An Update on Query Answering with Restricted Forms of Negation

Víctor Gutiérrez-Basulto[1], Yazmín Ibañez-García[2], and Roman Kontchakov[3]

[1] Fachbereich Mathematik und Informatik, Universität Bremen, Germany
victor@informatik.uni-bremen.de
[2] KRDB Research Centre, Free University of Bozen-Bolzano, Italy
ibanezgarcia@inf.unibz.it
[3] Department of CS and Information Systems, Birkbeck College, London, UK
roman@dcs.bbk.ac.uk

Abstract. One of the most prominent applications of description logic ontologies is their use for accessing data. In this setting, ontologies provide an abstract conceptual layer of the data schema, and queries over the ontology are then used to access the data. In this paper we focus on extensions of conjunctive queries (CQs) and unions of conjunctive queries (UCQs) with restricted forms of negations such as inequality and safe negation. In particular, we consider ontologies based on members of the *DL-Lite* family. We show that by extending UCQs with any form of negated atoms, the problem of query answering becomes undecidable even when considering ontologies expressed in the core fragment of *DL-Lite*. On the other hand, we show that answering CQs with inequalities is decidable for ontologies expressed in $\textit{DL-Lite}_{core}^{\mathcal{H}}$. To this end, we provide an algorithm matching the known coNP lower bound on data complexity. Furthermore, we identify a setting in which conjunctive query answering with inequalities is tractable. We regain tractability by means of syntactic restrictions on the queries, but keeping the expressiveness of the ontology.

1 Introduction

In recent years, the use of ontologies for accessing data has been recognized as one of the most prominent applications of description logics (DLs) in the Semantic Web (SW) and relational databases. The characteristic feature of *ontology-based data access* (OBDA) is the use of ontologies to enrich instance data with background knowledge, thus providing users with an interface for querying potentially incomplete data. The importance of OBDA as a key technology for the SW has been acknowledged by introduction of the Web Ontology Language (OWL) and its profiles based on tractable DLs. In the OBDA paradigm the study of query answering has mainly been focused on answering (unions of) conjunctive queries (CQs). In particular, a fairly clear landscape of the computational complexity of CQ answering has emerged, and specific algorithmic approaches have already been developed. Recently, some investigations on query answering using query languages beyond CQs have been initiated [19,6]. In particular, a desirable way to extend CQs, which belong to the positive existential fragment of first-order logic, is with some form of *negation*. Following the large body of literature on relational databases, we consider two ways of adding restricted forms of negation to CQs:

inequalities as atomic formulas ($CQ^{\neq}$) and *safe negation* ($CQ^{\neg s}$). In the OBDA setting, besides the class of queries, the DL for representing the ontology needs to be specified. Of special interest are those DLs allowing OBDA to scale to large amounts of data by answering queries in relational database management systems (RDBMSs). This is the case for the members of the *DL-Lite* family of DLs: *DL-Lite*$_{core}$ and *DL-Lite*$_{core}^{\mathcal{H}}$ [4]. Remarkably, CQ answering in these logics is in AC^0 in *data complexity*, which is an important measure of complexity when large amounts of data are considered.

The aim of this paper is to continue the study initiated by Rosati [19] on answering (U)$CQs^{\neq}$ and (U)$CQs^{\neg s}$ in DLs of the *DL-Lite* family. In particular, we provide undecidability and complexity results for answering (U)$CQs^{\neq}$, along with algorithmic approaches. Moreover, inspired by recent works we introduce syntactical restrictions to obtain tractable $CQ^{\neq}$ answering.

Related Work

In recent years, extensions of CQs with some form of negation have been studied in different areas of computer science related to management of incomplete information. The main research done in this respect focuses on establishing decidability boundaries, complexity results and algorithms for query answering. We outline relevant results in some of these areas below.

CQs with Inequalities and Negation in Description Logics Calvanese *et al.* [9] showed that in contrast to CQs, answering $CQs^{\neq}$ in highly expressive DL $\mathcal{DLR}$ is undecidable. Later on, Rosati [18,19] presented a deeper study of query answering with restricted forms of negation in several DLs by considering not only inequalities but also safe negation. Rosati shows undecidability of answering CQs with any form of negation in the DL $\mathcal{AL}$. Furthermore, Rosati shows that answering UCQs with any form of negation in fairly inexpressive DLs $\mathcal{EL}$ and *DL-Lite*$_{core}^{\mathcal{H}}$ (called *DL-Lite*$_{\mathcal{R}}$ in the paper) is undecidable. For the case of answering $CQs^{\neq}$ and $CQs^{\neg s}$ in *DL-Lite*$_{core}^{\mathcal{H}}$ Rosati provides a coNP-hardness result in data complexity, leaving the exact complexity of the problem (and even decidability) open.

CQs with Inequalities in Data Exchange (DE) In their seminal work, Fagin *et al.* [12] showed that in the DE setting answering $UCQs^{\neq}$ with target constraints given by weakly acyclic TGDs is coNP-complete. To provide the upper complexity bound they presented a procedure based on a variant of the *disjunctive chase* introduced by Deutsch *et al.* [11]. A remarkable contribution of this work is a PTIME algorithm for computing certain answers of UCQs with at *most* one inequality per disjunct. The lower bound for an arbitrary number of inequalities follows from a result previously established by Abiteboul *et al.* [1].

$CQs^{\neq}$ with Bounded Number of Inequalities It is known that the complexity of answering $UCQs^{\neq}$ can be affected by the number of inequalities allowed per query [15]. In settings dealing with incomplete information, Abiteboul *et al.* [1,2] showed in their work on answering queries via views that answering $UCQs^{\neq}$ is coNP-complete. In particular, their coNP-hardness proof (also utilized as a lower bound in the DE setting) requires six inequalities. However, later on Madry [17] closed the gap in the DE setting by showing that even the case of two inequalities is intractable.

CQs$^{\neq}$ with Other Syntactic Restrictions An orthogonal restriction to that on the number of inequalities has recently been proposed and investigated in the DE setting by Arenas *et al.* [3] on extensions of Datalog with negated atoms. Their approach is to define syntactic restrictions over the variables that can occur in inequalities. In particular, under such conditions one can have more than one inequality per disjunct without losing tractability.

Our paper is organized as follows. In Section 2, we provide Description Logic definitions. Section 3 is dedicated to the presentation of lower complexity bounds. Section 4 investigates the establishment of matching upper bounds. Section 5 studies syntactic restrictions over conjunctive queries with inequalities. Finally, in Section 6 we conclude with an outlook of the contribution and future research lines.

2 Preliminaries

In this section we recall some basics on description logics (DLs) and extensions of conjunctive queries (CQs) with negated atoms.

The Description Logic *DL-Lite*$_{core}^{\mathcal{H}}$: Syntax and Semantics

The language of *DL-Lite*$_{core}^{\mathcal{H}}$ [4] contains *individual names* $a_0, a_1, \ldots$, *concept names* $A_0, A_1, \ldots$, and *role names* $P_0, P_1, \ldots$. We define *complex roles* R and *basic concepts* B using the following grammar:

$$\begin{array}{rcl} R & ::= & P_i \mid P_i^-, \\ B & ::= & \bot \mid A_i \mid \exists R. \end{array}$$

A *DL-Lite*$_{core}^{\mathcal{H}}$ *TBox* $\mathcal{T}$ is a finite set of *concept* and *role inclusion axioms* of the form:

$$B_1 \sqsubseteq B_2, \qquad B_1 \sqsubseteq \neg B_2, \qquad R_1 \sqsubseteq R_2, \qquad R_1 \sqsubseteq \neg R_2.$$

Whenever we find it convenient we might use $B_1 \sqcap B_2 \sqsubseteq \bot$ instead of the equivalent $B_1 \sqsubseteq \neg B_2$. An *ABox* $\mathcal{A}$ is a finite set of *assertions* of the form:

$$A_k(a_i), \qquad P_k(a_i, a_j).$$

A *DL-Lite*$_{core}^{\mathcal{H}}$ *knowledge base* (*KB*) $\mathcal{K}$ is a pair $(\mathcal{T}, \mathcal{A})$ with $\mathcal{T}$ a TBox and $\mathcal{A}$ an ABox. In the following, we denote by $ind(\mathcal{A})$ the set of individual names occurring in $\mathcal{A}$, and by $role^{\pm}(\mathcal{K})$ the set of roles that consists of P_k and P_k^-, for each role name P_k in $\mathcal{K}$. *DL-Lite*$_{core}$ is the fragment of *DL-Lite*$_{core}^{\mathcal{H}}$ without role inclusion axioms in the TBox.

