Conjunctive Queries with Negation over DL-Lite: A Closer Look

Víctor Gutiérrez-Basulto[1], Yazmín Ibañez-García[2], Roman Kontchakov[3], and Egor V. Kostylev[4]

[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] Dept. of Computer Science and Inf. Systems, Birkbeck, University of London, UK
roman@dcs.bbk.ac.uk
[4] School of Informatics, University of Edinburgh, UK
ekostyle@inf.ed.ac.uk

Abstract. While conjunctive query (CQ) answering over *DL-Lite* has been studied extensively, there have been few attempts to analyse CQs with negated atoms. This paper deepens the study of the problem. Answering CQs with safe negation and CQs with a single inequality over *DL-Lite* with role inclusions is shown to be undecidable, even for a fixed TBox and query. Without role inclusions, answering CQs with one inequality is P-hard and with two inequalities coNP-hard in data complexity.

1 Introduction

The *ontology-based data access* (OBDA) paradigm of enriching instance data with background knowledge, provided by means of a description logic (DL) ontology, has become one of the most prominent approaches to management of incomplete data on the Web. In the past decade, a vast investigation of answering (unions of) conjunctive queries in the OBDA paradigm has been conducted, so that now a fairly clear landscape of the computational complexity has emerged and a number of algorithmic approaches have been presented and implemented in OBDA systems. Notably, special effort has been invested into developing DL languages that, on the one hand, are expressive enough to capture interesting aspects of the application domain and, on the other hand, allow OBDA systems to scale to large amounts of data, in particular, by delegating query evaluation to relational database management systems. Among the different proposed DLs fulfilling these requirements we find members of the *DL-Lite* family [1, 2], which form the basis of OWL 2 QL, one of the three profiles of the Web Ontology Language OWL 2, and where answering (unions of) CQs is in AC^0 in data complexity.

In recent years, the problem of answering more expressive queries over *DL-Lite* has been investigated [3–6]; in particular, following the tradition of relational databases, Rosati [3] and Gutiérrez-Basulto *et al.* [5] investigated extensions of CQs with two restricted forms of negated atoms: (1) *inequality* ($CQ^{\neq}$) and (2) *safe negation* ($CQ^{\neg s}$). It is well-known in databases and other areas related to management of incomplete data that answering CQs with these types of negation becomes harder than answering (positive) CQs. Rosati [3] and Gutiérrez-Basulto *et al.* [5] showed that this is even worse in the OBDA paradigm: the problems of answering *unions* of CQs$^{\neq}$ and *unions* of CQs$^{\neg s}$ were shown to be undecidable over the simplest language of *DL-Lite*$_{\text{core}}$ (in striking contrast to the highly tractable AC^0 upper bound for data complexity in case of unions of CQs).

Finding decision algorithms and analysing complexity of answering CQs$^{\neq}$ and CQs$^{\neg s}$ over *DL-Lite*$_{\text{core}}$ and its extension with role inclusions, *DL-Lite*$^{\mathcal{H}}_{\text{core}}$, has proven remarkably challenging. First, the weak expressive power of these ontology languages makes it difficult to show undecidability using encodings similar to those for *unions* of CQs$^{\neq}$ and CQs$^{\neg s}$. Second, in contrast to positive atoms of CQs, the negated atoms are not preserved under homomorphisms [7], hence query answering techniques based on the canonical model construction [1, 8] cannot be directly applied. In fact, up to now, the only known result is coNP-hardness for CQs$^{\neq}$ and CQs$^{\neg s}$ over *DL-Lite*$_{\text{core}}$ [3, 5]; Gutiérrez-Basulto *et al.* [5] claimed a matching upper bound for CQ$^{\neq}$ answering over *DL-Lite*$^{\mathcal{H}}_{\text{core}}$, alas, the presented algorithm is incorrect.

The purpose of this paper is to sharpen the panorama of answering CQs extended with inequalities and safe negation over *DL-Lite*$_{\text{core}}$ and *DL-Lite*$^{\mathcal{H}}_{\text{core}}$. In Section 2, we define the two DLs and conjunctive queries with negated atoms. In the first part of the paper, we study the problem of answering CQs$^{\neq}$ and CQs$^{\neg s}$ over *DL-Lite*$^{\mathcal{H}}_{\text{core}}$. In Section 3, we provide a general reduction of answering unions of (acyclic) CQs to answering single CQs over ontologies with role inclusions; this, in particular, results in undecidability of answering CQs$^{\neg s}$ over *DL-Lite*$^{\mathcal{H}}_{\text{core}}$. In Section 4, instead of using the method of Section 3 to obtain undecidability of answering CQs$^{\neq}$, we provide a more elaborate proof of the result for a CQ$^{\neq}$ with a *single* inequality (a proof along the lines of Section 3 would require many inequalities). We mention in passing that CQ$^{\neq}$ answering over light-weight description logic $\mathcal{EL}$ is also undecidable [9]; however, CQ answering in $\mathcal{EL}$ is P-complete rather than in AC^0 (in data complexity).

In the second part of the paper, we consider the problem of answering CQs$^{\neq}$ over *DL-Lite*$_{\text{core}}$, the language without role inclusions. While decidability is still an open problem, we analyse how far the coNP-hardness of this problem can be pushed down by restricting the *number* of inequalities in a query. In Section 5, we sharpen the lower bounds for data complexity: P-hardness with *one* inequality and coNP-hardness with two inequalities.

2 Preliminaries

The language of *DL-Lite*$^{\mathcal{H}}_{\text{core}}$ (and *DL-Lite*$_{\text{core}}$) [2] contains *individual names* $c_1, c_2, \ldots$, *concept names* $A_1, A_2, \ldots$, and *role names* $P_1, P_2, \ldots$. *Roles* R and *basic concepts* B are defined by the following grammar:

$$R \ ::= \ P_i \quad | \quad P_i^-, \qquad\qquad B \ ::= \ \bot \quad | \quad A_i \quad | \quad \exists R.$$

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

$$B_1 \sqsubseteq B_2, \qquad B_1 \sqcap B_2 \sqsubseteq \bot, \qquad R_1 \sqsubseteq R_2, \qquad R_1 \sqcap R_2 \sqsubseteq \bot.$$

A *DL-Lite*$_{core}$ TBox contains only concept inclusions. We will use conjunction on the right-hand side and disjunction on the left-hand side of inclusions (which is syntactic sugar). An *ABox* $\mathcal{A}$ is a finite set of *assertions* of the form $A_i(c_j)$ and $P_i(c_j, c_k)$. A *knowledge base* (*KB*) $\mathcal{K}$ is a pair $(\mathcal{T}, \mathcal{A})$, where $\mathcal{T}$ is a TBox and $\mathcal{A}$ an ABox.

An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is a nonempty domain $\Delta^{\mathcal{I}}$ with an interpretation function $\cdot^{\mathcal{I}}$ that assigns an element $c_i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ to each individual name c_i, a subset $A_i^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ to each concept name A_i, and a binary relation $P_i^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ to each role name P_i. As usual for *DL-Lite*, we adopt the *unique name assumption* (UNA): $c_i^{\mathcal{I}} \neq c_j^{\mathcal{I}}$, for all distinct individuals c_i, c_j. Our results, however, do not depend on UNA. The interpretation function $\cdot^{\mathcal{I}}$ is extended to roles and basic concepts in the standard way:

$$
\begin{aligned}
(P_i^-)^{\mathcal{I}} &= \{(d', d) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (d, d') \in P_i^{\mathcal{I}}\}, && \text{(inverse role)}\\
\bot^{\mathcal{I}} &= \emptyset, && \text{(empty set)}\\
(\exists R)^{\mathcal{I}} &= \big\{d \in \Delta^{\mathcal{I}} \mid \text{there is } d' \in \Delta^{\mathcal{I}} \text{ with } (d, d') \in R^{\mathcal{I}}\big\}. && \text{(role domain/range)}
\end{aligned}
$$

The *satisfaction relation* $\models$ is also standard:

$$
\begin{array}{rlcrl}
\mathcal{I} \models B_1 \sqsubseteq B_2 & \text{iff} \quad B_1^{\mathcal{I}} \subseteq B_2^{\mathcal{I}}, & & \mathcal{I} \models B_1 \sqcap B_2 \sqsubseteq \bot & \text{iff} \quad B_1^{\mathcal{I}} \cap B_2^{\mathcal{I}} = \emptyset,\\
\mathcal{I} \models R_1 \sqsubseteq R_2 & \text{iff} \quad R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}}, & & \mathcal{I} \models R_1 \sqcap R_2 \sqsubseteq \bot & \text{iff} \quad R_1^{\mathcal{I}} \cap R_2^{\mathcal{I}} = \emptyset,\\
\mathcal{I} \models A_i(c_j) & \text{iff} \quad c_j^{\mathcal{I}} \in A_i^{\mathcal{I}}, & & \mathcal{I} \models P_i(c_j, c_k) & \text{iff} \quad (c_j^{\mathcal{I}}, c_k^{\mathcal{I}}) \in P_i^{\mathcal{I}}.
\end{array}
$$

A KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is *satisfiable* if there is an interpretation $\mathcal{I}$ satisfying all inclusions of $\mathcal{T}$ and assertions of $\mathcal{A}$. 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 $\mathcal{A}$).

