Queries with Negation and Inequalities over Lightweight Ontologies

Víctor Gutiérrez-Basulto[a], Yazmín Ibáñez-García[a,b], Roman Kontchakov[c], Egor V. Kostylev[d]

[a] *Fachbereich Mathematik und Informatik, Universität Bremen, Germany*
[b] *KRDB Research Centre, Free University of Bozen-Bolzano, Italy*
[c] *Department of Computer Science and Information Systems, Birkbeck, University of London, UK*
[d] *Department of Computer Science, University of Oxford, UK and School of Informatics, University of Edinburgh, UK*

Abstract

While the problem of answering positive existential queries, in particular, conjunctive queries (CQs) and unions of CQs, over description logic ontologies has been studied extensively, there have been few attempts to analyse queries with negated atoms. Our aim is to sharpen the complexity landscape of the problem of answering CQs with negation and inequalities in lightweight description logics of the *DL-Lite* and $\mathcal{EL}$ families. We begin by considering queries with safe negation and show that there is a surprisingly significant increase in the complexity from AC^0 to undecidability (even if the ontology and query are fixed and only the data is regarded as input). We also investigate the problem of answering queries with inequalities and show that answering a single CQ with *one* inequality over *DL-Lite* with role inclusions is undecidable. In the light of our undecidability results, we explore syntactic restrictions to attain efficient query answering with negated atoms. In particular, we identify a novel class of local CQs with inequalities, for which query answering over *DL-Lite* is decidable.

Keywords: description logics, ontological query answering, conjunctive queries with negation, inequalities, DL-Lite, EL

1. Introduction

In recent years, the use of ontologies to access data has become one of the most prominent applications of description logic (DL) technologies in the Semantic Web. In the *ontology-based data access* (*OBDA*) setting, the 'plain' data is enriched with the background domain knowledge, which is represented in the form of a DL ontology. This distinguishing feature of the OBDA paradigm provides the user with a friendlier vocabulary for accessing data and extends information systems with a means of querying potentially incomplete data.

In classical database theory, conjunctive queries (CQs) have long played a key role due to their attractive theoretical properties. Following in these footsteps, a vast amount of research on answering CQs in the context of OBDA has been conducted in the last decade, so that we now have a fairly clear landscape of the computational complexity of answering CQs over both lightweight and expressive ontology languages. Moreover, with the aim of achieving a realistic use of OBDA in data-intensive Web applications, special efforts have been invested into the design of ontology languages with the following two desirable properties. First, they must be expressive enough to capture essential modelling aspects of the application domain. Second, they must allow OBDA systems to scale to large amounts of data. The latter can be achieved, for example, by delegating query evaluation to a relational database management system (RDBMS) or a datalog engine. DLs in the *DL-Lite* (Calvanese et al., 2007b; Artale et al., 2009) and $\mathcal{EL}$ (Baader et al., 2005) families were designed to meet these two requirements and underpin, respectively, the OWL 2 QL and OWL 2 EL profiles of the OWL 2 ontology language.[1] Notably, answering CQs and unions of CQs (UCQs) over OWL 2 QL ontologies is in AC^0 in data complexity, which enables a pure *query rewriting approach* to query answering in this case. Intuitively, one can rewrite a given query by including the knowledge provided by the ontology into an SQL query, which can then be answered by the RDBMS; see, e.g., (Calvanese et al., 2007b; Kikot et al., 2012) and references therein. Answering CQs (and UCQs) over OWL 2 EL ontologies is more complex, P-complete, and a pure query rewriting approach is not possible anymore. However, the so-called *combined approach* (Lutz et al., 2009; Kontchakov et al., 2010) allows one still to delegate query evaluation to the RDBMS. Roughly speaking, in the combined approach, not only the given query is rewritten but also the data is 'completed' with the knowledge of the ontology. A number of OBDA systems implementing these (and other) ideas have been developed; see, e.g., (Rodríguez-Muro et al., 2013; Lutz et al., 2013) and references therein.

Conjunctive queries belong to the positive existential fragment of first-order logic and therefore, lack any means of expressing '*complementation*' or '*difference*'. However, some natural queries require these constructs: for instance, retrieve '*all staff members who do* not *belong to any trade union*' or retrieve '*all students whose month of birth is* not (i.e., different

Email addresses: victor@informatik.uni-bremen.de (Víctor Gutiérrez-Basulto), ibanez@uni-bremen.de (Yazmín Ibáñez-García), roman@dcs.bbk.ac.uk (Roman Kontchakov), egor.kostylev@cs.ox.ac.uk (Egor V. Kostylev)

[1] www.w3.org/TR/owl2-profiles

from) *September'*. In order to overcome these shortcomings, extensions of CQs with some form of *negation* have been investigated in classical database theory and in different areas related to management of (incomplete) information, such as data exchange and reasoning about semi-structured data. In particular, the following three forms of negation have been advocated in the literature as important extensions of CQs: *safe negation* ($\mathrm{CQ}^{\neg s}$), *guarded negation* (GNCQ) and *inequalities* ($\mathrm{CQ}^{\neq}$). Recently, the DL community, with a similar motivation, have also taken a look at extensions of CQs with safe negation and inequalities (Rosati, 2007; Gutiérrez-Basulto et al., 2012, 2013).

A well-known fact from database theory is that answering CQs with negated atoms can be much harder than answering plain CQs; this is the case, for instance, for *open-world* query answering under integrity constraints (Rosati, 2006), query answering in the context of data exchange (Fagin et al., 2005) or query answering using materialised views (Abiteboul and Duschka, 1999). Rosati (2007) and Gutiérrez-Basulto et al. (2012) showed that the increase in the complexity is unfortunately dramatic in the OBDA setting: in striking contrast to the highly tractable AC^0 upper bound for data complexity of unions of CQs, the problems of answering unions of $\mathrm{CQs}^{\neq}$ and unions of $\mathrm{CQs}^{\neg s}$ turned out to be undecidable even over a very basic ontology language of *DL-Lite*$_{\mathrm{core}}$. The situation is similar for safe negation over $\mathcal{EL}$: answering unions of $\mathrm{CQs}^{\neg s}$ is undecidable. Remarkably, Klenke (2010) showed that in the language of $\mathcal{EL}$ extended with the empty concept ($\bot$) or, alternatively, under the standard unique name assumption (UNA), answering a *single* $\mathrm{CQ}^{\neq}$ is also undecidable. Interestingly, extending CQs and UCQs with negation has an effect not witnessed before in ontological query answering: there is a difference in the computational behaviour of *unions* of CQs and *single* CQs. In particular, a proof of undecidability of answering $\mathrm{UCQs}^{\neg s}$ (or $\mathrm{UCQs}^{\neq}$) cannot be straightforwardly adapted to the case of $\mathrm{CQs}^{\neg s}$ (respectively, $\mathrm{CQs}^{\neq}$). The intuitive reason is that, in the reduction of undecidable problems (such as the $\mathbb{N} \times \mathbb{N}$-tiling problem), each component of the union takes care of one of the several 'conditions' in the undecidable problem (colouring condition, matching condition, etc.), and it is not entirely obvious how to obtain a similar effect using a single query instead.

The addition of negation to CQs not only brings an increase in the computational complexity but also introduces further technical difficulties for the development of algorithmic approaches since negated atoms are not preserved under homomorphisms (Deutsch et al., 2008). As a consequence, to devise algorithms for answering $\mathrm{CQs}^{\neq}$ and $\mathrm{CQs}^{\neg s}$ over lightweight DLs we cannot directly use techniques based on the construction of the *canonical model* or the *chase* (Calvanese et al., 2007b; Kontchakov et al., 2010). Due to this reason, up to now, the only known results for answering CQs with negation over lightweight DLs are coNP-hardness for answering $\mathrm{CQs}^{\neq}$ and $\mathrm{CQs}^{\neg s}$ over *DL-Lite*$_{\mathrm{core}}$ (Rosati, 2007; Gutiérrez-Basulto et al., 2012), and the remarkable undecidability for $\mathrm{CQs}^{\neq}$ over $\mathcal{EL}_\bot$ (Klenke, 2010). Hence, the aim of this article is to sharpen the complexity picture for answering queries with safe negation and inequalities over lightweight ontologies.