An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a nonempty *domain* $\Delta^{\mathcal{I}}$ and an interpretation function $\cdot^{\mathcal{I}}$ that assigns an element $a_i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ to each object name a_i, a subset $A_k^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ to each concept name A_k, and a binary relation $P_k^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ to each role name P_k. As usual for *DL-Lite*, we adopt the *unique name assumption* (UNA): $a_i^{\mathcal{I}} \neq a_j^{\mathcal{I}}$, for all distinct individuals a_i, a_j.

The interpretation function $\cdot^{\mathcal{I}}$ is then extended to basic concepts and complex roles:

$$
\begin{aligned}
(P_k^-)^{\mathcal{I}} &= \{(y,x) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (x,y) \in P_k^{\mathcal{I}}\}, && \text{(inverse role)}\\
\bot^{\mathcal{I}} &= \emptyset, && \text{(empty set)}\\
(\exists R)^{\mathcal{I}} &= \left\{x \in \Delta^{\mathcal{I}} \mid \text{there is } y \in \Delta^{\mathcal{I}} \text{ with } (x,y) \in R^{\mathcal{I}}\right\}. && \text{(domain/range constraints)}
\end{aligned}
$$

We define the *satisfaction relation* $\models$ in a standard way:

$$
\begin{aligned}
\mathcal{I} \models B_1 \sqsubseteq B_2 \quad &\text{iff} \quad B_1^{\mathcal{I}} \subseteq B_2^{\mathcal{I}}, & \mathcal{I} \models R_1 \sqsubseteq R_2 \quad &\text{iff} \quad R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}},\\
\mathcal{I} \models B_1 \sqsubseteq \neg B_2 \quad &\text{iff} \quad B_1^{\mathcal{I}} \cap B_2^{\mathcal{I}} = \emptyset, & \mathcal{I} \models R_1 \sqsubseteq \neg R_2 \quad &\text{iff} \quad R_1^{\mathcal{I}} \cap R_2^{\mathcal{I}} = \emptyset,\\
\mathcal{I} \models A_k(a_i) \quad &\text{iff} \quad a_i^{\mathcal{I}} \in A_k^{\mathcal{I}}, & \mathcal{I} \models P_k(a_i,a_j) \quad &\text{iff} \quad (a_i^{\mathcal{I}}, a_j^{\mathcal{I}}) \in P_k^{\mathcal{I}}.
\end{aligned}
$$

A KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is *satisfiable* if there is an interpretation $\mathcal{I}$ satisfying all members of $\mathcal{T}$ and $\mathcal{A}$. In this case we write $\mathcal{I} \models \mathcal{K}$ (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 $\mathcal{A}$).

Conjunctive Queries with Restricted Forms of Negation

A *conjunctive query* (CQ) is an expression of the form

$$q(\boldsymbol{x}) \;=\; \exists \boldsymbol{y}\, \varphi(\boldsymbol{x}, \boldsymbol{y}), \tag{1}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ denote sequences of variables from a set of variables, and φ is conjunction of concept atoms $A(t)$ and role atoms $P(t,t')$ with t, t' *terms*, i.e., individual names or variables from $\boldsymbol{x}, \boldsymbol{y}$. We call variables in $\boldsymbol{x}$ *answer* variables and those in $\boldsymbol{y}$ (existentially) *quantified* variables. We denote by $\mathsf{var}(q)$ the set of variables, by $\mathsf{avar}(q)$ the set of answer variables $\boldsymbol{x}$, by $\mathsf{qvar}(q)$ the set of quantified variables $\boldsymbol{y}$ and by $\mathsf{term}(q)$ the set of terms in q. A *conjunctive query with inequalities* ($\text{CQ}^{\neq}$) is an expression of the form (1) with each conjunct of $\varphi(\boldsymbol{x}, \boldsymbol{y})$ being either a concept or role atom, or an expression of the form $t \neq t'$, where t and t' are terms. A *conjunctive query with safe negation* ($\text{CQ}^{\neg s}$) is an expression of the form (1) where $\varphi(\boldsymbol{x}, \boldsymbol{y})$ is formed by literals, i.e., atoms or negated atoms, and such that each variable of each literal occurs in at least one positive atom. A *union of conjunctive queries* (UCQ) is a disjunction of conjunctive queries. $\text{UCQ}^{\neq}$ and $\text{UCQ}^{\neg s}$ are defined accordingly.

Query Answering over *DL-Lite* KBs Let $\mathcal{I}$ be an interpretation and $q(\boldsymbol{x})$ a query with $\boldsymbol{x} = x_1, \ldots, x_k$. A map $\pi \colon \mathsf{term}(q) \to \Delta^{\mathcal{I}}$ with $\pi(a) = a^{\mathcal{I}}$, for a an individual name in $\mathsf{term}(q)$, is called a *match* for q in $\mathcal{I}$ if $\mathcal{I}$ satisfies q under the variable assignment that maps each answer variable x_i to $\pi(x_i)$. For a k-tuple of individual names $\boldsymbol{a} = a_1, \ldots, a_k$, a match π for q in $\mathcal{I}$ is called an $\boldsymbol{a}$*-match* if $\pi(x_i) = a_i^{\mathcal{I}}$. We say that $\boldsymbol{a}$ is an *answer* to q in an interpretation $\mathcal{I}$ if there is an $\boldsymbol{a}$-match for q in $\mathcal{I}$. We denote by $\mathsf{ans}(q, \mathcal{I})$ the set of all answers to q in $\mathcal{I}$. We say that $\boldsymbol{a} \subseteq ind(\mathcal{A})$ is a *certain answer* to q over a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ if $\boldsymbol{a} \in \mathsf{ans}(q, \mathcal{I})$, for all models $\mathcal{I}$ of $\mathcal{K}$. The set of all *certain answers* to q over $\mathcal{K}$ is denoted by $\mathsf{cert}(q, \mathcal{K})$. We consider the following query answering problem:

INPUT:	A query q, a *DL-Lite*$_{core}^{\mathcal{H}}$ KB $\mathcal{K}$ and a tuple of individuals $\boldsymbol{a}$.
QUESTION:	Is $\boldsymbol{a}$ in $\mathsf{cert}(q, \mathcal{K})$?

3 Lower Complexity Bounds

First, we analyse the case of unions of conjunctive queries and show that query answering with inequalities is undecidable even in the simplest of *DL-Lite* languages. In fact, the following proof will demonstrate that even though the ontology language is quite inexpressive, undecidable problems can still be encoded by means of (mostly) UCQs. In a nutshell, the proof uses the existential quantifiers of the TBox concept inclusion axioms to create an unbounded supply of elements, whereas the UCQ$^{\neq}$ allows one to express universal constraints in the following sense: the query has a positive answer iff there is no model of the KB satisfying the negated UCQ$^{\neq}$, which is a conjunction of universal sentences. We remark here that the result is claimed in Theorem 8 [19], however no proof is given.