A *conjunctive query* (CQ) $\boldsymbol{q}(\boldsymbol{x})$ is a formula of the form $\exists \boldsymbol{y}\, \varphi(\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{x}$ and $\boldsymbol{y}$ are tuples of variables and φ is a conjunction of concept atoms $A_i(t)$ and role atoms $P_i(t, t')$ with t and 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. A *conjunctive query with safe negation* (CQ$^{\neg s}$) is an expression of the form $\exists \boldsymbol{y}\, \varphi(\boldsymbol{x}, \boldsymbol{y})$, where φ is a conjunction of *literals*, that is, (positive) atoms and negated atoms, such that each variable occurs in at least one positive atom. A *conjunctive query with inequalities* (CQ$^{\neq}$) is an expression of the form $\exists \boldsymbol{y}\, \varphi(\boldsymbol{x}, \boldsymbol{y})$, where each conjunct of φ is a positive atom or an inequality $t \neq t'$, for terms t and t'. A *union of conjunctive queries* (UCQ) is a disjunction of CQs; UCQ$^{\neg s}$ and UCQ$^{\neq}$ are defined accordingly. We assume a CQ contains $P^-(z_1, z_2)$ if it contains $P(z_2, z_1)$. We write $\boldsymbol{q}$ instead of $\boldsymbol{q}(\boldsymbol{x})$ if $\boldsymbol{x}$ is clear from the context or empty—in the latter case the query is called *Boolean*.

Let $\boldsymbol{q}(\boldsymbol{x}) = \exists \boldsymbol{y}\, \varphi(\boldsymbol{x}, \boldsymbol{y})$ be a query with $\boldsymbol{x} = (x_1, \dots, x_k)$, $\mathcal{I}$ an interpretation and π a map from the set of terms of $\boldsymbol{q}$ to $\Delta^{\mathcal{I}}$ with $\pi(c) = c^{\mathcal{I}}$, for all individual names c in $\boldsymbol{q}$. We call π a *match for* $\boldsymbol{q}$ *in* $\mathcal{I}$ if $\mathcal{I}$ (as a first-order model) satisfies φ under the variable assignment mapping each variable z of φ to $\pi(z)$. A k-tuple of individual names $\boldsymbol{c} = (c_1, \dots, c_k)$ is an *answer to* $\boldsymbol{q}$ *in* $\mathcal{I}$ if there is a match for $\boldsymbol{q}$ in $\mathcal{I}$ with $\pi(x_i) = c_i^{\mathcal{I}}$ (such a π is called a match for $\boldsymbol{q}(\boldsymbol{c})$ in $\mathcal{I}$). We say that $\boldsymbol{c}$ is a *certain answer* to $\boldsymbol{q}$ over a KB $\mathcal{K}$ and write $\mathcal{K} \models \boldsymbol{q}(\boldsymbol{c})$ if $\boldsymbol{c}$ is an answer to $\boldsymbol{q}$ in all models of $\mathcal{K}$.

In OBDA scenarios the size of the query and the TBox is much smaller than the size of the ABox. This is why we explore the *data complexity* [10] of the query answering problem, that is, we assume that only the ABox is considered as part of the input. Formally, let $\mathcal{T}$ be a *DL-Lite*$^{\mathcal{H}}_{core}$ or *DL-Lite*$_{core}$ TBox and $\boldsymbol{q}(\boldsymbol{x})$ a (U)CQ$^{\neg s}$ or a (U)CQ$^{\neq}$. We are interested in the following family of problems.

CERTAIN ANSWERS $(\boldsymbol{q}, \mathcal{T})$	*Input*: *Question*:	An ABox $\mathcal{A}$ and a tuple of individuals $\boldsymbol{c}$. Is $\boldsymbol{c}$ a certain answer to $\boldsymbol{q}$ over $(\mathcal{T}, \mathcal{A})$?

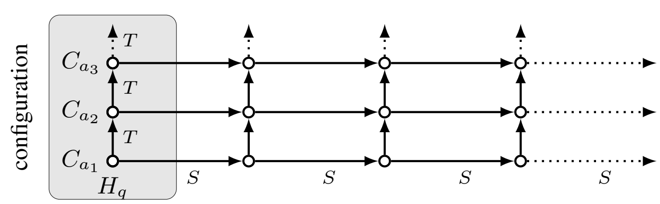
3 Answering CQs with Safe Negation is Undecidable

It is known [3] that computing certain answers to a *union* of CQs with safe negation (UCQ$^{\neg s}$) over *DL-Lite*$_{core}$ is undecidable if the TBox and the query are part of the problem input (which corresponds to the combined complexity [10]). We first show that the problem remains undecidable even if the TBox and the query are fixed. Then we proceed to show how the UCQ$^{\neg s}$ can be transformed into a single CQ$^{\neg s}$, thus obtaining the main result of this section. We note that the transformation is rather general (and also works with inequalities) and may be of general interest.

Theorem 1. *There is a Boolean UCQ*$^{\neg s}$ $\boldsymbol{q}$ *and a DL-Lite*$_{core}$ *TBox* $\mathcal{T}$ *such that the problem* CERTAIN ANSWERS $(\boldsymbol{q}, \mathcal{T})$ *is undecidable.*

Proof. The proof is by reduction of the halting problem for deterministic Turing machines. In particular, given a Turing machine M, we construct a TBox $\mathcal{T}$ and a query $\boldsymbol{q}$ such that M does not accept an input $\boldsymbol{w}$ encoded as an ABox $\mathcal{A}_{\boldsymbol{w}}$ iff $(\mathcal{T}, \mathcal{A}_{\boldsymbol{w}}) \not\models \boldsymbol{q}$ ($\mathcal{T}$ and $\boldsymbol{q}$ depend on M but not on $\boldsymbol{w}$). Applying this construction to a fixed deterministic *universal* Turing machine, i.e., a machine that accepts its input $\boldsymbol{w}$ iff the Turing machine encoded by $\boldsymbol{w}$ accepts the empty input, we obtain the required undecidability result.

Let $M = (\Gamma, Q, q_0, q_1, \delta)$ be a deterministic Turing machine, where Γ is an alphabet (containing the blank symbol $\llcorner\!\lrcorner$), Q a set of states, $q_0 \in Q$ and $q_1 \in Q$ are an initial and an accepting state, respectively, and $\delta\colon Q \times \Gamma \to Q \times \Gamma \times \{-1, 1\}$ is a transition function. Computations of M can be thought of as sequences of configurations, with each configuration determined by the contents of all (infinitely many) cells of the tape, the state and the head position. We are going to encode a computation by domain elements arranged, roughly speaking, into a two-dimensional grid: one dimension is the tape and the other is time (see the picture below, where the nodes are domain elements and the grey rectangle illustrates an initial configuration, in which the tape contains $a_1 a_2 a_3 \ldots$ and the head is positioned over the first cell in state q).



More precisely, we use a role T to point to the representation of the next cell on the tape (within the same configuration) and a role S to point to the representation of the same cell in a successive configuration. Concepts C_a, for $a \in \Gamma$, encode the contents of cells in the sense that a domain element belongs to C_a if the respective cell contains the symbol a. We use concepts H_q, for $q \in Q$, to indicate both the position of the head and the current state: a domain element belongs to H_q if the respective cell is under the head and the machine is in state q. We also use a concept $H_\emptyset$ to mark all other cells on the tape (that is, cells that are not under the head of the machine). Finally, roles P_{qa}, for $q \in Q$ and $a \in \Gamma$, are used to encode transitions; concepts D_σ and roles $T_{\emptyset\sigma}$, for $\sigma \in \{-1, +1\}$, to propagate the no-head marker backwards and forwards along the tape; and role T_0 to make sure the tape is initially blank beyond the input word.