In view of the additional complexity introduced by the presence of negative atoms in CQs, we also explore different syntactic restrictions on $\mathrm{CQs}^{\neg s}$ and $\mathrm{CQs}^{\neq}$ proposed in the literature. A robust approach to attain decidability for undecidable logics is to allow only for *guarded* quantification; this is the case, for example, of the guarded fragment of first-order logic and its extension with fixpoint operators (Andréka et al., 1998; Grädel and Walukiewicz, 1999). Inspired by these ideas, the notion of *guarded negation* was recently introduced in the context of decidable fragments of first-order logic, and later studied as an extension of positive existential queries (Bárány et al., 2011, 2012). In particular, Bárány et al. (2012) showed that, under the open-world semantics, answering first-order queries with guarded negation over frontier-guarded tuple-generating dependencies (fg-tgds) is decidable. Using this result as a departure point, we study the impact of guarded negation on answering $\mathrm{CQs}^{\neg s}$ over lightweight DLs. In another line, we look at restrictions on inequality atoms. Specifically, in the spirit of Arenas et al. (2011), we investigate possible ways of limiting the 'binding' of the variables occurring in inequalities. Finally, it has been observed that the *number* of negated atoms in a query can have an impact on the complexity (Klug, 1988; Fagin et al., 2005; Arenas et al., 2011; Bárány et al., 2012). So, we analyse the influence of this parameter on the complexity of answering CQs with negated atoms over lightweight DLs.

Summary of the Obtained Results. Our contributions can roughly be divided according to the two different forms of negation we explored: safe (including guarded) negation and inequalities; see Table 1 for a summary.

For CQs with safe negation, we first construct a $\mathrm{CQ}^{\neg s}$ with a *single* negated atom and an ontology in $\mathcal{ELI}_\bot$, an expressive member of the $\mathcal{EL}$ family, such that answering the query over the ontology amounts to checking whether the Turing machine encoded in the ontology terminates on the input encoded in the data. It follows that answering $\mathrm{CQs}^{\neg s}$ over $\mathcal{ELI}_\bot$ is undecidable even in the case where only the data is regarded as input (the ontology and the query are fixed, which corresponds to the data complexity). Having this result at hand, we describe how $\mathcal{ELI}_\bot$ concept inclusions can be translated into a *union* of $\mathrm{CQs}^{\neg s}$ over a *DL-Lite*$_{\mathrm{core}}$ ontology and thereby establish undecidability of answering unions of $\mathrm{CQs}^{\neg s}$ over *DL-Lite*$_{\mathrm{core}}$. We then show that the union of $\mathrm{CQs}^{\neg s}$ constructed in our undecidability proof can be replaced (preserving answers) by a single $\mathrm{CQ}^{\neg s}$ but at a price of adding a number of concept and *role* inclusions to the ontology. Consequently, answering $\mathrm{CQs}^{\neg s}$ over *DL-Lite*$^{\mathcal{H}}_{\mathrm{core}}$ is undecidable. (We note in passing that the transformation, however, is more general and applicable to a large class of unions of $\mathrm{CQs}^{\neg s}$ and $\mathrm{CQs}^{\neq}$ over ontologies in languages with role inclusions). Finally, we refine the borderline of undecidability for answering unions of $\mathrm{CQs}^{\neg s}$ and observe that the result holds for a fixed union of three $\mathrm{CQs}^{\neg s}$ over *DL-Lite*$_{\mathrm{core}}$ and a fixed union of two $\mathrm{CQs}^{\neg s}$ over $\mathcal{EL}_\bot$.

In the light of these negative results for safe negation we turn to a more restricted form of negation, guarded negation. Since frontier-guarded tuple-generating dependencies subsume $\mathcal{ELI}$ and CQs with guarded negation can express negative constraints in the ontology (concept and role inclusions with $\bot$), the results

		$\mathit{DL\text{-}Lite}_{core}$	$\mathit{DL\text{-}Lite}^{\mathcal{H}}_{core}$	$\mathcal{EL}_\bot$	$\mathcal{ELI}_\bot$
$\mathrm{UCQ}^{\neg s}$		undec. [Cor. 5]	undec.	undec. Rosati (2007)	undec.
$\mathrm{CQ}^{\neg s}$		coNP-hard [a]	undec. [Thm. 8]	coNP-hard [b]	undec. [Thm. 3]
UCQ / CQ with guarded negation	any	coNP ≥ [Lem. 11]	coNP	coNP	coNP ≤ Bárány et al. (2012)
	≤ 1 negation per CQ	P ≥ [Lem. 10]	P	P	P ≤ Bárány et al. (2012)
$\mathrm{UCQ}^{\neq}$		undec. [Thm. 14]	undec. Rosati (2007)	undec.	undec.
$\mathrm{CQ}^{\neq}$		coNP-hard [c]	undec. [Thm. 13]	undec. Klenke (2010)	undec.
UCQ / CQ with local inequalities	any	coNP-hard [Thm. 16] in coNExpTime	coNP-hard in coNExpTime [Thm. 20]	coNP-hard	coNP-hard
	≤ 1 inequality per CQ	P-hard [Thm. 15] in ExpTime	P-hard in ExpTime	P-hard	P-hard

[a] Thm. 9: undecidable for a union of *three* $\mathrm{CQs}^{\neg s}$, each with one negated atom.

[b] Cor. 6: undecidable for a union of *two* $\mathrm{CQs}^{\neg s}$, each with one negated atom (one of the components has guarded negation).

[c] Thm. 14: undecidable for a union of *three* $\mathrm{CQs}^{\neq}$, each with one inequality (two of the components have local inequalities).

Table 1: Summary of the data complexity results: C stands for C-complete; ≥ and ≤ with references indicate where, respectively, the lower and upper complexity bounds are established.

by Bárány et al. (2012) apply to both $\mathcal{ELI}_\bot$ and $\mathit{DL\text{-}Lite}^{\mathcal{H}}_{core}$: answering unions of CQs with guarded negation is in coNP in data complexity and in P if each of the constituent CQs contains at most one negated atom. We thus concentrate on establishing the matching lower complexity bounds: we construct an ontology with one negative concept inclusion (which belongs to all our DLs) and a CQ with one unary negated atom for P-hardness and a CQ with two unary negated atoms for coNP-hardness in data complexity.

The second form of negation in CQs we consider is inequalities. First, we prove that answering $\mathrm{CQs}^{\neq}$ over $\mathit{DL\text{-}Lite}^{\mathcal{H}}_{core}$ is undecidable. This result could be established using the method mentioned above: since answering unions of $\mathrm{CQs}^{\neq}$ over $\mathit{DL\text{-}Lite}_{core}$ is undecidable (Gutiérrez-Basulto et al., 2012), one could use additional concept and role inclusions to 'encode' the union into a single query. Following this route we would, however, obtain a query with multiple inequalities. Instead, we provide a more elaborate but direct proof using a $\mathrm{CQ}^{\neq}$ with a *single* inequality. Using the ideas developed for safe negation, we also establish undecidability of answering unions with at least three $\mathrm{CQs}^{\neq}$ over $\mathit{DL\text{-}Lite}_{core}$.

As the next step, we consider a restriction on the 'binding' of variables occurring in inequality atoms and identify a novel class of $\mathrm{CQs}^{\neq}$, *local* $\mathit{CQs}^{\neq}$, for which the query answering problem over $\mathit{DL\text{-}Lite}^{\mathcal{H}}_{core}$ ontologies is decidable. We also establish the lower complexity bounds over $\mathit{DL\text{-}Lite}_{core}$: P-hardness with one local inequality and coNP-hardness with two local inequalities; only coNP-hardness over $\mathit{DL\text{-}Lite}^{\mathcal{H}}_{core}$ was known (Rosati, 2007).

Related Work. Inequalities in the OBDA setting were first introduced by Calvanese et al. (1998, 2008a), who showed, in particular, that in contrast to answering CQs, answering $\mathrm{CQs}^{\neq}$ over a very expressive DL $\mathcal{DLR}$ is undecidable. Later, Rosati (2007) proved undecidability of answering CQs with safe negation and inequalities over a fairly inexpressive DL $\mathcal{AL}$. As discussed above, lightweight DLs were also analysed by Rosati (2007) and Gutiérrez-Basulto et al. (2012). A non-monotonic epistemic query language, EQL-Lite, was proposed by Calvanese et al. (2007a): it was shown that extensions of a number of query languages with negation over the epistemic S5 modality come with no increase in the complexity of query answering. In the context of Datalog±, ontology languages with equalities in the head of the rules have also been considered. Notably, Calì et al. (2012) investigated a restriction on the interaction of equalities (in the form of equality-generating dependencies) with Datalog± constraints that warranties decidability of the query answering problem. Recently, Hernich et al. (2013) presented extensions of Datalog± with non-monotonic negation under the well-founded semantics for normal logic programs.