Theorem 1. *Answering UCQs$^{\neq}$ is undecidable over DL-Lite$_{core}$ KBs.*

Proof. The proof is by reduction of (the complement of) the $\mathbb{N} \times \mathbb{N}$-tiling problem, which is known to be undecidable [14]. The $\mathbb{N} \times \mathbb{N}$ tiling problem is formulated as follows: given a set $\mathfrak{T}$ of square tile types with the four sides of each tile type t in $\mathfrak{T}$ coloured by $\mathit{top}(t), \mathit{right}(t), \mathit{bottom}(t), \mathit{left}(t)$, respectively, and a tile type $t_0 \in \mathfrak{T}$, decide whether $\mathbb{N} \times \mathbb{N}$ can be tiled by $\mathfrak{T}$ with t_0 placed at the origin, i.e., whether there is a function $\tau \colon \mathbb{N} \times \mathbb{N} \to \mathfrak{T}$ such that $\tau(0,0) = t_0$ and $\mathit{top}(\tau(i,j)) = \mathit{bottom}(\tau(i,j+1))$ and $\mathit{left}(\tau(i,j)) = \mathit{right}(\tau(i+1,j))$, for all $(i,j) \in \mathbb{N} \times \mathbb{N}$.

Given an instance of the $\mathbb{N} \times \mathbb{N}$-tiling problem, we construct a *DL-Lite$_{core}$* KB $(\mathcal{T}, \mathcal{A})$ that encodes the tiling problem by placing tiles over objects in its model. The top and right neighbours of a tile are referred to by roles H and V, respectively (from the horizontal and vertical successor). To represent the type of a tile we take ABox individuals t_i, for $t_i \in \mathfrak{T}$, and a role T that connects a tile to its type. So, the TBox $\mathcal{T}$ contains the following concept inclusions:

$$\exists T \sqsubseteq \exists H, \qquad \exists H^- \sqsubseteq \exists T, \qquad \exists T \sqsubseteq \exists V, \qquad \exists V^- \sqsubseteq \exists T.$$

We also require two roles, N_H and H_V, that define impossible horizontal and vertical tile neighbours: let $\mathcal{A}_{\mathfrak{T}}$ contain

$$\begin{array}{ll} N_H(t_i, t_j), & \text{for each } t_i, t_j \in \mathfrak{T} \text{ with } \mathit{right}(t_i) \neq \mathit{left}(t_j), \\ N_V(t_i, t_j), & \text{for each } t_i, t_j \in \mathfrak{T} \text{ with } \mathit{top}(t_i) \neq \mathit{bottom}(t_j). \end{array}$$

Consider now the UCQ$^{\neq}$ q (without answer variables) which consists of the negations of the following sentences:

$$\begin{array}{l} \forall x, y \, \big(T(x,y) \to \bigvee_i (y = t_i)\big), \\ \forall x, y, z, v, u \, \big(H(x,y) \wedge V(y,v) \wedge V(x,z) \wedge H(z,u) \to (u = v)\big), \\ \forall x, y, x', y' \, \big(H(x,y) \wedge T(x,x') \wedge T(y,y') \wedge N_H(x',y') \to \bot\big), \\ \forall x, y, x', y' \, \big(V(x,y) \wedge T(x,x') \wedge T(y,y') \wedge N_V(x',y') \to \bot\big). \end{array}$$

It can be shown that q has a negative answer over $(\mathcal{T}, \mathcal{A}_{\mathfrak{T}} \cup \{T(a, t_0)\})$ iff $\mathfrak{T}$ tiles $\mathbb{N} \times \mathbb{N}$ with t_0 placed at the origin. Indeed, if q has a negative answer then the above formulas guarantee that each tile object is related to one of the t_i, that the H- and V-successors from a proper $\mathbb{N} \times \mathbb{N}$-grid and, finally, that the adjacent colours match.

We remark in passing that answering UCQs with safe negation is undecidable even over extremely simple ontology languages [19], including *DL-Lite*$_{core}$. Although the proof of Theorem 15 [19] does not directly apply to *DL-Lite*$_{core}$, it can easily be adapted for our language:

Theorem 2 ([19]). *Answering UCQs$^{\neg s}$ is undecidable over DL-Lite$_{core}$ KBs.*

We show that even in the case of conjunctive queries adding inequalities makes the query answering problem in *DL-Lite*$_{core}$ harder. In particular, we show that answering CQs$^{\neq}$ is coNP-hard in data complexity in contrast to answering CQs, which is in AC^0. Our result strengthens Theorem 15 [19], which claims coNP-hardness for *DL-Lite*$_{core}^{\mathcal{H}}$ and refers to Abiteboul and Duschka [2] for the proof. That proof, however, is for a different first-order setting and, if translated to the language of DL, would require role inclusions and a sort of a counting quantifier in the CQ, which of course, can be expressed using inequality. We mention in passing that this proof would also imply coNP-hardness (in data complexity) of the satisfiability problem for the extension of *DL-Lite*$_{core}$ with arbitrary number restriction (cf. Theorem 8.4 [4]). Although the proof we present below is inspired by Theorem 3.4 [2], it does not use role inclusions and requires a more sophisticated query instead.

Theorem 3. *Answering CQs$^{\neq}$ over DL-Lite$_{core}$ KBs is* coNP*-hard in data complexity.*

Proof. The proof is by reduction of the complement of 3CNF. Suppose we are given a 3CNF φ with n clauses and m variables. We construct a KB $(\mathcal{T}, \mathcal{A}_\varphi)$ and a query q such that both $\mathcal{T}$ and q are fixed (i.e., do not depend on φ) and φ is satisfiable iff q has a negative answer over $(\mathcal{T}, \mathcal{A}_\varphi)$. We present the construction in two steps. To aid our explanations, we consider a model of $(\mathcal{T}, \mathcal{A}_\varphi)$ in which q is false.

First, we take a concept name V to stand for the set of variables of φ and three individuals v_j^0, v_j^1, x_j, for each of the m variables x_j of φ: one may think that v_j^0 represents the literal $\neg x_j$ and v_j^1 represents the literal x_j. The ABox $\mathcal{A}_\varphi$ contains, for each x_j, the following assertions:

$$V(x_j), \quad P_0(x_j, v_j^0), \quad P_1(x_j, v_j^1), \quad P_2(x_j, x_j) \qquad \text{and} \quad R_1(x_j, t),$$

where t is a fresh individual (t stands for *true*) and P_0, P_1, P_2 and R_1 are role names. So, every x_j has a P_i-successor, for each $0 \leq i \leq 2$, and an R_1-successor; moreover, by the UNA, the R_1-successor is distinct from the P_i-successors. Then, the TBox $\mathcal{T}$ contains the following two concept inclusions

$$V \sqsubseteq \exists R_0 \quad \text{and} \quad V \sqcap \exists R_0^- \sqsubseteq \bot,$$

where R_0 is a fresh role name, to ensure that every x_j also has an R_0-successor and that R_0-successor is not x_j itself; see Fig. 1 (a).

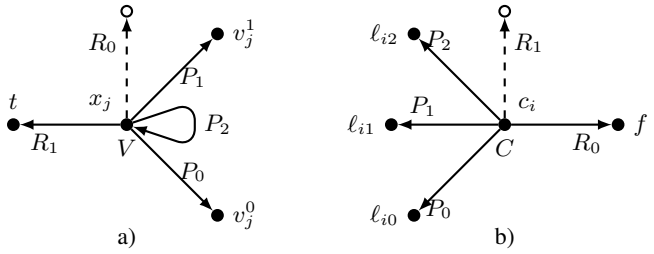

Fig. 1. Constellations of points in the proof of Theorem 3.