Consider a Boolean UCQ$^{\neg s}$ $\boldsymbol{q}$ that is a union of the existential closures of the negations of the following first-order formulas:

$$S(x,y) \land T(x,z) \land S(z,u) \to T(y,u), \tag{1}$$

$$H_q(x) \land C_a(x) \land S(x,y) \land T^\sigma(y,z) \to P_{q'a'}(y,z), \quad \text{for } \delta(q,a) = (q', a', \sigma), \tag{2}$$

$$H_\emptyset(x) \land C_a(x) \land S(x,y) \to C_a(y), \qquad \text{for } a \in \Gamma, \tag{3}$$

$$D_\sigma(y) \land T^\sigma(y,z) \to T_{\emptyset\sigma}(y,z), \qquad \text{for } \sigma \in \{-1,+1\}, \tag{4}$$

$$T_0(x,y) \to T(x,y), \tag{5}$$

where $T^\sigma(y,z)$ stands for $T(y,z)$ if $\sigma = +1$ and $T(z,y)$ if $\sigma = -1$, and a TBox $\mathcal{T}$ containing the following concept inclusions:

$$\exists T \sqsubseteq \exists S, \qquad \exists T_0^- \sqsubseteq \exists T_0 \sqcap C_{\text{\textvisiblespace}}, \tag{6}$$

$$\exists P_{qa}^- \sqsubseteq H_q, \qquad \exists P_{qa} \sqsubseteq C_a, \qquad \text{for } q \in Q \text{ and } a \in \Gamma, \tag{7}$$

$$H_q \sqsubseteq D_\sigma, \qquad \exists T_{\emptyset\sigma}^- \sqsubseteq D_\sigma \sqcap H_\emptyset, \qquad \text{for } q \in Q \text{ and } \sigma \in \{-1,+1\}, \tag{8}$$

$$H_{q_1} \sqsubseteq \bot. \tag{9}$$

For every input $\boldsymbol{w} = a_1 \dots a_n \in \Gamma^*$, we take the following ABox $\mathcal{A}_{\boldsymbol{w}}$:

$$H_{q_0}(c_1), \qquad C_{a_i}(c_i) \text{ and } T(c_i, c_{i+1}), \text{ for } 1 \le i \le n, \qquad T_0(c_n, c_{n+1}).$$

It can be shown that $(\mathcal{T}, \mathcal{A}_{\boldsymbol{w}}) \not\models \boldsymbol{q}$ iff M does not accept $\boldsymbol{w}$. Indeed, consider a model $\mathcal{I}$ of $(\mathcal{T}, \mathcal{A}_{\boldsymbol{w}})$ with $\mathcal{I} \not\models \boldsymbol{q}$. Then, by the definition of the ABox, (5) and (6), there exists an infinite sequence of (not necessarily distinct) domain elements $d_1, d_2, \dots$ that encode the initial configuration in a sense that $(d_i, d_{i+1}) \in T^{\mathcal{I}}$ for all $i \ge 1$, $d_1 \in H_{q_0}^{\mathcal{I}}$, $d_i \in H_\emptyset^{\mathcal{I}}$ for all $i > 1$, $d_i \in C_{a_i}^{\mathcal{I}}$, for each $1 \le i \le n$, and $d_i \in C_{\text{\textvisiblespace}}^{\mathcal{I}}$ for all $i > n$. By (6) and (1), there exists another sequence of T-connected domain elements $d_1', d_2', \dots$, such that $(d_i, d_i') \in S^{\mathcal{I}}$. This sequence represents the second configuration of M. Indeed, by (2) and (7), the head position and the state are changed according to the transition function δ of M. By (8) and (4), the domain element representing the head, say, d_k, belongs to $D_{+1}^{\mathcal{I}}$, whereas all d_i with $i > k$ belong to $D_{+1}^{\mathcal{I}}$ and $H_\emptyset^{\mathcal{I}}$. Similarly, $d_i \in H_\emptyset^{\mathcal{I}}$, for all $i < k$. Therefore, all cells but the one under the head belong to $H_\emptyset^{\mathcal{I}}$, whence, by (3), their contents is preserved by the transition. By the same reasoning, there exists a respective sequence of elements for each configuration of the computation of M. Finally, (9) guarantees that the accepting state never occurs in the computation, i.e., M does not accept $\boldsymbol{w}$. The converse direction is straightforward: the non-accepting computation of M, if it exists, can be encoded by an infinite two-dimensional grid satisfying $(\mathcal{T}, \mathcal{A}_{\boldsymbol{w}})$ and the negation of $\boldsymbol{q}$. ❑

We remark that the number of CQs (and safe negations) in the UCQ$^{\neg s}$ $\boldsymbol{q}$ in the proof of Theorem 1 depends on the number of states and symbols of the universal Turing machine we encode (more precisely, it is $4 + (|Q| + 1) \cdot |\Gamma|$).

We now proceed to show that under rather mild restrictions (satisfied by the query in Theorem 1), a UCQ$^{\neg s}$ $\boldsymbol{q}$ can be transformed into a single CQ$^{\neg s}$ $\boldsymbol{q}'$ with the same number of safe negations although at a price of introducing role inclusions in the TBox (Theorem 1 holds for *DL-Lite*$_{\text{core}}$). Intuitively, $\boldsymbol{q}'$ is a conjunction of all disjuncts $\boldsymbol{q}_i$ of $\boldsymbol{q}$, each with an atom $G_i(x, x_i)$ attached to it, where x is a common fresh existentially quantified variable and x_i is some (existentially quantified) variable of $\boldsymbol{q}_i$. Then, we extend the TBox to ensure that on every domain element of a model of the original TBox, the extended TBox 'generates', for each i, an incoming G_i-arrow from a constellation matching all disjuncts of $\boldsymbol{q}$ but $\boldsymbol{q}_i$. So, if a part of the model for the original TBox matches some $\boldsymbol{q}_i$ then it matches $\boldsymbol{q}'$ as well, because the rest of the match is provided by the generated constellations. We now present a more formal treatment. A Boolean CQ$^{\neg s}$ $\boldsymbol{q}$ is *tree-shaped* if it is connected, does not have individuals as terms and the primal graph of its positive part contains no cycles (the primal graph has an edge between variables z and z' just in case the query contains an atom $P(z, z')$).

Lemma 1. *Let $\mathcal{T}$ be a DL-Lite$^{\mathcal{H}}_{\text{core}}$ TBox and $\boldsymbol{q}$ a Boolean UCQ$^{\neg s}$ such that each disjunct $\boldsymbol{q}_i$ of $\boldsymbol{q}$ is tree-shaped and contains*

$$\textit{neither}\quad A(x) \textit{ with } \mathcal{T} \models A \sqsubseteq \exists R \quad \textit{nor} \quad S(x, z) \textit{ with } \mathcal{T} \models \exists S \sqsubseteq \exists R, \tag{10}$$

for every $\neg R(x, v)$ in $\boldsymbol{q}_i$. Then there exist a CQ$^{\neg s}$ $\boldsymbol{q}'$ and a DL-Lite$^{\mathcal{H}}_{\text{core}}$ TBox $\mathcal{T}'$ such that $(\mathcal{T}, \mathcal{A}) \models \boldsymbol{q}$ iff $(\mathcal{T}', \mathcal{A}) \models \boldsymbol{q}'$, for every ABox $\mathcal{A}$.

Proof. Let $\boldsymbol{q}$ be the union of $\boldsymbol{q}_i = \exists \boldsymbol{x}_i\, \varphi_i(\boldsymbol{x}_i)$, for $1 \leq i \leq n$. Without loss of generality, we can assume that the $\boldsymbol{x}_i$ are pairwise disjoint and that each $\boldsymbol{x}_i$ contains at least one variable, say, x_i. Let x be a fresh variable and, for each $1 \leq i \leq n$, let G_i be a fresh role name and define $\varphi'_i(x, \boldsymbol{x}_i) = G_i(x, x_i) \wedge \varphi_i(\boldsymbol{x}_i)$. Consider

$$\boldsymbol{q}' \quad = \quad \exists x\, \boldsymbol{x}_1 \dots \boldsymbol{x}_n \, \big(\varphi'_1(x, \boldsymbol{x}_1) \wedge \dots \wedge \varphi'_n(x, \boldsymbol{x}_n)\big).$$

Let D be a fresh concept name. Denote by $\mathcal{T}_D$ be the result of attaching the subscript 0 to each concept and role name in $\mathcal{T}$ and extending the TBox by $A_0 \sqsubseteq A \sqcap D$, for each concept name A, and by $P_0 \sqsubseteq P$ and $\exists P_0 \sqcup \exists P_0^- \sqsubseteq D$, for each role name P in $\mathcal{T}$ (the interpretation of D will contain the interpretations of all concepts of $\mathcal{T}_D$ including domains and ranges of its roles).