It is worth noting that other extensions of conjunctive queries have also been considered in the framework of OBDA. In particular, Calvanese et al. (2008b) and Kostylev and Reutter (2013) studied aggregate queries; Bienvenu et al. (2013, 2014) and Kostylev et al. (2015) explored regular path queries (RPQs) and their further extensions.

Plan of the Article. In Section 2, we introduce the basics of our DLs and query languages. In Section 3, we focus on queries with safe negation. We begin by presenting our undecidability results for answering $\mathrm{CQs}^{\neg s}$ and then show the lower complexity bounds for answering CQs with guarded negation. In Section 4, we present our results on answering queries with inequalities. We first establish undecidability of answering $\mathrm{CQs}^{\neq}$ with one inequality over $\mathit{DL\text{-}Lite}^{\mathcal{H}}_{core}$. Then, in order to attain decidability, we introduce a syntactic restriction on inequalities, show the lower complexity bounds for this case and develop a decision procedure to prove decidability of the restricted problem.

This article is an extended and improved version of the conference paper (Gutiérrez-Basulto et al., 2013). Specifically, we extend our results along two directions: the range of DLs includes ontology languages of the $\mathcal{EL}$ family; the range of query languages includes CQs with guarded negation (Section 3.3) and local inequalities (which is a novel class that guarantees decidability, see Section 4.3). We also improve the presentation of the proofs, establish close connection between $\mathcal{ELI}_\bot$ concept inclusions and CQs with safe negation over *DL-Lite*$^{\mathcal{H}}_{\text{core}}$ ontologies and sharpen the undecidability boundary in terms of the number and structure of CQs with safe negation over *DL-Lite*$_{\text{core}}$ and extensions of $\mathcal{EL}$.

2. Preliminaries

2.1. Ontology Languages

Ontology languages use a vocabulary that comprises *individual names* $c_1, c_2, \dots$, *concept names* $A_1, A_2, \dots$, and *role names* $P_1, P_2, \dots$. Ontologies (TBoxes in the DL parlour) consist of concept and role inclusions built from concepts and roles using the constructors available in the ontology language, as described below.

Roles R and *basic concepts B* in *DL-Lite* (Artale et al., 2009) are defined by the following grammar:

$$R \quad ::= \quad P_i \quad | \quad P_i^-, \tag{1}$$
$$B \quad ::= \quad \top \quad | \quad A_i \quad | \quad \exists R. \tag{2}$$

Roles of the form P_i^- are called *inverse roles* and concepts of the form $\exists R$ are called *unqualified existential restrictions*. We identify R^- with P_i if $R = P_i^-$. A *TBox* in *DL-Lite*$_{\text{core}}$ is a finite set of *positive* and *negative concept inclusions* of the following form, respectively:

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

A TBox in *DL-Lite*$^{\mathcal{H}}_{\text{core}}$ can also contain a finite number of *positive* and *negative role inclusions* of the form

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

Concepts in $\mathcal{ELI}$ (Baader et al., 2005) are constructed from concept names by means of (*qualified*) *existential restrictions* and *intersection*; more precisely, they are defined by the following grammar:

$$C \quad ::= \quad \top \quad | \quad A_i \quad | \quad \exists R.C \quad | \quad C_1 \sqcap C_2,$$

where R is a role; see (1). An $\mathcal{ELI}_\bot$ *TBox* is a finite set of *positive* and *negative concept inclusions* of the form

$$C_1 \sqsubseteq C_2, \qquad C \sqsubseteq \bot.$$

An $\mathcal{ELI}$ TBox contains only positive inclusions. Existential restrictions of *DL-Lite* are a particular kind of existential restrictions in $\mathcal{ELI}$: $\exists R$ is a shortcut for $\exists R.\top$. Thus, every concept inclusion in *DL-Lite* is also a concept inclusion in $\mathcal{ELI}_\bot$.

Concepts in $\mathcal{EL}$ are defined in the same way as in $\mathcal{ELI}$ except that they cannot use inverse roles. An $\mathcal{EL}_\bot$ TBox is a set of positive and negative inclusions for $\mathcal{EL}$ concepts, while an $\mathcal{EL}$ TBox contains only positive inclusions.

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. The size $|\mathcal{T}|$ (respectively, $|\mathcal{A}|$) of a TBox $\mathcal{T}$ (respectively, an ABox $\mathcal{A}$) is the number of symbols required to write it down.

An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is a non-empty 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.

Remark 1. We *do not* adopt the *unique name assumption* (*UNA*), which requires $c_i^{\mathcal{I}} \neq c_j^{\mathcal{I}}$, for all distinct individual names c_i and c_j. Our results on safe and guarded negation in Section 3 clearly do not depend on this choice. For inequalities, the proofs in Section 4, which concern *DL-Lite*, are applicable to the case of UNA as well. Some undecidability and lower complexity bounds constructions, however, can be streamlined if the UNA is adopted (possible simplifications are indicated in the proofs). It is of interest to note that CQ$^{\neq}$ answering over $\mathcal{EL}$ is tractable in general (Rosati, 2007) and undecidable if the UNA is adopted (Klenke, 2010). In the *DL-Lite* family, on the other hand, the UNA does not make such a drastic effect because the languages have negative concept inclusions (which can express a sort of local UNA).

The interpretation function $\cdot^{\mathcal{I}}$ is extended to roles and complex concepts in the standard way:

$$\begin{aligned}
(P_i^-)^{\mathcal{I}} &= \{ (d', d) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (d, d') \in P_i^{\mathcal{I}} \}, \\
\top^{\mathcal{I}} &= \Delta^{\mathcal{I}}, \\
(\exists R.C)^{\mathcal{I}} &= \{ d \in \Delta^{\mathcal{I}} \mid \text{there is } d' \in C^{\mathcal{I}} \text{ with } (d, d') \in R^{\mathcal{I}} \}, \\
(C_1 \sqcap C_2)^{\mathcal{I}} &= C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}.
\end{aligned}$$

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

$$\begin{aligned}
\mathcal{I} \models C_1 \sqsubseteq C_2 \quad &\text{iff} \quad C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}, \\
\mathcal{I} \models C \sqsubseteq \bot \quad &\text{iff} \quad C^{\mathcal{I}} = \emptyset, \\
\mathcal{I} \models R_1 \sqsubseteq R_2 \quad &\text{iff} \quad R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}}, \\
\mathcal{I} \models R_1 \sqcap R_2 \sqsubseteq \bot \quad &\text{iff} \quad R_1^{\mathcal{I}} \cap R_2^{\mathcal{I}} = \emptyset, \\
\mathcal{I} \models A_i(c_j) \quad &\text{iff} \quad c_j^{\mathcal{I}} \in A_i^{\mathcal{I}}, \\
\mathcal{I} \models P_i(c_j, c_k) \quad &\text{iff} \quad (c_j^{\mathcal{I}}, c_k^{\mathcal{I}}) \in P_i^{\mathcal{I}}.
\end{aligned}$$

A KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is *consistent* (*satisfiable*) if there is an interpretation $\mathcal{I}$ satisfying all inclusions in $\mathcal{T}$ and assertions in $\mathcal{A}$. 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* $\mathcal{K}$ (as well as of $\mathcal{T}$ and $\mathcal{A}$). We also write $\mathcal{T} \models \alpha$ if a concept or role inclusion α is satisfied in all models of $\mathcal{T}$; in this case we say that α is *entailed* by $\mathcal{T}$.

Remark 2. In *DL-Lite*$^{\mathcal{H}}_{\text{core}}$ TBoxes, we will often use concept inclusions of the form $B \sqsubseteq C$, where B is a basic concept and C an $\mathcal{ELI}$ concept. This is justified because, given such a concept inclusion, one can construct (in polynomial time) a *DL-Lite*$^{\mathcal{H}}_{\text{core}}$ TBox $\mathcal{T}$ which is a model conservative extension of α: that is,

- $\mathcal{T} \models \alpha$ and,
- conversely, every model of α can be extended to a model of $\mathcal{T}$ by giving an interpretation to the fresh names in $\mathcal{T}$.