Consider now a CQ$^{\neq}$ q (without answer variables) which is equivalent to the following sentence:

$$\forall x, y_1, y_2, y_3, z_0, z_1 \Big(\bigwedge_{i=0}^{2} P_i(x, y_i) \ \wedge \bigwedge_{k=0,1} R_k(x, z_k) \ \rightarrow \ \bigvee_{i=0}^{2} \bigvee_{k=0,1} (y_i = z_k) \Big).$$

If q has a negative answer then, for any point with P_i- and R_k-successors, either its R_0- or its R_1-successor coincides with one of the P_i-successors. In particular, when applied to individuals x_j, this means that the R_0-successor must coincide either with v_j^0 or v_j^1—in the latter case the literal $\neg x_j$ (represented by v_j^0) is chosen false by R_0 (and so, we say the variable x_j takes value *true*) and in the former case the literal x_j (represented by v_j^1) is chosen false by R_0 (and we say x_j takes value value *false*).

Second, we encode clauses in a similar way: we take a concept name C to stand for the set of clauses of φ and an individual c_i, for each of the n clauses of φ. Then the ABox $\mathcal{A}_\varphi$ contains the following assertions, for each clause $c_i = L_{i0} \vee L_{i1} \vee L_{i2}$:

$$C(c_i), \quad P_0(c_i, \ell_{i0}), \quad P_1(c_i, \ell_{i1}), \quad P_2(c_i, \ell_{i2}) \quad \text{and} \quad R_0(c_i, f),$$

where $\ell_{ik} = v_j^0$ if $L_{ik} = \neg x_j$ and $\ell_{ik} = v_j^1$ if $L_{ik} = x_j$, for each $0 \leq k \leq 2$, and f is a fresh individual (f stands for *false*). Similarly to the case of variables, we need the following concept inclusion in TBox $\mathcal{T}$:

$$C \sqsubseteq \exists R_1,$$

which, together with the ABox, ensures that every c_i has P_i-successors, for $0 \leq i \leq 2$, an R_0-successor (distinct from the P_i-successors) and an R_1-successor; see Fig. 1 (b). But then, if q has a negative answer, the R_1-successor must coincide with one of the P_i-successors. This choice of the P_i determines the literal of the clause that is required to be true—if the R_1-successor of c_i is v_j^0 then the variable x_j needs to be *false* and if it is v_j^1 then x_j needs to be *true*.

To sum up, the R_1-successors of the c_i identify the literals v_j^0/v_j^1 required to be true, while the R_0-successors of the x_j choose the literals v_j^0/v_j^1 required to be false. So, the last concept inclusion of $\mathcal{T}$ ensures the choices are consistent:

$$\exists R_0^- \sqcap \exists R_1^- \sqsubseteq \bot.$$

It should be clear that q has a negative answer over $(\mathcal{T}, \mathcal{A}_\varphi)$ iff the 3CNF φ is satisfiable.

4 Answering CQs$^{\neq}$ in *DL-Lite*$^{\mathcal{H}}_{core}$: Upper Complexity Bound

In this section, a CONP in data complexity algorithm to decide CQ$^{\neq}$ answering in *DL-Lite*$^{\mathcal{H}}_{core}$ is provided. We begin by recalling some important notions and properties of canonical models.

Canonical Model

The notion of the *canonical model* in DLs [8,4,16] is related to those of the *chase*, *universal model* and *universal solution* present in data exchange and data integration settings [11]. A key characteristic of *DL-Lite*$^{\mathcal{H}}_{core}$ ontologies is that they can be regarded as sets of Horn clauses, and so, for every satisfiable *DL-Lite*$^{\mathcal{H}}_{core}$ KB $\mathcal{K}$, there is a *universal model* $\mathcal{U}$ that can be homomorphically embedded into every other model $\mathcal{J}$ of $\mathcal{K}$. Since positive existential formulas (e.g., CQs) are preserved under homomorphisms, universal models clearly become handy in tackling the query answering problem in Horn DLs such as *DL-Lite*$^{\mathcal{H}}_{core}$ [8,16]. Next, we recall the definition of universal and canonical models, as well as some properties to be used in the rest of the paper.

Definition 1. *Given two interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ and $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$, a* homomorphism *from $\mathcal{I}$ to $\mathcal{J}$ is a mapping $h\colon \Delta^{\mathcal{I}} \to \Delta^{\mathcal{J}}$ satisfying the following conditions*:

1. $h(a^{\mathcal{I}}) = a^{\mathcal{J}}$*, for each individual name a,*
2. $h(d) \in A^{\mathcal{J}}$*, for every $d \in A^{\mathcal{I}}$ and each concept name A,*
3. $(h(d), h(e)) \in P^{\mathcal{J}}$*, for every $(d, e) \in P^{\mathcal{I}}$ and each role name P.*

An interpretation $\mathcal{U}$ is said to be a *universal model* of a KB $\mathcal{K}$ if, for every interpretation $\mathcal{J}$ with $\mathcal{J} \models \mathcal{K}$ there exist a homomorphism from $\mathcal{U}$ to $\mathcal{J}$.

Since CQs, and more generally UCQs, are positive existential formulas, they are preserved under homomorphisms and so, a standard way of computing certain answers to a given UCQ q over a KB $\mathcal{K}$ is evaluating q in a universal model $\mathcal{U}$ of $\mathcal{K}$:

Lemma 1. *Let $\mathcal{K}$ be a satisfiable DL-Lite KB and let $\mathcal{U}$ be a universal model of $\mathcal{K}$. Then* $\mathsf{cert}(q, \mathcal{K}) = \mathsf{ans}(q, \mathcal{U})$*, for each UCQ q.*

Kontchakov *et al.* [16] present a way of constructing a universal model of a given a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$. The constructed model is called the *canonical model* of $\mathcal{K}$ and is denoted $\mathcal{U}_{\mathcal{K}}$.

(*i*) First, the ABox $\mathcal{A}$ is saturated by applying the concept and role inclusions of $\mathcal{T}$ in a bottom-up fashion: e.g., if $A(a) \in \mathcal{A}$ and $\mathcal{T} \models A \sqsubseteq A'$ then the ABox is extended by $A'(a)$. Note that at this stage existential quantifiers do not create any new individuals. We denote the resulting ABox by $\mathcal{A}^{+}$.

(*ii*) On the second stage, new individuals d_R for roles R are created to witness all existential quantifiers that are not witnessed in the ABox $\mathcal{A}^{+}$: e.g., if $A(a)$ is in the ABox and $\mathcal{T} \models A \sqsubseteq \exists R$ but the ABox does not contain $R(a, b)$, for any b, then it is extended by all $S(a, d_R)$[4] for all $\mathcal{T} \models R \sqsubseteq S$ and all $A(d_R)$ with $\mathcal{T} \models \exists R^{-} \sqsubseteq A$; the *generating relation* $\rightsquigarrow$ is extended by (a, d_R); note that a here is not necessarily an individual from the ABox $\mathcal{A}$ and can also be another d_S.

[4] We write $R(a, b) \in \mathcal{A}$ for $P(a, b) \in \mathcal{A}$ if $R = P$ and $P(b, a) \in \mathcal{A}$ if $R = P^{-}$.