Since the positive part of each $\varphi'_i(x, \boldsymbol{x}_i)$ is tree-shaped, it has a spanning tree with root x; moreover, that root has a single successor, x_i. We will write $y \prec z$ if y is a (unique) predecessor of z in the spanning trees. For each edge (y, z) with $y \prec z$, we take a fresh role E_{yz}. Let $\mathcal{T}_G$ contain the following inclusions, for all $1 \leq i \leq n$:

$$D \sqsubseteq \exists (G_i^0)^-, \tag{11}$$

$$\exists G_i^0 \sqsubseteq \exists G_j^1, \qquad \text{for all } 1 \leq j \leq n \text{ with } j \neq i, \tag{12}$$

$$G_i^k \sqsubseteq G_i, \qquad \text{for } k = 0, 1, \tag{13}$$

$$G_i^1 \sqsubseteq E_{xx_i}, \tag{14}$$

$$\exists E_{yz}^{-} \sqsubseteq \exists E_{zv}, \qquad \text{for each } y \prec z \prec v, \tag{15}$$

$$\exists E_{yz}^{-} \sqsubseteq A, \qquad \text{for each } A(z) \text{ in } \varphi_i' \text{ with } y \prec z, \tag{16}$$

$$E_{yz} \sqsubseteq R, \qquad \text{for each } R(y,z) \text{ in } \varphi_i' \text{ with } y \prec z, \tag{17}$$

$$\exists E_{yz}^{-} \sqcap \exists R \sqsubseteq \bot, \qquad \text{for each } \neg R(z,v) \text{ in } \varphi_i' \text{ with } y \prec z. \tag{18}$$

Let $\mathcal{T}' = \mathcal{T}_D \cup \mathcal{T}_G$. We claim that $(\mathcal{T}, \mathcal{A}) \models \boldsymbol{q}$ iff $(\mathcal{T}', \mathcal{A}) \models \boldsymbol{q}'$, for every $\mathcal{A}$. Indeed, suppose that $(\mathcal{T}, \mathcal{A}) \models \boldsymbol{q}$ and let $\mathcal{I}$ be a model of $(\mathcal{T}', \mathcal{A})$. Then $\mathcal{I} \models \mathcal{T}_D$, whence, by construction, $\mathcal{I} \models (\mathcal{T}, \mathcal{A})$. Thus, $\mathcal{I} \models \boldsymbol{q}$ and so, for some $1 \leq i \leq n$, there exists a match π for $\boldsymbol{q}_i$ in $\mathcal{I}$. By construction, $\pi(x_i)$ belongs to $A^{\mathcal{I}}$, for a concept name A of $\mathcal{T}$, or to $(\exists R)^{\mathcal{I}}$, for a role R of $\mathcal{T}$; whence, $\pi(x_i) \in D^{\mathcal{I}}$. Let $\boldsymbol{q}_*$ consist of all atoms of $\boldsymbol{q}'$ not in $\varphi_i(\boldsymbol{x}_i)$. Since $\mathcal{I} \models \mathcal{T}_G$, there exists a match π' for $\boldsymbol{q}_*$ in $\mathcal{I}$ with $\pi'(x_i) = \pi(x_i)$. Indeed, by (14)–(16), the tree of positive atoms of $\boldsymbol{q}_*$ is matched by the $(G_i^0)^-$-successor of $\pi(x_i)$; by (18), the negative atoms are satisfied by π'. Hence, $\pi \cup \pi'$ is a match for $\boldsymbol{q}'$ in $\mathcal{I}$.

Conversely, let $\mathcal{I}$ be a model of $(\mathcal{T}, \mathcal{A})$ with $\mathcal{I} \not\models \boldsymbol{q}$. Denote by $\mathcal{I}_0$ an interpretation that coincides with $\mathcal{I}$ on all individuals and concept and role names of $\mathcal{T}$ and, additionally, interprets D by $\Delta^{\mathcal{I}}$, each A_0 by $A^{\mathcal{I}}$, for a concept name A in $\mathcal{T}$, and each P_0 by $P^{\mathcal{I}}$, for a role name P in $\mathcal{T}$. Clearly, $\mathcal{I}_0 \models (\mathcal{T}_D, \mathcal{A})$ and $\mathcal{I}_0 \not\models \boldsymbol{q}$. Let $\mathcal{I}'$ be the (finite) chase of $\mathcal{I}_0$ with $\mathcal{T}_G$, which exists by (10). By definition, $\mathcal{I}' \models (\mathcal{T}', \mathcal{A})$. The chase part, however, ensures that $\mathcal{I}' \not\models \boldsymbol{q}'$. ❑

The UCQ$^{\neg s}$ $\boldsymbol{q}$ in the proof of Theorem 1 satisfies the conditions of Lemma 1, thus solving the open problem of decidability of CQ$^{\neg s}$ answering over *DL-Lite*$^{\mathcal{H}}_{\text{core}}$ [3, 5].

Theorem 2. *There exist a CQ*$^{\neg s}$ $\boldsymbol{q}$ *and a DL-Lite*$^{\mathcal{H}}_{\text{core}}$ *TBox* $\mathcal{T}$ *such that the problem* CERTAIN ANSWERS $(\boldsymbol{q}, \mathcal{T})$ *is undecidable.*

4 Answering CQs with One Inequality is Undecidable

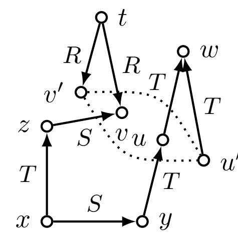
In this section we prove that CQ$^{\neq}$ answering over *DL-Lite*$^{\mathcal{H}}_{\text{core}}$ is undecidable. In principle, the technique of Lemma 1 can be adapted to queries with inequalities and by using, e.g., a modification of the proof of Theorem 1 [5], this would prove the claim. The resulting CQ$^{\neq}$ would, however, contain many inequalities. Instead, we substantially rework some ideas of the undecidability proof for CQ$^{\neq}$ answering over $\mathcal{EL}$ [9] and show that even one inequality suffices for *DL-Lite*$^{\mathcal{H}}_{\text{core}}$.

Theorem 3. *There exist a Boolean CQ*$^{\neq}$ $\boldsymbol{q}$ *with one inequality and a DL-Lite*$^{\mathcal{H}}_{\text{core}}$ *TBox* $\mathcal{T}$*, such that the problem* CERTAIN ANSWERS $(\boldsymbol{q}, \mathcal{T})$ *is undecidable.*

Proof. The proof is by reduction of the halting problem for deterministic Turing machines (see Theorem 1). In this proof we use a two-dimensional grid of similar structure. The grid is established (along with functionality of certain roles) by means of a CQ$^{\neq}$ $\boldsymbol{q}$, which is the existential closure of the negation of the following first-order formula:

$$S(x,y) \land T(x,z) \land S(z,v) \land T(y,u) \land$$
$$T(u,w) \land T(u',w) \land R(t,v) \land R(t,v')$$
$$\to (u' = v').$$

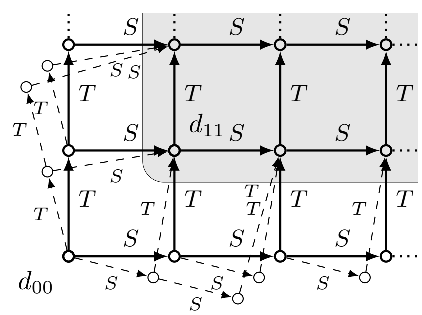
Note that this formula, in fact, implies $v = v' = u' = u$; see the dotted shape in the picture on the right.



We present the construction of the TBox $\mathcal{T}$ in a series of smaller TBoxes. As an aid to our explanations, we assume that an interpretation $\mathcal{I}$ with $\mathcal{I} \not\models \boldsymbol{q}$ is given; for each of the building blocks of $\mathcal{T}$ we then show that if $\mathcal{I}$ is a model of the TBox then $\mathcal{I}$ enjoys certain structural properties. So, let TBox $\mathcal{T}_G$ contain the following concept inclusions:

$$\exists S^- \sqsubseteq \exists T, \qquad\qquad \exists T^- \sqsubseteq \exists T, \qquad\qquad \exists S^- \sqsubseteq \exists R^-.$$

If $\mathcal{I} \models \mathcal{T}_G$ and $\mathcal{I} \models \exists T \sqsubseteq \exists S$ then the fragment of $\mathcal{I}$ rooted in $d_{00} \in (\exists T)^{\mathcal{I}}$ has a grid-like structure depicted below (each domain element in $(\exists S^-)^{\mathcal{I}}$ also has an $R^{\mathcal{I}}$-predecessor, which is not shown).