Indeed, a concept inclusion of the form $B \sqsubseteq C_1 \sqcap C_2$ is equivalent to two concept inclusions $B \sqsubseteq C_i$, for $i = 1, 2$; and a concept inclusion of the form $B \sqsubseteq \exists R.C$ can be replaced by two concept inclusions $B \sqsubseteq \exists R_C$, $\exists R_C^- \sqsubseteq C$ and a role inclusion $R_C \sqsubseteq R$; for more details see, e.g., (Artale et al., 2009). Therefore, the presence of concept inclusions of the form $B \sqsubseteq C$ does not affect any of our results on *DL-Lite*$^{\mathcal{H}}_{\text{core}}$.

Note, however, that such a shortcut is not available in *DL-Lite*$_{\text{core}}$ because it contains no role inclusions.

2.2. Query Languages

A *conjunctive query* (CQ) $\boldsymbol{q}(\boldsymbol{x})$ is a first-order 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 (concept and role) atoms and negated atoms, such that each variable occurs in at least one positive atom. A CQ$^{\neg_1 s}$ is a CQ$^{\neg s}$ with at most one negative atom. A CQ$^{\neg s}$ is said to be a *conjunctive query with guarded negation* (GNCQ) if, for each negative atom, the query contains a positive atom, a *guard*, containing all the variables of the negative atom (thus, in contrast to general CQs$^{\neg s}$, all variables of any negative atom in a GNCQ must occur in the *same* positive atom).

A *conjunctive query with inequalities* (CQ$^{\neq}$) is an expression of the form $\exists \boldsymbol{y}\, \varphi(\boldsymbol{x}, \boldsymbol{y})$, where φ is a conjunction of positive atoms and inequalities $t \neq t'$, for terms t and t'.

A *union of conjunctive queries* (UCQ) is a disjunction of CQs that share the same tuple of answer variables; a UCQ$^{\neg s}$ and UCQ$^{\neq}$ are defined accordingly. Without loss of generality, in this article we always assume that the tuples of quantified variables in UCQ components are pairwise disjoint.

Given a query $\boldsymbol{q}(\boldsymbol{x})$, we usually write $\boldsymbol{q}$ if $\boldsymbol{x}$ is clear from the context (or irrelevant). The size $|\boldsymbol{q}|$ of $\boldsymbol{q}$ is the number of symbols required to write it down.

We will often regard a CQ $\boldsymbol{q}$ (possibly, with negative atoms) as a set of its atoms and assume that $\boldsymbol{q}$ contains $P_i^-(t, t')$ if it contains $P_i(t', t)$ (and similarly for the negative atoms). We extend this convention to basic concepts and assume that $\boldsymbol{q}$ contains unary 'atoms' $B(t)$ and $B'(t')$ if it contains $R(t, t')$, where $B = \exists R$ and $B' = \exists R^-$. We will also associate with $\boldsymbol{q}$ an undirected graph, called the *primal graph of* $\boldsymbol{q}$, whose vertices are the terms of $\boldsymbol{q}$ and which has an edge between t and t' if and only if the query contains a positive atom of the form $R(t, t')$ (note that the negative atoms are not taken into account).

A query $\boldsymbol{q}(\boldsymbol{x})$ is called *Boolean* if $\boldsymbol{x}$ is empty. A Boolean CQ$^{\neg s}$ $\boldsymbol{q}$ is *tree-shaped* if does not contain individuals as terms and its primal graph is a tree (a *tree* is any connected undirected graph without simple cycles).

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 a 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 $\pi(x_i) = c_i^{\mathcal{I}}$ (in this case π is also a match for the Boolean query $\boldsymbol{q}(\boldsymbol{c})$ 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 $\mathcal{K}$. For a Boolean query $\boldsymbol{q}$, if there is a match for $\boldsymbol{q}$ in every model of $\mathcal{K}$, that is, if the empty tuple is a certain answer, then we say that the certain answer is *yes* (or that $\boldsymbol{q}$ has a positive answer over $\mathcal{K}$).

2.3. Canonical Interpretation for DL-Lite$^{\mathcal{H}}_{\text{core}}$

Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a *DL-Lite*$^{\mathcal{H}}_{\text{core}}$ knowledge base. We can consider the ABox $\mathcal{A}$ as an interpretation and extend the notation for the satisfaction relation $\models$ to roles: $\mathcal{A} \models R(c, c')$ abbreviates $P(c, c') \in \mathcal{A}$ if $R = P$ and $P(c', c) \in \mathcal{A}$ if $R = P^-$. Similarly, for a basic concept B, we use $\mathcal{A} \models B(c)$ as a shortcut for $A(c) \in \mathcal{A}$ if $B = A$ and for $\mathcal{A} \models R(c, c')$, for some c', if $B = \exists R$.

The *canonical interpretation* $\mathcal{C}_{\mathcal{K}}$ of $\mathcal{K}$ is an interpretation with the domain $\Delta^{\mathcal{C}_{\mathcal{K}}}$ comprising all elements of the form $d_{cR_1 \dots R_n}$, for an individual name c and roles $R_1, \dots, R_n$, $n \geq 0$, such that

- if $n \geq 1$ then there is a basic concept B with $\mathcal{A} \models B(c)$ and $\mathcal{T} \models B \sqsubseteq \exists R_1$ but $\mathcal{A} \not\models R(c, c')$, for all c' and R with $\mathcal{T} \models R \sqsubseteq R_1$;
- $\mathcal{T} \models \exists R_{i-1}^- \sqsubseteq \exists R_i$ but $\mathcal{T} \not\models R_{i-1}^- \sqsubseteq R_i$, for each i, $1 < i \leq n$,

and the interpretation function $\cdot^{\mathcal{C}_{\mathcal{K}}}$ defined for individual names c, concept names A and role names P as follows:

$$\begin{aligned}
c^{\mathcal{C}_{\mathcal{K}}} &= d_c, \\
A^{\mathcal{C}_{\mathcal{K}}} &= \{\, d_c \mid \mathcal{A} \models B(c) \text{ and } \mathcal{T} \models B \sqsubseteq A \,\} \cup \\
&\quad\ \{\, d_{cR_1 \dots R_n} \mid n \geq 1,\ \mathcal{T} \models \exists R_n^- \sqsubseteq A \,\}, \\
P^{\mathcal{C}_{\mathcal{K}}} &= \{\, (d_{c_1}, d_{c_2}) \mid \mathcal{A} \models R(c_1, c_2) \text{ and } \mathcal{T} \models R \sqsubseteq P \,\} \cup \\
&\quad\ \{\, (d_{cR_1 \dots R_{n-1}}, d_{cR_1 \dots R_n}) \mid n \geq 1,\ \mathcal{T} \models R_n \sqsubseteq P \,\} \cup \\
&\quad\ \{\, (d_{cR_1 \dots R_n}, d_{cR_1 \dots R_{n-1}}) \mid n \geq 1,\ \mathcal{T} \models R_n \sqsubseteq P^- \,\}.
\end{aligned}$$

It is well-known (see e.g., Artale et al. 2009) that a *DL-Lite*$^{\mathcal{H}}_{\text{core}}$ knowledge base $\mathcal{K}$ is consistent if and only if its canonical interpretation satisfies all negative concept and role inclusions in the TBox. Moreover, if $\mathcal{K}$ is consistent then the canonical interpretation is a *universal* model in the sense that it can be homomorphically mapped to any other model of $\mathcal{K}$. This means, in particular, that $\mathcal{C}_{\mathcal{K}}$ provides all the information required for computing certain answers to any CQ or UCQ $\boldsymbol{q}(\boldsymbol{x})$ over $\mathcal{K}$:

$$\mathcal{K} \models \boldsymbol{q}(\boldsymbol{c}) \qquad \text{iff} \qquad \mathcal{C}_{\mathcal{K}} \models \boldsymbol{q}(\boldsymbol{c}).$$

The analogous claim fails for queries with negative atoms because only sentences equivalent to positive existential formulas are preserved under homomorphisms (Homomorphism Preservation Theorem; for more recent results, see, e.g., Rossman

2008). In the sequel we shall see that it has a dramatic effect on the complexity of query answering.

Finally, we note that canonical interpretations could similarly be defined in $\mathcal{ELI}_\bot$ and its fragments but they are not needed in this article.