The ABox resulting from applying (*i*) and (*ii*) is clearly finite; the interpretation determined by this ABox is called the *generating interpretation* and is denoted by $\mathcal{I}_\mathcal{K}$. However, $\mathcal{I}_\mathcal{K}$ is not necessarily a universal model. A standard way to construct a universal model from $\mathcal{I}_\mathcal{K}$ is to *unravel* it into a forest-shaped interpretation. A *path* in $\mathcal{I}_\mathcal{K}$ is a finite sequence $ad_{R_1}\cdots d_{R_k}$ $k \geq 0$, where $a \in ind(\mathcal{A})$, $a \rightsquigarrow d_{R_1}$ and $d_{R_i} \rightsquigarrow d_{R_{i+1}}$. We use $\mathsf{paths}(\mathcal{I}_\mathcal{K})$ to denote the set of all paths in $\mathcal{I}_\mathcal{K}$ and $\mathsf{tail}(\sigma)$ to denote the last element of a path $\sigma \in \mathsf{paths}(\mathcal{I}_\mathcal{K})$. The *canonical model* $\mathcal{U}_\mathcal{K}$ of $\mathcal{K}$ is then defined as follows:

$$
\begin{aligned}
\Delta^{\mathcal{U}_\mathcal{K}} &= \mathsf{paths}(\mathcal{I}_\mathcal{K}),\\
a^{\mathcal{U}_\mathcal{K}} &= a, \text{ for all } a \in ind(\mathcal{A}),\\
A^{\mathcal{U}_\mathcal{K}} &= \{\sigma \in \Delta^{\mathcal{U}_\mathcal{K}} \mid \mathsf{tail}(\sigma) \in A^{\mathcal{U}_\mathcal{K}}\},\\
P^{\mathcal{U}_\mathcal{K}} &= \{(a,b) \in ind(\mathcal{A}) \times ind(\mathcal{A}) \mid P(a,b) \in \mathcal{A}\} \cup\\
&\quad\ \{(\sigma, \sigma \cdot d_R) \in \Delta^{\mathcal{U}_\mathcal{K}} \times \Delta^{\mathcal{U}_\mathcal{K}} \mid \mathcal{T} \models R \sqsubseteq P\} \cup\\
&\quad\ \{(\sigma \cdot d_R, \sigma) \in \Delta^{\mathcal{U}_\mathcal{K}} \times \Delta^{\mathcal{U}_\mathcal{K}} \mid \mathcal{T} \models R \sqsubseteq P^-\}.
\end{aligned}
$$

The canonical model $\mathcal{U}_\mathcal{K}$ of $\mathcal{K}$ enjoys the following structural properties:

(abox) $(a_i, a_j) \in R^{\mathcal{U}_\mathcal{K}}$ iff $R(a_i, a_j) \in \mathcal{A}^+$, for all individuals a_i, a_j and roles R;

(forest) the graph $G = (\Delta^{\mathcal{U}_\mathcal{K}}, E)$ with $E = \{(\sigma, \sigma \cdot d_R) \mid \sigma \cdot d_R \in \Delta^{\mathcal{U}_\mathcal{K}}\}$ is a forest; moreover, each ABox individual a induces a partitioning of the graph into disjoint labelled trees $\mathfrak{T}_a = (T_a, E_a, \ell_a)$ with nodes $T_a = \{\sigma \in \Delta^{\mathcal{U}_\mathcal{K}} \mid \sigma = a \cdot \sigma'\}$, edges $E_a = E \cap (T_a \times T_a)$ and labelling function $\ell_a \colon E_a \to \mathit{role}^{\pm}(\mathcal{K})$ such that, for every $\sigma, \sigma' \in T_a$, we have $(\sigma, \sigma') \in P^{\mathcal{U}_\mathcal{K}}$ iff

$$\text{either } \ell_a(\sigma, \sigma') = R \text{ and } \mathcal{T} \models R \sqsubseteq P \quad \text{or} \quad \ell_a(\sigma', \sigma) = R \text{ and } \mathcal{T} \models R \sqsubseteq P^-;$$

(iso) for each role R, all labelled subtrees generated by $\sigma \cdot d_R \in \Delta^{\mathcal{U}_\mathcal{K}}$ are isomorphic.

The following lemma is a consequence of the results by Kontchakov *et al.* [16]:

Lemma 2. *A DL-Lite$_{core}^{\mathcal{H}}$ KB $\mathcal{K}$ is satisfiable iff $\mathcal{U}_\mathcal{K} \models \mathcal{K}$.*

In contrast to the classical CQ answering problem, certain answers to CQ$^{\neq}$s over a KB $\mathcal{K}$ cannot be obtained by evaluating queries in the canonical model $\mathcal{U}_\mathcal{K}$. The main reason for this is that CQ$^{\neq}$s are not preserved under homomorphisms. Hence, the fact that $\boldsymbol{a} \in \mathsf{ans}(q, \mathcal{U}_\mathcal{K})$ does not necessarily imply that $\boldsymbol{a} \in \mathsf{ans}(q, \mathcal{I})$, for every model $\mathcal{I}$ of $\mathcal{K}$. We illustrate this situation by the following example, which is adapted from [11].

Example 1. Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a KB and q a CQ$^{\neq}$ with

$$
\begin{aligned}
&\mathcal{T} = \{\exists R_1 \sqsubseteq \exists R_2,\ \ \exists R_1^- \sqsubseteq \exists R_3^-,\ \ \exists R_2^- \sqsubseteq \exists R_3\},\\
&\mathcal{A} = \{R_1(a_1, b_1)\},\\
&q(x, z) = \exists y, y' \big(R_1(x, z) \wedge R_2(x, y) \wedge R_3(y', z) \wedge (y \neq y')\big).
\end{aligned}
$$

The canonical model $\mathcal{U}_\mathcal{K}$ is depicted in Fig. 2 on the left; it can be seen that $\mathsf{ans}(q, \mathcal{U}_\mathcal{K}) = \{(a_1, b_1)\}$. However, there is a model $\mathcal{J}$ of $\mathcal{K}$, depicted in Fig. 2 on the right, where $\mathsf{ans}(q, \mathcal{J}) = \emptyset$. Therefore, $\mathsf{cert}(q, \mathcal{K}) = \emptyset$.

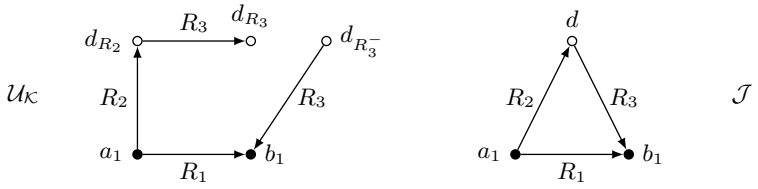

Fig. 2. The canonical model $\mathcal{U}_\mathcal{K}$ of $\mathcal{K}$ and another model $\mathcal{J}$ of $\mathcal{K}$.

The Decision Procedure

We proceed to show that the coNP lower complexity bound from Theorem 3 is in fact tight for answering CQs$^{\neq}$ over *DL-Lite*$^{\mathcal{H}}_{core}$ KBs and provide an algorithm for deciding CQ$^{\neq}$ answering: this non-deterministic algorithm will require time polynomial in the size of the given ABox. We observe that the problem of deciding, given a CQ$^{\neq}$ q and a KB $\mathcal{K}$, whether $\boldsymbol{a} \in \mathsf{cert}(q, \mathcal{K})$ can be reduced to the problem of deciding whether $\mathsf{cert}(q(\boldsymbol{a}), \mathcal{K}) \neq \emptyset$, where $q(\boldsymbol{a})$ is the query obtained by substituting $\boldsymbol{x}$ in q by $\boldsymbol{a}$; thus, $q(\boldsymbol{a})$ has no answer variables and is usually called a Boolean query. Furthermore, by the certain answer semantics, we can consider the problem of answering Boolean queries as a logical entailment problem, i.e., $\mathsf{cert}(q, \mathcal{K}) \neq \emptyset$ iff $\mathcal{K} \models q$, i.e., $\mathcal{I} \models q$ in every model $\mathcal{I}$ of $\mathcal{K}$. So, $\mathsf{cert}(q, \mathcal{K}) = \emptyset$ iff

$$\mathcal{K} \cup \neg q \text{ is satisfiable, i.e., there is a model } \mathcal{I} \text{ of } \mathcal{K} \text{ such that } \mathcal{I} \models \neg q. \tag{2}$$

It is not hard to see that there is a correspondence between negated CQs$^{\neq}$ and so-called disjunctive EGDs.