Observe that $S^{\mathcal{I}}$ and $T^{\mathcal{I}}$ are functional in all domain elements in the shaded area (we say that, e.g., $S^{\mathcal{I}}$ is *functional in* d if $d' = d''$ whenever $(d,d'), (d,d'') \in S^{\mathcal{I}}$). Let $\circ$ denote composition: e.g., $S^{\mathcal{I}} \circ T^{\mathcal{I}} = \{(d,d'') \mid (d,d') \in S^{\mathcal{I}}, (d',d'') \in T^{\mathcal{I}}\}$. Then the domain elements in the shaded area enjoy the following property.

Claim 3.1. If $\mathcal{I} \models \mathcal{T}_G$ and $\mathcal{I} \not\models \boldsymbol{q}$ then, for every d with an $(S^-)^{\mathcal{I}} \circ T^{\mathcal{I}} \circ S^{\mathcal{I}}$-predecessor,

- both $S^{\mathcal{I}}$ and $T^{\mathcal{I}}$ are functional in d,
- the $S^{\mathcal{I}} \circ T^{\mathcal{I}}$- and $T^{\mathcal{I}} \circ S^{\mathcal{I}}$-successors of d coincide,
- $(T^-)^{\mathcal{I}}$ is functional in the $T^{\mathcal{I}}$-successor of d,
- $R^{\mathcal{I}}$ is functional in any $R^{\mathcal{I}}$-predecessor of d.

Note that $S^{\mathcal{I}}$ does not have to be functional in the bottom row and $T^{\mathcal{I}}$ in the left column (see the picture above); $(T^-)^{\mathcal{I}}$ does not have to be functional outside the shaded area and in the first row of the shaded area; $R^{\mathcal{I}}$ does not have to be functional anywhere but in $R^{\mathcal{I}}$-predecessors of the domain elements in the shaded area; $(S^-)^{\mathcal{I}}$ and $(R^-)^{\mathcal{I}}$ do not have to be functional anywhere. For our purposes, however, it suffices that $\mathcal{I}$ has a grid structure starting from d_{11}; moreover, as we shall see, the non-functionality of $(S^-)^{\mathcal{I}}$ plays a crucial role in the construction.

In addition to the grid-like structure of $S^{\mathcal{I}}$ and $T^{\mathcal{I}}$, we also need functionality of $S^{\mathcal{I}}$ in points outside the grid. To this end, we use a technique similar to the proof of Lemma 1. Let TBox $\mathcal{T}_S$ contain the following concept and role inclusions, for a fresh concept name E and a fresh role name V (similar to the 'edge' roles E_{yz} in Lemma 1):

$$E \sqsubseteq \exists S, \qquad E \sqsubseteq \exists V, \qquad V \sqsubseteq T^-, \qquad \exists V^- \sqsubseteq \exists S.$$

Claim 3.2. If $\mathcal{I} \models \mathcal{T}_G \cup \mathcal{T}_S$ and $\mathcal{I} \not\models \boldsymbol{q}$ then $S^{\mathcal{I}}$ is functional in every $d \in E^{\mathcal{I}}$.

We also require role R to be functional not only in $R^{\mathcal{I}}$-predecessors of the grid points but also in the grid points themselves. Let TBox $\mathcal{T}_R$ contain the following inclusions, for a fresh concept name D and fresh role names U_0, U_1 and U_2, with $i = 1, 2$:

$$D \sqsubseteq \exists U_0, \quad U_0 \sqsubseteq R, \quad \exists U_{i-1}^- \sqsubseteq \exists U_i, \quad U_1 \sqsubseteq S^-, \quad U_2 \sqsubseteq T^-, \quad \exists U_2^- \sqsubseteq \exists S.$$

Claim 3.3. If $\mathcal{I} \models \mathcal{T}_G \cup \mathcal{T}_R$ and $\mathcal{I} \not\models \boldsymbol{q}$ then $R^{\mathcal{I}}$ is functional in every $d \in D^{\mathcal{I}}$.

We now describe a TBox that encodes computations of a given Turing machine. Let $M = (\mathit{\Gamma}, Q, q_0, q_1, \delta)$ be a deterministic Turing machine, where $\mathit{\Gamma} = \{1, \llcorner\!\lrcorner\}$ is a *two-symbol* tape alphabet, Q a set of states, $q_0 \in Q$ an initial and $q_1 \in Q$ an accepting state, and $\delta \colon Q \times \mathit{\Gamma} \to Q \times \mathit{\Gamma} \times \{-1, +1\}$ a deterministic transition function.

We use concept H_q, for $q \in Q$, that contains the representations of all tape cells observed by the head of M (in state q); concept $H_\emptyset$ represents the cells not observed by the head of M. Role S has two sub-roles, $S_{\llcorner\!\lrcorner}$ and S_1, for the two symbols of the alphabet $\mathit{\Gamma}$ to encode cell contents: all cells represented by the range of S_a contain $a \in \mathit{\Gamma}$.

The most natural way of encoding a transition $\delta(q, a) = (q', a', \sigma)$ of M would be to use a concept inclusion of the form $\exists H_q \sqcap S_a^- \sqsubseteq \exists S_{a'} \sqcap \exists S_{q'\sigma}$, where $S_{q'\sigma}$ is also a sub-role of role S, which is functional in the grid. Alas, *DL-Lite*$^{\mathcal{H}}_{\mathit{core}}$ does not have conjunction on the left-hand side of concept inclusions. The following construction allows us to simulate the required inclusions by using functionality of just two roles, R and S. Let $\mathcal{T}_F$ be the union of $\mathcal{T}_S$, $\mathcal{T}_R$ and the following concept and role inclusions with fresh role names P_q, Q_a and P_{qa}, for each $q \in Q \cup \{\emptyset\}$ and $a \in \mathit{\Gamma}$:

$$H_q \sqsubseteq \exists P_q \sqcap D, \qquad \exists S_a^- \sqsubseteq \exists Q_a, \qquad P_q \sqcup Q_a \sqsubseteq R,$$
$$\exists P_q^- \sqsubseteq \exists P_{q\llcorner\!\lrcorner} \sqcap \exists P_{q1} \sqcap E, \qquad P_{q\llcorner\!\lrcorner} \sqcup Q_{\llcorner\!\lrcorner}^- \sqsubseteq R, \qquad P_{q1} \sqcup Q_1^- \sqsubseteq S.$$

Claim 3.4. If $\mathcal{I} \models \mathcal{T}_G \cup \mathcal{T}_F$ and $\mathcal{I} \not\models \boldsymbol{q}$ then $d \in (\exists P_{qa}^-)^{\mathcal{I}}$ whenever $d \in H_q^{\mathcal{I}} \cap (\exists S_a^-)^{\mathcal{I}}$, for each d with an $(S^-)^{\mathcal{I}} \circ T^{\mathcal{I}} \circ S^{\mathcal{I}}$-predecessor, each $q \in Q \cup \{\emptyset\}$ and $a \in \mathit{\Gamma}$.