2.4. Data Complexity

In OBDA scenarios the size of the query and the TBox (ontology) is usually much smaller than the size of the ABox (data). This is why we explore the *data complexity* (Vardi, 1982) 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 TBox and $\boldsymbol{q}(\boldsymbol{x})$ a query in one of the classes defined above. We are interested in the following family of problems:

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

3. Answering CQs with Safe and Guarded Negation

In this section we study queries with safe and guarded negation. Rosati (2007) established initial results on the complexity of answering such queries. Specifically, it was shown that answering $\mathrm{CQs}^{\neg s}$ over knowledge bases that admit so-called saturated models (and, in particular, contain no negative inclusions) has the same complexity as answering CQs; this result thus applies to $\mathcal{EL}$, $\mathcal{ELI}$ and the RDFS fragment of *DL-Lite*$_{\mathit{core}}$. It was also shown that, in contrast, answering *unions* of CQs with safe negation over *DL-Lite*$^{\mathcal{H}}_{\mathit{core}}$ and $\mathcal{EL}$ is undecidable. The proofs of the undecidability results regard, along with the ABox, both the TBox and the query as part of the problem input, which corresponds to the *combined complexity* (Vardi, 1982). We begin this section by a transparent reduction of the halting problem for deterministic Turing machines to answering a *single* fixed Boolean $\mathrm{CQ}^{\neg_1 s}$ over $\mathcal{ELI}_\bot$ KBs with a fixed TBox (Theorem 3), which proves undecidability of $\mathrm{CQ}^{\neg_1 s}$ answering over $\mathcal{ELI}_\bot$ even in data complexity. Then, in Lemma 4 we establish a close correspondence between $\mathcal{ELI}_\bot$ TBoxes and *unions* of $\mathrm{CQs}^{\neg_1 s}$ over *DL-Lite*$_{\mathit{core}}$ TBoxes, which in particular implies undecidability of answering unions of $\mathrm{CQs}^{\neg_1 s}$ over *DL-Lite*$_{\mathit{core}}$ in data complexity (Corollary 5). Another result following from Theorem 3 is undecidability of answering unions of two $\mathrm{CQs}^{\neg_1 s}$ over $\mathcal{EL}_\bot$ (Corollary 6); the case of one $\mathrm{CQ}^{\neg s}$ is, however, left open.

We then proceed to show, in Lemma 7, that the union of tree-shaped $\mathrm{CQs}^{\neg s}$ in the proof of Corollary 5 can be replaced by a *single* $\mathrm{CQ}^{\neg s}$ and a number of role inclusions. Thus, we extend the undecidability result to the problem of answering CQs with safe negation over *DL-Lite*$^{\mathcal{H}}_{\mathit{core}}$. We point out that the transformation of Lemma 7 is general and may be of wider interest; in particular, it is also applicable to plain CQs and CQs with inequalities.

In Theorem 9, we explore the limits of undecidability and prove that answering unions of *three* $\mathrm{CQs}^{\neg_1 s}$ over *DL-Lite*$_{\mathit{core}}$ (without role inclusions) is undecidable. We leave the case of unions with one or two disjuncts as an open problem.

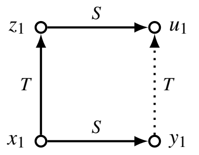


Figure 1: Completing the square with Boolean $\mathrm{CQ}^{\neg_1 s}$ (3).

Finally, we turn to the problem of answering CQs with guarded negation, which is known (Bárány et al., 2012) to be decidable and in coNP in data complexity (in P for GNCQs with one negated atom) over lightweight DLs, and establish matching lower bounds over a *DL-Lite*$_{\mathit{core}}$ TBox with a single negative concept inclusion.

3.1. Safe Negation: Undecidability over $\mathcal{ELI}_\bot$

Our undecidability results are obtained by reduction of the halting problem for deterministic Turing machines. The key observation is that a configuration of a Turing machine (that is, the content of the tape, the current state and the position of the head at a particular step of a computation) can be written down on a sequence of domain elements with a role, T, pointing to the representation of the next cell of the tape. Then a computation of the Turing machine can be thought of as a two-dimensional grid, where another role, S, points to the representation of the cell in the successive configuration.

In order to establish the required two-dimensional grid, we are going to use the following Boolean $\mathrm{CQ}^{\neg_1 s}$ $\boldsymbol{q}_1$:

$$\exists x_1, y_1, z_1, u_1\, (S(x_1, y_1) \land T(x_1, z_1) \land S(z_1, u_1) \land \neg T(y_1, u_1)). \quad (3)$$

It can be readily seen that in any interpretation $\mathcal{I}$ where $\boldsymbol{q}_1$ has a negative answer, that is, $\mathcal{I} \not\models \boldsymbol{q}_1$, for every four elements forming the three sides of a square, there is a T-edge that completes the square, as shown in Fig. 1. This property can also be expressed by the following first-order sentence:

$$S(x_1, y_1) \land T(x_1, z_1) \land S(z_1, u_1) \to T(y_1, u_1), \qquad (3^\neg)$$

where all variables are universally quantified. Indeed, sentence ($3^\neg$) holds in every model of a KB $\mathcal{K}$ if and only if query (3) has a negative answer over $\mathcal{K}$. In other words, sentence ($3^\neg$) is equivalent to the negation of the query. In the sequel, we will often prefer to represent Boolean CQs with safe negation (as well as with inequalities) in their *negated form*, that is, as implications with all variables universally quantified.

Once the grid has been established, we can use the expressive description logic $\mathcal{ELI}_\bot$ to ensure that the elements of the grid encode successive configurations in a computation of a given deterministic Turing machine. This observation leads us to our first undecidability result.

Theorem 3. *There are a Boolean* $\mathrm{CQ}^{\neg_1 s}$ $\boldsymbol{q}$ *and an* $\mathcal{ELI}_\bot$ *TBox* $\mathcal{T}$ *such that the problem* CertainAnswers($\boldsymbol{q}, \mathcal{T}$) *is undecidable.*

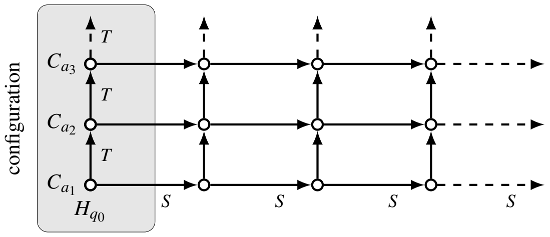


Figure 2: Encoding computations of a Turing machine.

Proof. Given a deterministic 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 $\mathcal{A}_{\boldsymbol{w}}$ if and only if $(\mathcal{T}, \mathcal{A}_{\boldsymbol{w}}) \not\models \boldsymbol{q}$; note that neither $\boldsymbol{q}$ nor $\mathcal{T}$ depends on $\boldsymbol{w}$. By 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 shall 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 is a set of *states*, $q_0 \in Q$ and $q_1 \in Q$ are an *initial* and *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 content 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.

More precisely, we use the following signature:

- role T points to the representation of the next cell on the tape (within the same configuration) and role S points to the representation of the same cell in the successive configuration;

- concepts C_a, for $a \in \Gamma$, encode the contents of cells in the sense that a domain element belongs to the interpretation of C_a if the cell contains symbol a;

- concepts H_q, for $q \in Q$, indicate both the current state and the position of the head: a domain element belongs to the interpretation of H_q if the cell is under the head and the machine is in state q;

- concept $H_\emptyset$ marks all other cells on the tape (that is, cells that are not under the head of the machine);

- concepts D^q_σ and D_σ, for $q \in Q$ and $\sigma \in \{-1, +1\}$, propagate the head and no-head markers backwards and forwards along the tape, respectively;

- concept I is required to ensure that the tape is initially blank beyond the input word.

The grid is illustrated in Fig. 2, where the nodes are domain elements and the grey rectangle highlights an initial configuration: initially, the infinite tape contains the input padded with $\llcorner\!\lrcorner$ and the head is positioned over the first cell in state q_0.