We remind the reader that an *equality-generating dependency* (*EGD*) [5] is a formula of the form $\forall \boldsymbol{x} \big(\phi(\boldsymbol{x}) \rightarrow (x_1 = x_2)\big)$, where x_1, x_2 are among the variables in $\boldsymbol{x}$. A *disjunctive EGD* [12] is a formula of the form

$$\forall \boldsymbol{x} \big(\phi(\boldsymbol{x}) \rightarrow \bigvee_{i=1}^{n} (x_i^1 = x_i^2)\big). \tag{3}$$

Note that an EGD is a disjunctive EGD whose right-hand side has only one equality.

Given a Boolean CQ$^{\neq}$ $q = \exists \boldsymbol{x} \, (\phi(\boldsymbol{x}) \wedge \bigwedge_i (x_i^1 \neq x_i^2))$, where $\phi(\boldsymbol{x})$ is a conjunction of concept and role atoms, it should be clear that $\neg q$ is logically equivalent to a disjunctive EGD of the form (3). Disjucntive EGDs are clearly able to express concept inclusion axioms with arbitrary number restrictions (in particular, functionality of roles), which is known to increase the complexity of reasoning in *DL-Lite* [7].

The previous discussion suggests the following algorithm for checking condition (2): non-deterministically guess a model $\mathcal{J}$ of $\mathcal{K}$ and then check in polynomial time whether $\mathcal{J}$ satisfies $\neg q$. In order to obtain the coNP result, $\mathcal{J}$ needs not only to be finite but also small enough—at most polynomial in the size of the ABox of $\mathcal{K}$. Unfortunately, this straightforward approach is too naive. Indeed, since disjunctive EGDs allow to express *global functionality* of roles, and the extension of *DL-Lite*$^{\mathcal{H}}_{core}$ with functional

roles does not enjoy the finite model property then we cannot guess a finite model $\mathcal{J}$ and check whether $\mathcal{J} \models \neg q$, as shown by the following example.

Example 2. Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ with $\mathcal{T} = \{\exists P^- \sqsubseteq \exists P,\ A \sqsubseteq \neg\exists P^-,\ A \sqsubseteq \exists P\}$, $\mathcal{A} = \{A(a)\}$ and $q = \exists x, y_1, y_2 \left(P(y_1, x) \land P(y_2, x) \land (y_1 \neq y_2)\right)$, which 'says' that P^- is functional. The canonical model $\mathcal{U}_\mathcal{K}$ is an infinite P-chain starting at a, whence $\mathcal{U}_\mathcal{K} \models \neg q$. However, there is no finite model of $\mathcal{K}$ satisfying $\neg q$. In fact, for every model $\mathcal{J}$ of $\mathcal{K}$ with $\mathcal{J} \models \neg q$ there is an *injective* homomorphism from $\mathcal{U}_\mathcal{K}$ to $\mathcal{J}$.

In order to have an effective algorithm we need then to find a way to simulate the possibly infinite model of a KB $\mathcal{K}$ in a small finite initial fragment of the canonical model $\mathcal{U}_\mathcal{K}$ of $\mathcal{K}$. We start by recalling that for answering CQs only a linear number (in the size of the TBox of $\mathcal{K}$) of existential witnesses are need to be considered [4,16]. Next, we show that for answering CQs$^{\neq}$ in *DL-Lite*$_{core}^{\mathcal{H}}$ it is also enough to consider only a linear number of existential witnesses for falsifying the inequalities in q. The main difference is that for answering CQs$^{\neq}$ we need to try all possible configurations of identifying objects in the initial fragment of the model, and hence the increase in complexity.

Let us fix a *DL-Lite*$_{core}^{\mathcal{H}}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and a CQ$^{\neq}$ q for the rest of this section. An *expansion* $\mathcal{A}'$ *of* $\mathcal{A}$ is a (possibly infinite) set of assertions (in the signature of $\mathcal{T}$) that contains $\mathcal{A}$ and whose individuals are taken from the domain of the canonical model $\mathcal{U}_\mathcal{K}$ of $\mathcal{K}$. In other words, an expansion is a description of a part of the canonical model $\mathcal{U}_\mathcal{K}$. Consider now a disjunctive EGD of the form (3) which is equivalent to $\neg q$. We associate with it the following set $\mathbf{E}$ of individual EGDs:

$$\mathbf{E} \ = \ \{\ \underbrace{\forall \boldsymbol{x}\left(\phi(\boldsymbol{x}) \rightarrow (x_1^1 = x_1^2)\right)}_{e_1},\ \ldots,\ \underbrace{\forall \boldsymbol{x}\left(\phi(\boldsymbol{x}) \rightarrow (x_n^1 = x_n^2)\right)}_{e_n}\ \}.$$

Definition 2. *Let $\mathcal{A}'$ be an ABox expansion of $\mathcal{A}$ and h a homomorphism from $\phi(\boldsymbol{x})$ to $\mathcal{A}'$. We say that e_i is* applicable to $\mathcal{A}'$ with h *if $h(x_i^1) \neq h(x_i^2)$ and the result of applying e_i to $\mathcal{A}'$ with h is one of the following*:

(fail) *a* failure*, in which case we write $\mathcal{A}' \xrightarrow{h,e_i} \bot$, if either*
1. $h(x_i^1), h(x_i^2) \in ind(\mathcal{A})$*, or*
2. $B_1(h(x_i^1)), B_2(h(x_i^2)) \in \mathcal{A}'$*, for some $\mathcal{T} \models B_1 \sqcap B_2 \sqsubseteq \bot$, or*
3. $R_1(a, h(x_i^1)), R_2(a, h(x_i^2)) \in \mathcal{A}'$*, for $a \in ind(\mathcal{A}')$ and $\mathcal{T} \models R_1 \sqsubseteq \neg R_2$*;

(id) *an ABox expansion $\mathcal{A}''$ (written $\mathcal{A}' \xrightarrow{h,e_i} \mathcal{A}''$) obtained by identifying $h(x_i^1)$ and $h(x_i^2)$: every occurrence of $h(x_i^1)$ and $h(x_i^2)$ is replaced by $h(x_i^k)$, if $h(x_i^k) \in ind(\mathcal{A})$ for $k = 1$ or 2, and by $h(x_i^1)$, otherwise (the choice of x_i^1 here is arbitrary as neither of them is in the ABox).*

We say that $\mathbf{E}$ is *applicable to $\mathcal{A}'$ with h* if e_i is applicable to $\mathcal{A}'$ with h for every $1 \leq i \leq n$. The result of applying $\mathbf{E}$ to $\mathcal{A}'$ with h is the set $\{\mathcal{A}'_1, \ldots, \mathcal{A}'_n\}$, where each $\mathcal{A}'_i$ is the result of applying e_i to $\mathcal{A}'$ with h ($\mathcal{A}'_i$ may be $\bot$); we write $\mathcal{A}' \xrightarrow{h,\mathbf{E}} \{\mathcal{A}'_1, \ldots \mathcal{A}'_n\}$ in this case. If every $\mathcal{A}'_i = \bot$ we say that the application of $\mathbf{E}$ to $\mathcal{A}$ with h *fails*, and write $\mathcal{A}' \xrightarrow{h,\mathbf{E}} \bot$; otherwise we say it is *non-failing*.