Proof of claim. Let $d \in H_q^{\mathcal{I}} \cap (\exists S_1^-)^{\mathcal{I}}$. Then d has a $P_q^{\mathcal{I}}$- and a $Q_1^{\mathcal{I}}$-successor, which coincide since $R^{\mathcal{I}}$ is functional in $d \in D^{\mathcal{I}}$. Let d' be the $R^{\mathcal{I}}$-successor of d. The *inverse* of Q_1 is also a sub-role of S, and thus, $(d, d') \in S^{\mathcal{I}}$. On the other hand, d' has a $P_{q1}^{\mathcal{I}}$-successor d'', whence $(d'', d') \in S^{\mathcal{I}}$. By Claim 3.3, $S^{\mathcal{I}}$ is functional in d', whence $d = d''$. Thus, $d \in (\exists P_{q1}^-)^{\mathcal{I}}$. For $d \in H_q^{\mathcal{I}} \cap (\exists S_{\llcorner\!\lrcorner}^-)^{\mathcal{I}}$, the argument is similar with R replacing S as a super-role of both $P_{q\llcorner\!\lrcorner}$ and $Q_{\llcorner\!\lrcorner}^-$ ($R^{\mathcal{I}}$ is functional in any $R^{\mathcal{I}}$-predecessor of d by Claim 3.1). ❑

We are now in a position to define the encoding of Turing machine computations. Using the roles P_{qa} from $\mathcal{T}_F$, we can encode transitions:

$$\exists P_{qa}^- \sqsubseteq \exists S_{a'} \sqcap \exists S_{q'\sigma}, \qquad \text{for } q \in Q \text{ and } a \in \Gamma \text{ with } \delta(q,a) = (q', a', \sigma), \quad (19)$$

$$S_a \sqcup S_{q\sigma} \sqsubseteq S, \qquad \text{for } a \in \Gamma, q \in Q \text{ and } \sigma \in \{-1, +1\}, \quad (20)$$

where $S_{q,-1}$ and $S_{q,+1}$ are fresh role names used to propagate the new state to the next configuration. Recall now that roles $P_{\emptyset a}$ identify cells that are not observed by the head of M; the contents of such cells is then preserved with the help of concept inclusions

$$\exists P_{\emptyset a}^- \sqsubseteq \exists S_a, \qquad \text{for } a \in \Gamma. \quad (21)$$

The location of the head in the next configuration is ensured by the following inclusions:

$$\exists S_{q\sigma}^- \sqsubseteq \exists T_{q\sigma}, \qquad \exists T_{q\sigma}^- \sqsubseteq H_q, \qquad \text{for } q \in Q \text{ and } \sigma \in \{-1, +1\}, \quad (22)$$

$$T_{q,+1} \sqsubseteq T, \qquad T_{q,-1} \sqsubseteq T^-, \qquad \text{for } q \in Q \cup \{\emptyset\}, \quad (23)$$

where the $T_{q,+1}$ and $T_{q,-1}$ are used to propagate the head in the state q along the tape (both T and T^- are functional in the grid); finally, the following concept inclusions with (23) for $q = \emptyset$ are required to propagate the no-head marker $H_\emptyset$:

$$H_q \sqsubseteq \exists T_{\emptyset\sigma} \qquad \exists T_{\emptyset\sigma}^- \sqsubseteq \exists T_{\emptyset\sigma} \sqcap H_\emptyset, \qquad \text{for } q \in Q \text{ and } \sigma \in \{-1, +1\}. \quad (24)$$

Next, we define the ABox $\mathcal{A}_{\boldsymbol{w}}$ that encodes an input $\boldsymbol{w} = a_1, \dots, a_n \in \Gamma^*$ of M:

$$Z(c_{00}, c_{10}), \qquad T(c_{10}, c_{11}), \qquad H_{q_0}(c_{11}),$$
$$T(c_{0(i-1)}, c_{0i}) \text{ and } S_{a_i}(c_{0i}, c_{1i}), \quad \text{for } 1 \leq i \leq n, \qquad T_0(c_{0n}, c_{0(n+1)}),$$

where Z is a fresh role name to create the bottom row of the grid and T_0 is a fresh role name to fill the rest of the tape by blanks:

$$\exists Z^- \sqsubseteq \exists Z, \qquad Z \sqsubseteq S, \qquad \exists T_0^- \sqsubseteq \exists S_\llcorner \sqcap \exists T_0, \qquad T_0 \sqsubseteq T. \quad (25)$$

Finally, the following ensures that the accepting state q_1 never occurs in a computation:

$$H_{q_1} \sqsubseteq \bot. \quad (26)$$

Let $\mathcal{T}_M$ contain (19)–(26) encoding computations of M and $\mathcal{T} = \mathcal{T}_G \cup \mathcal{T}_F \cup \mathcal{T}_M$. If $(\mathcal{T}, \mathcal{A}_{\boldsymbol{w}}) \not\models \boldsymbol{q}$ then there is a model $\mathcal{I}$ of $(\mathcal{T}, \mathcal{A}_{\boldsymbol{w}})$ with $\mathcal{I} \not\models \boldsymbol{q}$. It should then be clear that in this case we can extract a computation of M encoded by $\mathcal{I}$ and that computation does not accept $\boldsymbol{w}$. Conversely, if M does not accept $\boldsymbol{w}$ then we can construct a model $\mathcal{I}$ of $(\mathcal{T}, \mathcal{A}_{\boldsymbol{w}})$ such that $\mathcal{I} \not\models \boldsymbol{q}$. First, it is routine to construct a model $\mathcal{J}$ of $\mathcal{T}_G$ with

$$\Delta^{\mathcal{J}} \;=\; \{d_{ij} \mid i, j \geq 0\} \;\cup\; \{d'_{ij}, d''_{ij} \mid i > 0 \text{ and } j \geq 0\} \;\cup\; \{b_i \mid i > 0\}$$

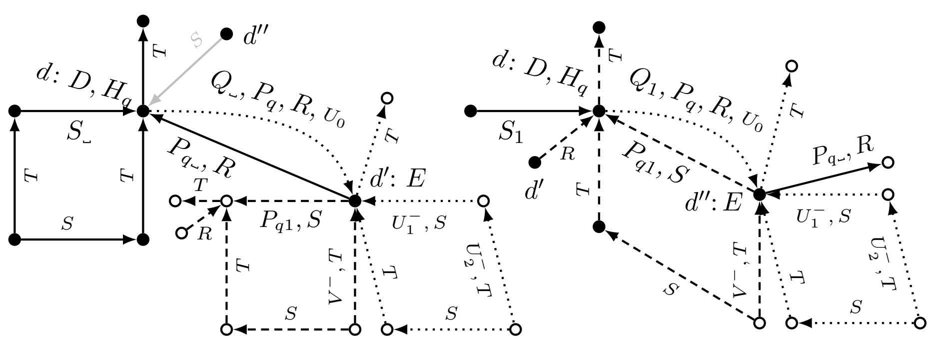
such that the d_{ij} form a grid structure on roles S and T, each d'_{ij} is an $R^{\mathcal{J}}$-predecessor of d_{ij} and each d''_{ij} is an $S^{\mathcal{J}}$-predecessor of d_{ij} (note that d_{ij} has another $S^{\mathcal{J}}$-predecessor, $d_{(i-1)j}$). Next, we choose the interpretation of concepts and roles in $\mathcal{T}_M$ on the domain of $\mathcal{J}$ in such a way that the part of $\mathcal{J}$ rooted in d_{11} encodes a unique computation of M on $\boldsymbol{w}$ and $\mathcal{J} \models (\mathcal{T}_M, \mathcal{A}_{\boldsymbol{w}})$. In particular, the computation determines the interpretation of H_q, S_a and $S_{q\sigma}$, for $q \in Q$, $a \in \Gamma$ and $\sigma \in \{-1, +1\}$. It then should be clear how to interpret $H_\emptyset$ and $T_{q\sigma}$, for $q \in Q \cup \{\emptyset\}$ and $\sigma \in \{-1, +1\}$: the only non-trivial case is $T_{\emptyset,-1}$, where, in order to satisfy (24), we take b_i to be a $T^{\mathcal{J}}_{\emptyset,-1}$-successor (and so, a $T^{\mathcal{J}}$-predecessor) of both d_{i1} and b_i, for each $i > 0$ (as we noted, $(T^-)^{\mathcal{J}}$ does not have to be functional in any d_{i1}; $T^{\mathcal{J}}$, however, must be functional in each d_{i0} and cannot have a $T^{\mathcal{J}}$-loop). As the final step of the construction of $\mathcal{J}$, we set

$$(d'_{ij}, d_{ij}) \in P^{\mathcal{J}}_{q\textvisiblespace} \text{ and } (d_{ij}, d'_{ij}) \in R^{\mathcal{J}} \quad \text{if} \quad d_{ij} \in H^{\mathcal{J}}_q \cap (\exists S^-_{\textvisiblespace})^{\mathcal{J}},$$
$$(d''_{ij}, d_{ij}) \in P^{\mathcal{J}}_{q1} \text{ and } (d_{ij}, d''_{ij}) \in R^{\mathcal{J}} \quad \text{if} \quad d_{ij} \in H^{\mathcal{J}}_q \cap (\exists S^-_1)^{\mathcal{J}}.$$

It remains to show that $\mathcal{J}$ can be extended to satisfy $\mathcal{T}_F$. Observe that only concept names H_q and role names R, S, T, S_a and P_{qa}, for $q \in Q \cup \{\emptyset\}$ and $a \in \Gamma$, are shared between $\mathcal{T}_F$ and $\mathcal{T}_G \cup \mathcal{T}_M$; all other concept and roles names in $\mathcal{T}_F$ are *fresh in* $\mathcal{T}_F$. We show that $\mathcal{J}$ can be extended (by fresh domain elements) to a model of $\mathcal{T}_F$ without changing concepts and roles on grid, i.e., the domain elements of $\mathcal{J}$.