Let $\boldsymbol{q}$ be the Boolean CQ$^{\neg_1 s}$ given by (3) and let $\mathcal{T}$ be an $\mathcal{ELI}_\bot$ TBox containing the following concept inclusions:

$$
\begin{aligned}
H_q \sqcap C_a &\sqsubseteq \exists S.(C_{a'} \sqcap D^{q'}_\sigma), && \text{for } \delta(q, a) = (q', a', \sigma), && (4)\\
H_\emptyset \sqcap C_a &\sqsubseteq \exists S.C_a, && \text{for } a \in \Gamma, && (5)\\
H_q &\sqsubseteq D_{-1} \sqcap D_{+1}, && \text{for } q \in Q, && (6)\\
\exists T.D^q_{-1} &\sqsubseteq H_q, && \text{for } q \in Q, && (7)\\
\exists T^-.D^q_{+1} &\sqsubseteq H_q, && \text{for } q \in Q, && (8)\\
\exists T.D_{-1} &\sqsubseteq H_\emptyset \sqcap D_{-1}, && && (9)\\
\exists T^-.D_{+1} &\sqsubseteq H_\emptyset \sqcap D_{+1}, && && (10)\\
I &\sqsubseteq \exists T.(I \sqcap C_{\llcorner\!\lrcorner}), && && (11)\\
H_{q_1} &\sqsubseteq \bot. && && (12)
\end{aligned}
$$

For every input $\boldsymbol{w} = a_1 \dots a_n \in \Gamma^*$, we take the following ABox $\mathcal{A}_{\boldsymbol{w}}$ with individual names $c_1, \dots, c_n$:

$$H_{q_0}(c_1), \quad C_{a_i}(c_i) \text{ and } T(c_i, c_{i+1}), \text{ for } 1 \leq i < n, \quad I(c_n).$$

We claim that $(\mathcal{T}, \mathcal{A}_{\boldsymbol{w}}) \not\models \boldsymbol{q}$ if and only if M does not accept $\boldsymbol{w}$.

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 and (11), there exists an infinite sequence of (not necessarily distinct) domain elements $d_1, d_2, \dots$ that encode the initial configuration in the sense that $(d_i, d_{i+1}) \in T^{\mathcal{I}}$ for all $i \geq 1$, $d_1 \in H^{\mathcal{I}}_{q_0}$, $d_i \in C^{\mathcal{I}}_{a_i}$, for each $1 \leq i \leq n$, and $d_i \in C^{\mathcal{I}}_{\llcorner\!\lrcorner}$ for all $i > n$. By (6) and (10), $d_i \in H^{\mathcal{I}}_\emptyset$ for all $i > 1$. Then, by (4) and (5), there exist elements $d'_1, d'_2, \dots$ such that $(d_i, d'_i) \in S^{\mathcal{I}}$. Since $\mathcal{I} \not\models \boldsymbol{q}$, they form another T-connected sequence, that is, $(d'_i, d'_{i+1}) \in T^{\mathcal{I}}$ for all i, which represents the second configuration of the computation. Indeed, by (5), the symbols in the cells not under the head are preserved by the transition. On the other hand, by (4), the symbol in the cell under the head is changed according to the transition function δ of M, and the new head position and state are recorded in the concept $D^{q'}_\sigma$. By (7) and (8), the recorded head position and the state are passed onto the correct cell. Then, by (6), the domain element representing the head, say, d'_k, belongs to $D^{\mathcal{I}}_{+1}$, whence, by (10), all d'_i with $i > k$ belong to $D^{\mathcal{I}}_{+1}$ and $H^{\mathcal{I}}_\emptyset$. Similarly, by (6) and (9), $d'_i \in H^{\mathcal{I}}_\emptyset$, for all $i < k$. Therefore, again, all cells that are not under the head belong to $H^{\mathcal{I}}_\emptyset$. By the same argument, there exists a respective sequence of elements for each configuration of the computation. Finally, (12) guarantees that the accepting state never occurs in the computation, that is, M does not accept $\boldsymbol{w}$.

Conversely, if the computation of M on $\boldsymbol{w}$ is non-accepting then we can encode it by an infinite two-dimensional grid interpretation satisfying $(\mathcal{T}, \mathcal{A}_{\boldsymbol{w}})$ but not $\boldsymbol{q}$.

Since the problem of deciding whether a given deterministic machine accepts a given input is undecidable, we obtain the claim of the theorem. ❑

Unlike $\mathcal{ELI}_\bot$, *DL-Lite*$_{core}$ does not have qualified existential restrictions and so, we cannot propagate information about the contents of the tape and the position of the head using concept inclusions (4)–(5) and (7)–(11). Nevertheless, we show that

$\mathcal{ELI}_\bot$ concept inclusions can be ‘encoded’ over *DL-Lite*$_{\mathrm{core}}$ with the help of additional concept inclusions and *unions* of CQs$^{\neg s}$.

We illustrate the main idea of our second undecidability result for answering unions of CQs$^{\neg 1 s}$ over *DL-Lite*$_{\mathrm{core}}$ on two examples. Consider first the following Boolean CQ$^{\neg 1 s}$ $\boldsymbol{q}_2$:

$$\exists x_2, y_2\, (T(x_2, y_2) \land \neg R(y_2, x_2)), \tag{13}$$

or in negated form:

$$T(x_2, y_2) \to R(y_2, x_2). \tag{13$^\neg$}$$

It can be easily seen that $\mathcal{I} \not\models \boldsymbol{q}_2$ if and only if $\mathcal{I} \models T^- \sqsubseteq R$, for any interpretation $\mathcal{I}$. Thus, one can think of a role inclusion as a negated CQ$^{\neg 1 s}$. Then, by Remark 2, we can encode any $\mathcal{ELI}_\bot$ concept inclusion of the form $B \sqsubseteq C$, for a basic concept B, as a *DL-Lite*$_{\mathrm{core}}$ TBox and a Boolean UCQ$^{\neg 1 s}$. Note that a set of role inclusions is true in an interpretation $\mathcal{I}$ if and only if none of the corresponding queries have a positive answer in $\mathcal{I}$, that is, their *union* has a negative answer in $\mathcal{I}$.

For our second example, consider an $\mathcal{ELI}$ concept inclusion $B_1 \sqcap \exists R.B_2 \sqsubseteq A$. Evidently, this concept inclusion is satisfied in $\mathcal{I}$ if and only if the following Boolean CQ$^{\neg 1 s}$ has a negative answer in $\mathcal{I}$:

$$\exists x, y\, (B_1(x) \land R(x, y) \land B_2(y) \land \neg A(x)).$$

So, we can also think of concept inclusions of the form $C \sqsubseteq A$, for an $\mathcal{ELI}$ concept C and a concept name A, simply as (tree-shaped) Boolean queries with one safe negation.

Taking stock, any $\mathcal{ELI}_\bot$ concept inclusion can be encoded as a *DL-Lite*$_{\mathrm{core}}$ TBox and a Boolean UCQ$^{\neg s}$, and we thus arrive at the following lemma.

Lemma 4. *For any $\mathcal{ELI}_\bot$ TBox $\mathcal{T}$, one can construct a DL-Lite$_{\mathrm{core}}$ TBox $\mathcal{T}'$ and a Boolean UCQ$^{\neg 1 s}$ $\boldsymbol{q}'$ such that*

- *every model $\mathcal{I}$ of $\mathcal{T}'$ with $\mathcal{I} \not\models \boldsymbol{q}'$ is also a model of $\mathcal{T}$, and*
- *every model of $\mathcal{T}$ can be extended to a model $\mathcal{I}$ of $\mathcal{T}'$ with $\mathcal{I} \not\models \boldsymbol{q}'$ by interpreting fresh names in $\mathcal{T}'$.*

As a corollary of Theorem 3 and Lemma 4 we immediately obtain undecidability of answering *unions* of CQs$^{\neg 1 s}$ over *DL-Lite*$_{\mathrm{core}}$ KBs.

Corollary 5. *There is a Boolean UCQ$^{\neg 1 s}$ $\boldsymbol{q}$ and a DL-Lite$_{\mathrm{core}}$ TBox $\mathcal{T}$ such that* CertainAnswers$(\boldsymbol{q}, \mathcal{T})$ *is undecidable.*

Observe that the TBox in the proof of Theorem 3 belongs to $\mathcal{EL}$ except for concept inclusions (8), (10) and (12). Consider now a UCQ$^{\neg 1 s}$ comprising $\exists x\, H_{q_1}(x)$ and queries (3) and (13). By replacing the inverse role T^- in (8) and (10) by R and removing the negative concept inclusion (12), we can strengthen the undecidability result for UCQ$^{\neg s}$ over $\mathcal{EL}$ KBs established by Rosati (2007).