We first show some technical results on disjunctive EGD applications. The following is an easy consequence of the definition of (non-failing) application of $\mathbf{E}$ to an ABox expansion $\mathcal{A}'$:

Proposition 1. *Let $\mathcal{A}'$ be a finite ABox expansion of $\mathcal{A}$ and $\mathcal{A}' \xrightarrow{h,\mathbf{E}} \{\mathcal{A}'_1, \dots \mathcal{A}'_n\}$ a non-failing application of $\mathbf{E}$ to $\mathcal{A}'$. Then $dom(\mathcal{A}') \supseteq dom(\mathcal{A}'_i)$, for all $1 \leq i \leq n$ with $\mathcal{A}'_i \neq \bot$.*

In fact, given any model $\mathcal{J}$ of $\mathcal{K}$, every homomorphism from an ABox expansion to $\mathcal{J}$ can be extended to a homomorphism from a non-failng application of $\mathbf{E}$:

Lemma 3. *Let $\mathcal{A}'$ be a finite ABox expansion of $\mathcal{A}$ and $\mathcal{A}' \xrightarrow{h,\mathbf{E}} \{\mathcal{A}'_1, \dots \mathcal{A}'_n\}$ a non-failing application of $\mathbf{E}$ to $\mathcal{A}'$ with h and $\mathcal{J}$ a model of $\mathcal{K}$ such that $\mathcal{J} \models \neg q$ and there is a homomorphism g from $\mathcal{A}'$ into $\mathcal{J}$. Then there exists a homomorphism g_j from $\mathcal{A}'_j$ into $\mathcal{J}$ for some $1 \leq j \leq n$.*

Now we consider *non-failing sequence of application of* $\mathbf{E}$, which are sequence of the form $\mathcal{A}' \xrightarrow{h_1,e_{i_1}} \mathcal{A}'_1 \xrightarrow{h_2,e_{i_2}} \dots \xrightarrow{h_k,e_{i_k}} \mathcal{A}'_k$ with $1 \leq i_j \leq n$ and $\mathcal{A}'_j \neq \bot$, for every $1 \leq i \leq k$. It turns out that after applying a non-failing application of EGD in some part of the ABox expansion the successive applications of the disjunctive EGD do not "use the same match", and therefore, the process will eventually either succeed on the application or fail:

Proposition 2. *For every non-failing sequence $\mathcal{A}' \xrightarrow{h_1,e_{i_1}} \mathcal{A}_1 \xrightarrow{h_2,e_{i_2}} \dots \xrightarrow{h_k,e_{i_k}} \mathcal{A}_k$ of applications of $\mathbf{E}$, we have $h_j(x) \neq h_{j'}(x)$, for all $1 \leq j < j' \leq k$ and some $x \in \boldsymbol{x}$.*

Next, we show that for checking whether there is a model $\mathcal{I}$ of $\mathcal{K}$ with $\mathcal{I} \models \neg q$ it suffices to apply $\mathbf{E}$ to an ABox expansion $\widehat{\mathcal{A}}$ that corresponds to the canonical model 'truncated' to points of depth up to $N = |role^{\pm}(\mathcal{K})| + |q|$. More formally, given a natural number N, the *truncation* $\mathcal{U}^N_{\mathcal{K}}$ of the canonical model $\mathcal{U}_{\mathcal{K}}$ to depth N is the restriction of $\mathcal{U}_{\mathcal{K}}$ to the following domain:

$$\Delta^{\mathcal{U}^N_{\mathcal{K}}} = \{\sigma \in \Delta^{\mathcal{U}_{\mathcal{K}}} \mid \|\sigma\| \leq N\},$$

where $\|\sigma\|$ is the length of a path σ. By Proposition 2, there is a bound on the number of possible applications of $\mathbf{E}$ to $\widehat{\mathcal{A}}$. More precisely, the length of every application sequence of $\mathbf{E}$ to $\widehat{\mathcal{A}}$ is bounded by a polynomial in the size of $\mathcal{A}$.

Finally, we show the following:

Lemma 4. *Let $\widehat{\mathcal{A}}$ the ABox expansion of $\mathcal{A}$ induced the truncation $\mathcal{U}^N_{\mathcal{K}}$ of the canonical model $\mathcal{U}_{\mathcal{K}}$ of $\mathcal{K}$ to depth $N = |role^{\pm}(\mathcal{K})|+|q|$. The following statements are equivalent*:

1. *there exists a model $\mathcal{J}$ of $\mathcal{K}$ such that $\mathcal{J} \models \neg q$;*
2. *there is a sequence $e_1, \dots, e_k$ of elements of $\mathbf{E}$ such that $\widehat{\mathcal{A}} \xrightarrow{h_1,e_1} \mathcal{A}_1 \xrightarrow{h_2,e_2} \dots \xrightarrow{h_k,e_k} \mathcal{A}_k$ is non-failing and $\mathcal{A}_k$ satisfies $\neg q$.*

Now, given a *DL-Lite*$^{\mathcal{H}}_{core}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and a CQ$^{\neq}$ q, our algorithm for checking condition (2) works as follows:

1. It constructs the ABox expansion $\widehat{\mathcal{A}}$ of $\mathcal{A}$ induced by $\mathcal{U}^N_{\mathcal{K}}$, for $N = |role^{\pm}(\mathcal{K})|+|q|$.

2. Guesses a sequence Σ of elements of $\mathbf{E}$.
3. Checks whether $\mathbf{E}$ is satisfied after the application of Σ to $\widehat{\mathcal{A}}$.

It is not hard to see that this non-deterministic algorithm runs in polynomial time in the size of the ABox. So, by (2) and Lemma 4, we obtain a matching upper bound for Theorem 3, which results in the following:

Theorem 4. *Answering CQs$^{\neq}$ over DL-Lite$^{\mathcal{H}}_{core}$ KBs is* coNP*-complete in data complexity.*

5 Tractable Cases

In this section, we define syntactic restrictions on the class of CQs$^{\neq}$ in order to achieve tractability of CQ$^{\neq}$ answering in *DL-Lite$^{\mathcal{H}}_{core}$*. In the data exchange setting (DE) it has been shown that answering CQs$^{\neq}$ with *at most* two inequalities in the presence of target constraints expressed by *weakly acyclic* TGDs is coNP-complete in data complexity [17]. In the case of DLs, we note that even very simple *DL-Lite$^{\mathcal{H}}_{core}$* TBoxes are not weakly acyclic. On the other hand, the reductions used for proving coNP-hardness of the CQ$^{\neq}$ answering problem in the DE setting make use of ternary relations, which is outside the expressive power of *DL-Lite$^{\mathcal{H}}_{core}$*. Up to this point, we can only conjecture that answering CQs$^{\neq}$ in *DL-Lite$^{\mathcal{H}}_{core}$* that contain at least two inequalities is coNP-hard. Therefore, we based our syntactic restrictions on the latter assumption.

We shall consider CQs with at most two inequalities. Roughly, in order to have a polynomial algorithm in data complexity for checking that $\mathcal{K} \models q$ or alternatively that $\mathcal{K} \cup \neg q$ is unsatisfiable, we need to be able 1) to simulate the infinite chase in a finite search space and 2) perform the evaluation using a small amount of space (e.g., constant in the size of the ABox). In order to have a correct and complete algorithm fulfilling these conditions, we impose syntactic restrictions enforcing that, for every possible match π for a query q in a given interpretation $\mathcal{I}$ and for every inequality $x_i^1 \neq x_i^2$ in q, either $\pi(x_i^1) = a$ or $\pi(x_i^2) = a$, for some individual name a. This condition is enough to ensure polynomial-time query evaluation because, although this kind of inequalities are not preserved under homomorphisms, they induce only few possible models.

We adopt and adapt the notions of *constant joins* and *almost constant inequalities* introduced by Arenas *et al.* [3]. For defining these notions in the DL setting, it is convenient to identify the concepts that may need to be 'realised outside' the ABox in every model of a KB.