Claim 3.5. If $\mathcal{J} \models \mathcal{T}_G$ and $\mathcal{J} \not\models \boldsymbol{q}$ then $\mathcal{J}$ can be extended to a model $\mathcal{I}$ of $\mathcal{T}_F$ so that
(a) $d \in H^{\mathcal{I}}_q \cap (\exists S^-_a)^{\mathcal{I}}$ whenever $d \in (\exists P^-_{qa})^{\mathcal{I}}$, for every $d \in \Delta^{\mathcal{J}}$ with an $(S^-)^{\mathcal{J}} \circ T^{\mathcal{J}} \circ S^{\mathcal{J}}$-predecessor, an $R^{\mathcal{J}}$-predecessor d' and another $S^{\mathcal{J}}$-predecessor d'', and
(b) $A^{\mathcal{I}} \cap \Delta^{\mathcal{J}} = A^{\mathcal{J}}$ and $P^{\mathcal{I}} \cap (\Delta^{\mathcal{J}} \times \Delta^{\mathcal{J}}) = P^{\mathcal{J}}$, for all concept names A and role names P that are *not fresh* in $\mathcal{T}_F$.

Proof of claim. The cases of $P_{q\textvisiblespace}$ and P_{q1} are illustrated below on the left and right, respectively; some edges are not shown to avoid clutter: each domain element in $(\exists S^-)^{\mathcal{I}}$ also has an incoming $R^{\mathcal{I}}$-edge and each $T^{\mathcal{I}}$-edge starts an infinite chain of $T^{\mathcal{I}}$-edges.



The three black (solid, dashed and dotted) patterns of edges on the left correspond to the three sets of positive atoms of $\boldsymbol{q}$ so that the inequality atom, $(u' = v')$, 'identifies' certain domain elements of the pattern; similarly, for the two patterns on the right. Black nodes are in the domain of $\mathcal{J}$, while white nodes are in the domain of $\mathcal{I}$ proper. It can be seen that d is added only to D, and (d, d') or (d, d''), depending on the $(\exists S^-_a)^{\mathcal{J}}$, are added only to roles P_q, Q_a and U_0 (which are all fresh in $\mathcal{T}_F$). ❑

So, $(\mathcal{T}, \mathcal{A}_{\boldsymbol{w}}) \not\models \boldsymbol{q}$ iff M does not accept $\boldsymbol{w}$. Take M to be a fixed deterministic *universal* Turing machine, i.e., a machine that accepts $\boldsymbol{w}$ iff the empty input is accepted by the Turing machine encoded by $\boldsymbol{w}$. This finishes the proof of Theorem 3. ❑

5 Lower Bounds for CQ$^{\neq}$ Answering without Role Inclusions

In the previous sections we established undecidability of CQ$^{\neg s}$ and CQ$^{\neq}$ answering over *DL-Lite*$^{\mathcal{H}}_{core}$. The reductions, however, essentially use role inclusions. Leaving the problems of decidability of CQ$^{\neg s}$ and CQ$^{\neq}$ answering over *DL-Lite*$_{core}$ open, we establish lower complexity bounds for the second case.

Theorem 4. *There exist a Boolean CQ*$^{\neq}$ $\boldsymbol{q}$ *with one inequality and a DL-Lite*$_{core}$ *TBox* $\mathcal{T}$ *such that the problem* CERTAIN ANSWERS $(\boldsymbol{q}, \mathcal{T})$ *is* P-*hard.*

Proof. The proof is by reduction of the complement of HORN-3SAT, the satisfiability problem for Horn clauses with at most 3 literals, which is known to be P-complete (see e.g., [11]). Suppose we are given a conjunction ψ of clauses of the form p, $\neg p$, and $p_1 \wedge p_2 \to p$. Fix a TBox $\mathcal{T}$ containing the following concept inclusions:

$$V_T \sqcap V_F \sqsubseteq \bot, \qquad G \sqsubseteq \exists T, \qquad \exists T^- \sqsubseteq \exists T, \qquad \exists T^- \sqsubseteq V_T,$$

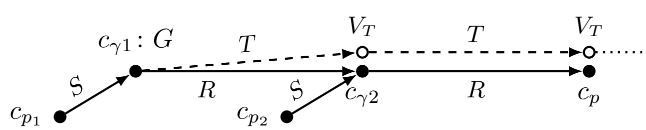
and a Boolean CQ$^{\neq}$ $\boldsymbol{q}$ which is the existential closure of the negation of the following:

$$V_T(x) \wedge S(x,y) \wedge R(y,z_1) \wedge T(y,z_2) \to (z_1 = z_2).$$

Note that $\mathcal{T}$ and $\boldsymbol{q}$ do not depend on ψ. Next, we construct an ABox $\mathcal{A}_\psi$ such that ψ is satisfiable iff $(\mathcal{T}, \mathcal{A}_\psi) \not\models \boldsymbol{q}$. The ABox $\mathcal{A}_\psi$ uses an individual name c_p, for each variable p in ψ, and individual names $c_{\gamma 1}$ and $c_{\gamma 2}$ for each clause γ of the form $p_1 \wedge p_2 \to p$ in ψ. For each clause γ, the ABox $\mathcal{A}_\psi$ contains the following assertions:

$$\begin{array}{ll} V_T(c_p), \quad \text{if } \gamma = p, & V_F(c_p), \quad \text{if } \gamma = \neg p, \\ S(c_{p_1}, c_{\gamma 1}),\ G(c_{\gamma 1}),\ R(c_{\gamma 1}, c_{\gamma 2}),\ S(c_{p_2}, c_{\gamma 2}),\ R(c_{\gamma 2}, c_p), & \text{if } \gamma = p_1 \wedge p_2 \to p. \end{array}$$

Suppose first there is a model $\mathcal{I}$ of $(\mathcal{T}, \mathcal{A}_\psi)$ with $\mathcal{I} \not\models \boldsymbol{q}$. We show that ψ is satisfiable. For each clause γ of ψ of the form $p_1 \wedge p_2 \to p$ (the other two cases are trivial), $\mathcal{I}$ contains a configuration depicted below (the black nodes represent ABox individuals and the white ones—anonymous individuals generated by the TBox).



If $c_{p_1}^{\mathcal{I}} \in V_T^{\mathcal{I}}$ then the $T^{\mathcal{I}}$- and $R^{\mathcal{I}}$-successors of $c_{\gamma 1}^{\mathcal{I}}$ coincide, whence $c_{\gamma 2}^{\mathcal{I}} \in (\exists T)^{\mathcal{I}}$, which triggers the second 'application' of the query to identify $c_p^{\mathcal{I}}$ with the $T^{\mathcal{I}}$-successor of $c_{\gamma 2}^{\mathcal{I}}$ resulting in $c_p^{\mathcal{I}} \in V_T^{\mathcal{I}}$ but *only if* $c_{p_2}^{\mathcal{I}} \in V_T^{\mathcal{I}}$. So, as follows from the argument above, we can define a satisfying assignment $\mathfrak{a}$ for ψ by taking $\mathfrak{a}(p)$ true iff $c_p^{\mathcal{I}} \in V_T^{\mathcal{I}}$.