Corollary 6. (i) *There are a union $\boldsymbol{q}$ of two Boolean CQs$^{\neg 1 s}$ and an $\mathcal{EL}_\bot$ TBox $\mathcal{T}$ such that* CertainAnswers$(\boldsymbol{q}, \mathcal{T})$ *is undecidable.*

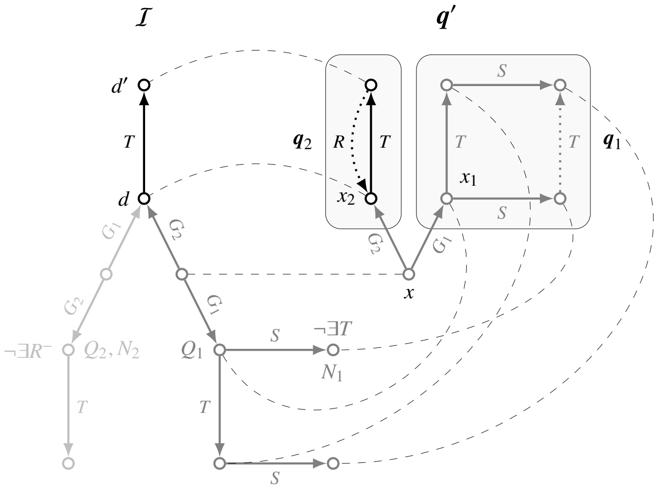


Figure 3: Matching CQ$^{\neg s}$ $\boldsymbol{q}'$ obtained from $\boldsymbol{q}_1 \lor \boldsymbol{q}_2$ in the extended model.

(ii) *There are a union $\boldsymbol{q}$ of a Boolean CQ and two CQs$^{\neg 1 s}$, and an $\mathcal{EL}$ TBox $\mathcal{T}$ such that* CertainAnswers$(\boldsymbol{q}, \mathcal{T})$ *is undecidable.*

(iii) *There are a union $\boldsymbol{q}$ of a Boolean CQ and a CQ$^{\neg 1 s}$, and an $\mathcal{ELI}$ TBox $\mathcal{T}$ such that* CertainAnswers$(\boldsymbol{q}, \mathcal{T})$ *is undecidable.*

The last result is in stark contrast to P-completeness of answering *single* CQs$^{\neg s}$ (Rosati, 2007) and unions of CQs over $\mathcal{ELI}$ TBoxes (Ortiz et al., 2006).

3.2. From UCQs to CQs: the Case of DL-Lite$^{\mathcal{H}}_{\mathrm{core}}$

We now proceed to show that under rather mild restrictions, any union of tree-shaped Boolean CQs$^{\neg s}$ can be transformed into a single Boolean CQ$^{\neg s}$ that has the same answers over knowledge bases with TBoxes *extended* by a number of concept and role inclusions. This will allow us to obtain undecidability of answering a *single* CQ$^{\neg s}$ over *DL-Lite*$^{\mathcal{H}}_{\mathrm{core}}$ (in contrast to Corollary 5, which holds for the language without role inclusions).

We illustrate the transformation by considering a Boolean UCQ$^{\neg s}$ $\boldsymbol{q}$ comprising the two queries from Section 3.1:

$$\begin{aligned} \boldsymbol{q}_1 &= \exists x_1, y_1, z_1, u_1\, (S(x_1, y_1) \land T(x_1, z_1) \land S(z_1, u_1) && (3)\\ &\qquad\qquad \land \neg T(y_1, u_1)), \\ \boldsymbol{q}_2 &= \exists x_2, y_2\, (T(x_2, y_2) \land \neg R(y_2, x_2)); && (13) \end{aligned}$$

these queries are also given in negated form by (3$^\neg$) and (13$^\neg$), respectively. Note first that the sets of variables in $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ are disjoint, and therefore, we can merge them into a single CQ$^{\neg s}$ without introducing a connection between the primal graphs of the constituents. Then, we take a fresh variable x and consider a Boolean CQ$^{\neg s}$ $\boldsymbol{q}'$ that consists of all the atoms of $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ together with $G_1(x, x_1)$ and $G_2(x, x_2)$, where G_1 and G_2 are fresh role names; see Fig. 3 on the right.

The resulting CQ$^{\neg s}$ $\boldsymbol{q}'$ is in general not equivalent to $\boldsymbol{q}$. However, we can guarantee that, for any TBox $\mathcal{T}$ satisfying some mild restrictions (to be defined below), there is a TBox $\mathcal{T}'$ such that the union $\boldsymbol{q}$ has the same answer over $(\mathcal{T}, \mathcal{A})$ as $\boldsymbol{q}'$ over $(\mathcal{T} \cup \mathcal{T}', \mathcal{A})$. The extension TBox $\mathcal{T}'$ is constructed in such a way that from any model $\mathcal{I}$ of $(\mathcal{T}, \mathcal{A})$ we can obtain a model $\mathcal{I}'$ of $(\mathcal{T} \cup \mathcal{T}', \mathcal{A})$ that coincides with $\mathcal{I}$ on $\Delta^{\mathcal{I}}$ and satisfies the following properties:

1. the interpretation of a special concept name D contains every domain element in $\mathcal{I}$;
2. for each CQ$^{\neg s}$ $\boldsymbol{q}_i$ in the union $\boldsymbol{q}$ and every d in the interpretation of D, there is a map that sends x_i to d and matches *all* atoms (including the negative ones) of the merged $\boldsymbol{q}'$ *except*, possibly, the atoms of $\boldsymbol{q}_i$.

For example, consider a model $\mathcal{I}$ of $\mathcal{T}$ with a single T-edge (d, d'); see the black arrow in Fig. 3 on the left. According to Item 1, the extended TBox should guarantee that both d and d' belong to the interpretation of D in the model $\mathcal{I}'$ of $\mathcal{T} \cup \mathcal{T}'$. By Item 2, it should also guarantee that d has the dark-grey fragment attached to it to match all atoms of $\boldsymbol{q}'$ but $\boldsymbol{q}_1$ and the light-grey fragment to match all atoms of $\boldsymbol{q}'$ but $\boldsymbol{q}_2$ (d' should also be in the interpretation of D and, hence, have similar fragments in $\mathcal{I}'$, but they are not depicted to reduce clutter). Moreover, it should be clear that $\boldsymbol{q}'$ has a positive answer in $\mathcal{I}'$ if and only if either $\boldsymbol{q}_1$ has a positive answer in $\mathcal{I}$ (the rest of $\boldsymbol{q}'$ is matched by the light-grey fragment) *or* $\boldsymbol{q}_2$ has a positive answer in $\mathcal{I}$ (the rest of $\boldsymbol{q}'$ is matched by the dark-grey fragment), which is the same as their union, $\boldsymbol{q}$, having a positive answer in $\mathcal{I}$.

The fragments required to match the positive atoms of $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ can easily be generated, for example, by the *DL-Lite*$^{\mathcal{H}}_{\text{core}}$ concept inclusions

$$D \sqsubseteq \exists G_2^-.\exists G_1.Q_1, \qquad Q_1 \sqsubseteq \exists T.\exists S \sqcap \exists S.N_1, \tag{14}$$

$$D \sqsubseteq \exists G_1^-.\exists G_2.Q_2, \qquad Q_2 \sqsubseteq \exists T \sqcap N_2, \tag{15}$$

where Q_1, N_1, Q_2 and N_2 are fresh concept names (see Fig. 3). We also need the following negative concept inclusions to ensure that the negative atoms of $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ can always be matched in the respective fragments of the model generated by the positive inclusions (14)–(15):

$$N_1 \sqcap \exists T \sqsubseteq \bot \qquad \text{and} \qquad N_2 \sqcap \exists R^- \sqsubseteq \bot. \tag{16}$$

We now generalise the intuition above and show that we can apply this transformation to a union of an arbitrary number of tree-shaped CQs$^{\neg s}$.

It should be clear that any tree-shaped Boolean CQ$^{\neg s}$ gives rise to a *DL-Lite*$^{\mathcal{H}}_{\text{core}}$ TBox similar to (14)–(16). To make sure that the negative concept inclusions of the form (16) are not inconsistent with the positive inclusions of the form (14)–(15), we require an additional definition. We say that a variable z in a CQ$^{\neg s}$ $\boldsymbol{q}$ is $\mathcal{T}$*-loose* (or *loose*, if $\mathcal{T}$ is clear from the context) in case $\mathcal{T} \not\models B_1 \sqsubseteq B_2$, for each pair of atoms $B_1(z)$ and $\neg B_2(z)$ in $\boldsymbol{q}$ (to simplify notation, the B_i refer here to basic concepts; similarly to positive atoms, the query is assumed to contain $\neg\exists P(z_1)$ and $\neg\exists P^-(z_2)$ if it contains $\neg P(z_1, z_2)$). For instance, in the example above, variable y_1 is loose in $\boldsymbol{q}_1$ provided that the original TBox does not entail $\exists S^- \sqsubseteq \exists T$; in other words, if (the interpretation of) $\exists S^-$ may contain a domain element that is not in $\exists T$—otherwise the first negative inclusion in (16) would imply emptiness of D with the extended TBox (indeed, the S-successor of an element in Q_1 would have to belong to $\exists S^-$ and N_1, which are subsets of the disjoint $\exists T$ and N_1, respectively). Also, u_1 is loose in $\boldsymbol{q}_1$ if the original TBox does not entail $\exists S^- \sqsubseteq \exists T^-$; similarly, both x_2 and y_2 are loose in $\boldsymbol{q}_2$ provided that the original TBox does not entail $\exists T \sqsubseteq \exists R^-$ and $\exists T^- \sqsubseteq \exists R$, respectively. Note, however, that both of these concept inclusions will hold in any interpretation $\mathcal{I}$ with $\mathcal{I} \not\models \boldsymbol{q}_2$ because the query 'encodes' the role inclusion $T^- \sqsubseteq R$. These examples show that the requirement for each negative atom to have a loose variable is not particularly restrictive and, in fact, not much stronger than simply non-entailment of the negation of the constituent CQ$^{\neg s}$ by the original TBox alone.

