-Schneider, editors, “The Description Logic Handbook: Theory, Implementation, and Applications,” Cambridge University Press, 2003.

[4] Blackburn, P., M. de Rijke and Y. Venema, “Modal logic,” Cambridge University Press, New York, NY, USA, 2001.

[5] Calvanese, D., G. D. Giacomo, D. Lembo, M. Lenzerini and R. Rosati, *Data complexity of query answering in description logics*, in: P. Doherty, J. Mylopoulos and C. A. Welty, editors, *KR* (2006), pp. 260–270.

[6] Calvanese, D., G. D. Giacomo, D. Lembo, M. Lenzerini and R. Rosati, *Tractable reasoning and efficient query answering in description logics: The DL-Lite family* **39** (2007), pp. 385–429.

[7] Glimm, B., Y. Kazakov and C. Lutz, *Status qio: An update*, in: *Proceedings of the 24th International Workshop on Description Logics*, 2011, to appear.

[8] Glimm, B., C. Lutz, I. Horrocks and U. Sattler, *Conjunctive query answering for the description logic shiq*, J. Artif. Intell. Res. (JAIR) **31** (2008), pp. 157–204.

[9] Goré, R., *Tableau methods for modal and temporal logics*, in: M. D’Agostino, D. M. Gabbay, R. Hahnle and J. Posegga, editors, *Handbook of Tableau Methods* (1999), pp. 297–396.

[10] Horrocks, I. and U. Sattler, *A tableau decision procedure for $\mathcal{SHOIQ}$*, J. Autom. Reason. **39** (2007), pp. 249–276.

[11] Levy, A. Y. and M.-C. Rousset, *Combining horn rules and description logics in carin*, Artif. Intell. **104** (1998), pp. 165–209.

[12] Ortiz, M., D. Calvanese and T. Eiter, *Data complexity of query answering in expressive description logics via tableaux*, J. Autom. Reasoning **41** (2008), pp. 61–98.

[13] Papadimitriou, C. H., “Computational complexity,” Addison-Wesley, 1994.

[14] Rudolph, S. and B. Glimm, *Nominals, inverses, counting, and conjunctive queries or: Why infinity is your friend!*, J. Artif. Intell. Res. (JAIR) **39** (2010), pp. 429–481.

[15] Schwendimann, S., *A new one-pass tableau calculus for pltl*, in: *TABLEAUX ’98: Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods* (1998), pp. 277–292.

[16] Seylan, I., E. Franconi and J. de Bruijn, *Effective query rewriting with ontologies over dboxes*, in: C. Boutilier, editor, *IJCAI*, 2009, pp. 923–925.