Conversely, if ψ is satisfiable then we can construct a model $\mathcal{I}$ of $(\mathcal{T}, \mathcal{A}_\psi)$ with $\mathcal{I} \not\models \boldsymbol{q}$. ❑

Theorem 5. *There exist a Boolean CQ*$^{\neq}$ $\boldsymbol{q}$ *with two inequalities and a DL-Lite*$_{core}$ *TBox* $\mathcal{T}$ *such that the problem* CERTAIN ANSWERS $(\boldsymbol{q}, \mathcal{T})$ *is* CONP-*hard.*

Proof. The proof is by reduction of the complement of 3SAT, which is known to be coNP-complete (see e.g., [11]). Suppose we are given a conjunction ψ of clauses of the form $\ell_1 \vee \ell_2 \vee \ell_3$, where the ℓ_k are literals, i.e., propositional variables or their negations (we can assume that all literals in each clause are distinct). Fix a TBox $\mathcal{T}$ containing the following concept inclusions:

$$V_T \sqsubseteq \exists T \sqcap \exists F, \qquad \exists T^- \sqsubseteq V_T, \qquad \exists T^- \sqcap \exists F^- \sqsubseteq \bot, \qquad A_1 \sqcap A_2 \sqsubseteq \bot,$$

and a Boolean CQ$^{\neq}$ $\boldsymbol{q}$ which is the existential closure of the negation of the following:

$$V_T(x) \wedge P(x,y) \wedge T(x,y_1) \wedge F(x,y_2) \to (y = y_1) \vee (y = y_2).$$

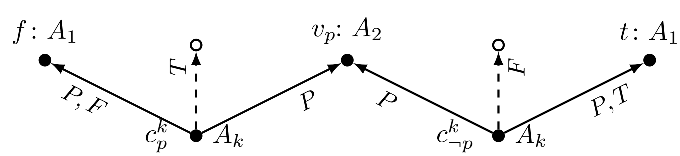
Claim 5.1. Let $\mathcal{I}$ be a model of $\mathcal{T}$ with $\mathcal{I} \not\models \boldsymbol{q}$. If $d \in V_T^{\mathcal{I}}$ and $(d, d_k) \in P^{\mathcal{I}}$, $d_k \in A_k^{\mathcal{I}}$, $k = 1, 2$, then either $(d, d_1) \in F^{\mathcal{I}}$ and $(d, d_2) \in T^{\mathcal{I}}$ or $(d, d_1) \in T^{\mathcal{I}}$ and $(d, d_2) \in F^{\mathcal{I}}$.

Proof of claim. Clearly, each pair (d, d_k) belongs either to $T^{\mathcal{I}}$ or $F^{\mathcal{I}}$. Suppose to the contrary that $(d, d_k) \in T^{\mathcal{I}}$, $k = 1, 2$. Consider $\boldsymbol{q}$ with $x \mapsto d$, $y \mapsto d_1$, $y_1 \mapsto d_2$ and any $F^{\mathcal{I}}$-successor of d as y_2. By disjointness of the A_k, $d_1 \neq d_2$, and so, we can only choose $y = y_2$, whence $(d, d_1) \in F^{\mathcal{I}}$ contrary to disjointness of $\exists T^-$ and $\exists F^-$. ❑

Again, $\mathcal{T}$ and $\boldsymbol{q}$ do not depend on ψ. The ABox $\mathcal{A}_\psi$ is constructed as follows. Let t and f be two individuals with $A_1(t)$ and $A_1(f)$ in $\mathcal{A}_\psi$. For each propositional variable p of ψ, take the following assertions, for $k = 1, 2$, with 5 individuals v_p, $c^k_{\neg p}$ and c^k_p:

$$\begin{array}{ll} A_2(v_p), \qquad\qquad & P(c^k_p, v_p), \;\; P(c^k_p, f), \;\; F(c^k_p, f), \;\; A_k(c^k_p), \\ & P(c^k_{\neg p}, v_p), \;\; P(c^k_{\neg p}, t), \;\; T(c^k_{\neg p}, t), \;\; A_k(c^k_{\neg p}), \end{array}$$

where the c^k_p and $c^k_{\neg p}$ represent the literals p and $\neg p$, respectively, see the picture below.



Observe that, by Claim 5.1, if $(c^k_{\neg p})^{\mathcal{I}} \in V_T^{\mathcal{I}}$ in a model $\mathcal{I}$ of $(\mathcal{T}, \mathcal{A}_\psi)$ with $\mathcal{I} \not\models \boldsymbol{q}$ then $v_p^{\mathcal{I}} \in (\exists F^-)^{\mathcal{I}}$, that is, if the literal $\neg p$ is chosen (by means of V_T) then p must be false; on the other hand, if $\neg p$ is not chosen (that is, $(c^k_{\neg p})^{\mathcal{I}} \notin V_T^{\mathcal{I}}$) then $v_p^{\mathcal{I}}$ does not have to be in $(\exists F^-)^{\mathcal{I}}$ and p can be anything; and similarly for $(c^k_p)^{\mathcal{I}}$ with $v_p^{\mathcal{I}} \in (\exists T^-)^{\mathcal{I}}$.

Next, $\mathcal{A}_\psi$ contains, for each clause γ of the form $\ell_1 \vee \ell_2 \vee \ell_3$ in ψ, the following assertions, where $c_{\gamma 1}$ and $c_{\gamma 2}$ are two fresh individuals:

$$V_T(c_{\gamma 1}), \qquad P(c_{\gamma 1}, c^1_{\ell_1}), \;\; P(c_{\gamma 1}, c_{\gamma 2}), A_2(c_{\gamma 2}), \qquad P(c_{\gamma 2}, c^1_{\ell_2}), \;\; P(c_{\gamma 2}, c^2_{\ell_3}).$$

It should be clear that ψ is satisfiable iff $(\mathcal{T}, \mathcal{A}_\psi) \not\models \boldsymbol{q}$. Indeed, if there is a model $\mathcal{I}$ of $(\mathcal{T}, \mathcal{A}_\psi)$ with $\mathcal{I} \not\models \boldsymbol{q}$ then, by Claim 5.1 and the observation above, we can construct a satisfying assignment $\mathfrak{a}$ for ψ by taking $\mathfrak{a}(p)$ true iff $v_p^{\mathcal{I}} \in V_T^{\mathcal{I}}$. The converse direction is straightforward and omitted due to space restrictions. ❑

6 Conclusions

Our investigation made further steps towards a clearer understanding of the impact of extending CQs with safe negation or inequalities on the complexity of the query answering problem in the OBDA paradigm. We showed that over $\mathit{DL\text{-}Lite}^{\mathcal{H}}_{\mathit{core}}$ ontologies these extensions lead to a surprisingly big increase, going from AC^0 for answering (positive) CQs to undecidability for answering $\mathrm{CQs}^{\neg s}$ and $\mathrm{CQs}^{\neq}$ with a single inequality. Furthermore, we showed that over the simpler $\mathit{DL\text{-}Lite}_{\mathit{core}}$ the problem for $\mathrm{CQs}^{\neq}$ is also harder than for CQs: P-hard for queries with one inequality and coNP-hard for queries with at least two inequalities. Two important problems are left as future work: decidability of answering $\mathrm{CQ}^{\neg s}$ and $\mathrm{CQs}^{\neq}$ over $\mathit{DL\text{-}Lite}_{\mathit{core}}$ ontologies.

Acknowledgements. The fourth author was supported by the UK EPSRC grant EP/J017728/1 (SOCIAM project).

References

1. 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) (2007) 385–429
2. Artale, A., Calvanese, D., Kontchakov, R., Zakharyaschev, M.: The DL-Lite family and relations. J. Artif. Intell. Res. (JAIR) **36** (2009) 1–69
3. Rosati, R.: The limits of querying ontologies. In: Proc. of the 11th Int. Conf. on Database Theory (ICDT). Volume 4353 of LNCS., Springer (2007) 164–178
4. Bienvenu, M., Ortiz, M., Simkus, M.: Answering expressive path queries over lightweight DL knowledge bases. In: Proc. of the 2012 Int. Workshop on Description Logics (DL). Volume 846 of CEUR-WS. (2012)
5. Gutiérrez-Basulto, V., Ibáñez-García, Y., Kontchakov, R.: An update on query answering with restricted forms of negation. In: Proc. of the 6th Int. Conf. on Web Reasoning and Rule Systems (RR). Volume 7497 of LNCS., Springer (2012) 75–89
6. Kostylev, E.V., Reutter, J.L.: Answering counting aggregate queries over ontologies of the DL-Lite family. In: Proc. of the 27th AAAI Conf. on Artificial Intelligence (AAAI). (2013)
7. Deutsch, A., Nash, A., Remmel, J.B.: The chase revisited. In: Proc. of the 27th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS), ACM Press (2008) 149–158
8. Kontchakov, R., Lutz, C., Toman, D., Wolter, F., Zakharyaschev, M.: The combined approach to query answering in DL-Lite. In: Proc. of the 12th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR), AAAI Press (2010)
9. Klenke, T.: Über die Entscheidbarkeit von konjunktiv Anfragen mit Ungleichheit in der Beschreibungslogik $\mathcal{EL}$. Master's thesis, Universität Bremen (2010)
10. Vardi, M.Y.: The complexity of relational query languages (extended abstract). In: Proc. of 14th Annual ACM Symposium on Theory of Computing (STOC), ACM (1982) 137–146
11. Papadimitriou, C.H.: Computational complexity. Academic Internet Publ. (2007)