Definition 3. *Let $\mathcal{T}$ be a DL-Lite$^{\mathcal{H}}_{core}$ TBox. A concept $\exists R$ is called* affected in $\mathcal{T}$ *if either*

1. $\mathcal{T} \models A \sqsubseteq \exists R^-$*, for some concept name A, or*
2. $\mathcal{T} \models \exists S \sqsubseteq \exists R^-$*, for some role S with $\mathcal{T} \not\models S \sqsubseteq R^-$.*

We say an inequality $(x_1 \neq x_2)$ in q is *almost constant for $\mathcal{T}$* if q contains either some $R(t, x_1)$ or some $R(t, x_2)$ such that $\exists R^-$ is not affected in $\mathcal{T}$. Intuitively, queries with almost constant inequalities ensure that at least one variable in each inequality is forced to be an ABox individual. A query q is said to have *constant joins for $\mathcal{T}$* if either $\exists R_1^-$ or $\exists R_2^-$ is not affected in $\mathcal{T}$, for every join $R_1(t_1, t), R_2(t_2, t)$ in q. This means

that t has to be mapped to an ABox individual by every possible match for q in any model of the KB.

Definition 4. *A CQ$^{\neq}$ q is said to be* safe *if one the following conditions holds*:

1. *q has no inequalities,*
2. *q has exactly one inequality, which is almost constant,*
3. *q has exactly two inequalities, which are almost constant, and constant joins.*

Intuitively, to falsify the inequalities in a safe CQ$^{\neq}$ it suffices to consider only inequalities of the form $a_1 \neq d$ and $a_1 \neq a_2$, where $a_1, a_2 \in ind(\mathcal{A})$ and d is an anonymous individual in the canonical model, i.e., a path in $\Delta^{\mathcal{U}_\mathcal{K}}$ of the form $\sigma \cdot d_R$. This means, that q can actually be evaluated in the ABox expansion corresponding to the generating interpretation $\mathcal{I}_\mathcal{K}$.

Given a safe CQ$^{\neq}$, we need to provide an algorithm for deciding whether $\mathcal{K} \models q$. The algorithm presented in Section 4 considers a truncation of the canonical model $\mathcal{U}_\mathcal{K}$ of $\mathcal{K}$ for evaluating $\neg q$. However, in this case—as we argued above—by the syntactic restrictions on q, the evaluation requires only to consider the generating interpretation $\mathcal{I}_\mathcal{K}$ as in the case for queries without inequalities and suggests that we can adapt the combined approach for query answering [16] by making minor changes to the rewriting of q. Thus, we obtain the following result:

Theorem 5. *Answering safe CQ$^{\neq}$ over DL-Lite$_{core}^{\mathcal{H}}$ KBs is in AC^0 in data complexity.*

6 Conclusions

The known and obtained complexity results on answering CQs and UCQs with safe negation and inequalities are presented in the table below:

	CQ$^{\neq}$	UCQ$^{\neq}$	CQ$^{\neg s}$	UCQ$^{\neg s}$
DL-Lite$_{core}$	coNP Thms. 3, 4	undec. Thm. 1	coNP-hard [19, Thm. 13]	undec. Thm. 2
DL-Lite$_{core}^{\mathcal{H}}$	coNP [19, Thm. 6] Thm. 4	undec. [19, Thm. 8]	coNP-hard [19, Thm. 13]	undec. [19, Thm. 15]

We have presented some further steps towards a systematic study of query answering in DLs when extensions of CQs with negated atoms are considered. In particular, we build on previous work by Rosati [19], and extend this investigation by adapting techniques from such areas as data exchange to identify tractable cases of CQ$^{\neq}$ answering in logics from the *DL-Lite* family. Clearly, more investigations needs to be done to construct a complete picture of the computational complexity and to develop algorithmic approaches. We outline below some research questions we will address in the future:

1. Investigating query answering with inequalities in other logics of *DL-Lite* family and $\mathcal{EL}$ family.
2. Closing the gap on the number of inequalities needed to make CQ$^{\neq}$ answering intractable in DLs of the *DL-Lite* family.

3. An interesting and challenging problem is the development of a decision procedures for answering CQs$^{\neg s}$. We also plan to consider other types of negation such as Boolean combinations of CQs (BCCQs) advocated in areas related to the management of incomplete information [10,13].

References

1. Abiteboul, S., Duschka, O.M.: Complexity of answering queries using materialized views. In: Proc. of PODS. pp. 254–263. ACM Press (1998)
2. Abiteboul, S., Duschka, O.M.: Complexity of answering queries using materialized views. Tech. Rep. Gemo Report 383, INRIA Saclay (1999)
3. Arenas, M., Barceló, P., Reutter, J.L.: Query languages for data exchange: Beyond unions of conjunctive queries. Theory Comput. Syst. 49(2), 489–564 (2011)
4. Artale, A., Calvanese, D., Kontchakov, R., Zakharyaschev, M.: The DL-Lite family and relations. J. Artif. Intell. Res. (JAIR) 36, 1–69 (2009)
5. Beeri, C., Vardi, M.Y.: A proof procedure for data dependencies. J. ACM 31(4), 718–741 (1984)
6. Bienvenu, M., Ortiz, M., Simkus, M.: Answering expressive path queries over lightweight DL knowledge bases. In: Proc. of DL. CEUR Workshop Proceedings, vol. 846 (2012)
7. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Data complexity of query answering in description logics. In: Doherty, P., Mylopoulos, J., Welty, C.A. (eds.) Proc. of KR. pp. 260–270. AAAI Press (2006)
8. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. J. Autom. Reasoning 39(3), 385–429 (2007)
9. Calvanese, D., De Giacomo, G., Lenzerini, M.: On the decidability of query containment under constraints. In: Mendelzon, A.O., Paredaens, J. (eds.) Proc. of PODS. pp. 149–158. ACM Press (1998)
10. ten Cate, B., Chiticariu, L., Kolaitis, P.G., Tan, W.C.: Laconic schema mappings: Computing the core with SQL queries. PVLDB 2(1), 1006–1017 (2009)
11. Deutsch, A., Nash, A., Remmel, J.B.: The chase revisited. In: Lenzerini, M., Lembo, D. (eds.) Proc. of PODS. pp. 149–158. ACM Press (2008)
12. Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L.: Data exchange: semantics and query answering. Theor. Comput. Sci. 336(1), 89–124 (2005)
13. Gheerbrant, A., Libkin, L., Tan, T.: On the complexity of query answering over incomplete XML documents. In: Deutsch, A. (ed.) Proc. of ICDT. pp. 169–181. ACM (2012)
14. Harel, D.: Effective transformations on infinite trees, with applications to high undecidability, dominoes, and fairness. J. ACM 33(1), 224–248 (1986)
15. Klug, A.: On conjunctive queries containing inequalities. J. ACM 35(1), 146–160 (1988)
16. Kontchakov, R., Lutz, C., Toman, D., Wolter, F., Zakharyaschev, M.: The combined approach to query answering in DL-Lite. In: Lin, F., Sattler, U., Truszczynski, M. (eds.) Proc. of KR. AAAI Press (2010)
17. Madry, A.: Data exchange: On the complexity of answering queries with inequalities. Inf. Process. Lett. 94(6), 253–257 (2005)
18. Rosati, R.: On the decidability and finite controllability of query processing in databases with incomplete information. In: Vansummeren, S. (ed.) Proc. of PODS. pp. 356–365. ACM (2006)
19. Rosati, R.: The limits of querying ontologies. In: Schwentick, T., Suciu, D. (eds.) Proc. of ICDT. LNCS, vol. 4353, pp. 164–178. Springer (2007)