Lemma 7. *Let $\mathcal{T}$ be a DL-Lite$^{\mathcal{H}}_{\text{core}}$ TBox and $\boldsymbol{q}$ a Boolean UCQ$^{\neg s}$ such that each component $\boldsymbol{q}_i$ of $\boldsymbol{q}$ is tree-shaped and each negative atom in each $\boldsymbol{q}_i$ contains a $\mathcal{T}$-loose variable. Then there exist a DL-Lite$^{\mathcal{H}}_{\text{core}}$ TBox $\mathcal{T}'$ and a CQ$^{\neg s}$ $\boldsymbol{q}'$ such that*

$$(\mathcal{T}, \mathcal{A}) \models \boldsymbol{q} \quad \textit{iff} \quad (\mathcal{T} \cup \mathcal{T}', \mathcal{A}) \models \boldsymbol{q}', \quad \textit{for every ABox } \mathcal{A}.$$

Proof. Let $\boldsymbol{q}_i$ be of the form $\exists \boldsymbol{y}_i\, \varphi_i(\boldsymbol{y}_i)$, for $1 \leq i \leq n$. Since tree-shaped queries contain no individuals, each $\boldsymbol{y}_i$ is non-empty and we can fix a variable, say, y_{i1}, in each $\boldsymbol{y}_i$. Let y be a fresh variable and, for each $1 \leq i \leq n$, let G_i be a fresh role name. Define $\varphi'_i(y, \boldsymbol{y}_i) = G_i(y, y_{i1}) \wedge \hat{\varphi}_i(\boldsymbol{y}_i)$, where $\hat{\varphi}_i$ is the result of replacing each concept name A with a fresh $\hat{A}$ and each role name P with a fresh $\hat{P}$ in φ_i. Consider

$$\boldsymbol{q}' \;=\; \exists y\, \boldsymbol{y}_1 \ldots \boldsymbol{y}_n \bigwedge_{1 \leq i \leq n} \varphi'_i(y, \boldsymbol{y}_i).$$

Let D be a fresh concept name. Let $\mathcal{T}_D$ consist of $A \sqsubseteq \hat{A}$ and $A \sqsubseteq D$, for each concept name A occurring in $\mathcal{T}$ or $\boldsymbol{q}$, and $P \sqsubseteq \hat{P}$, $\exists P \sqsubseteq D$ and $\exists P^- \sqsubseteq D$, for each role name P in $\mathcal{T}$ or $\boldsymbol{q}$. Thus, in any model of $\mathcal{T}_D$, the interpretation of D contains the interpretations of all concepts of $\mathcal{T}$ and $\boldsymbol{q}$, including domains and ranges of its roles.

Since each $\varphi'_i(y, \boldsymbol{y}_i)$ is tree-shaped, we can assume that its primal graph is a rooted tree with root y (so that each edge has a natural orientation away from the root); by construction, the root has a single successor, y_{i1}. We write $z \prec z'$ if z is a (unique) immediate predecessor of z' in one of these trees. For each edge (z, z') with $z \prec z'$, we take a fresh role $E_{zz'}$. Let $\mathcal{T}_G$ contain the following inclusions, for all $1 \leq i \leq n$:

$$D \sqsubseteq \exists G_{i,0}^-, \tag{17}$$

$$\exists G_{i,0} \sqsubseteq \exists G_{j,1}, \qquad \text{for } 1 \leq j \leq n \text{ with } j \neq i, \tag{18}$$

$$G_{i,k} \sqsubseteq G_i, \qquad \text{for } k = 0, 1, \tag{19}$$

$$G_{i,1} \sqsubseteq E_{yy_{i1}}, \tag{20}$$

$$\exists E_{zz'}^- \sqsubseteq \exists E_{z'z''}, \qquad \text{for } z \prec z' \prec z'', \tag{21}$$

$$\exists E_{zz'}^- \sqsubseteq \hat{A}, \qquad \text{for all } \hat{A}(z') \text{ in } \hat{\varphi}_i, \tag{22}$$

$$E_{zz'} \sqsubseteq \hat{R}, \qquad \text{for all } \hat{R}(z, z') \text{ in } \hat{\varphi}_i, \tag{23}$$

$$\exists E_{zz'}^- \sqcap \hat{A} \sqsubseteq \bot, \qquad \text{for all } \neg\hat{A}(z') \text{ in } \hat{\varphi}_i, \tag{24}$$

$$\exists E_{zz'}^- \sqcap \exists\hat{R} \sqsubseteq \bot, \qquad \text{for all } \neg\hat{R}(z', z'') \text{ in } \hat{\varphi}_i \text{ with loose } z', \tag{25}$$

where $G_{i,0}$ and $G_{i,1}$ are fresh role names. Let $\mathcal{T}' = \mathcal{T}_D \cup \mathcal{T}_G$. Note that it is crucial that z' is loose in both (24) and (25)—for otherwise $\mathcal{T} \cup \mathcal{T}'$ would imply emptiness of any interpretation of D. We claim that $\mathcal{T}'$ and $\boldsymbol{q}'$ are as required.

Suppose first that $(\mathcal{T}, \mathcal{A}) \models \boldsymbol{q}$ and let $\mathcal{I}$ be a model of $(\mathcal{T} \cup \mathcal{T}', \mathcal{A})$. As $\mathcal{I} \models (\mathcal{T}, \mathcal{A})$, we have $\mathcal{I} \models \boldsymbol{q}$. So, for some i, $1 \leq i \leq n$, there exists a match π for $\boldsymbol{q}_i$ in $\mathcal{I}$. Since the negations in $\boldsymbol{q}$ are safe, $\pi(y_{i1})$ belongs to $A^{\mathcal{I}}$, for some concept name A in $\mathcal{T}$, or to $(\exists R)^{\mathcal{I}}$, for some role R in $\mathcal{T}$; whence, $\pi(y_{i1}) \in D^{\mathcal{I}}$. Let $\boldsymbol{q}_*$ consist of all atoms of $\boldsymbol{q}'$ that are not in $\hat{\varphi}_i(\boldsymbol{y}_i)$. Since $\mathcal{I} \models \mathcal{T}_G$, there exists a match π_* for $\boldsymbol{q}_*$ in $\mathcal{I}$ with $\pi_*(y_{i1}) = \pi(y_{i1})$. Indeed, by (20)–(23), the tree of the positive atoms of $\boldsymbol{q}_*$ can be matched in the tree rooted in the $G^-_{i,0}$-successor of $\pi(y_{i1})$; by (24) and (25), the negative atoms are also matched 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 $\mathcal{T}$ or $\boldsymbol{q}$, and, additionally, interprets D by $\Delta^{\mathcal{I}}$, and $\hat{A}$ and $\hat{P}$ by $A^{\mathcal{I}}$ and $P^{\mathcal{I}}$, for each concept name A and role name P in $\mathcal{T}$ or $\boldsymbol{q}$. By construction, $\mathcal{I}_0 \models (\mathcal{T} \cup \mathcal{T}_D, \mathcal{A})$ and $\mathcal{I}_0 \not\models \boldsymbol{q}$. Denote by $\mathcal{C}_d$ the canonical interpretation of $(\mathcal{T}_G, \{D(d)\})$, for $d \in \Delta^{\mathcal{I}_0}$ (we slightly abuse notation here and treat domain elements as fresh individual names assuming that $d^{\mathcal{C}_d} = d$). By definition, each $\mathcal{C}_d$ is finite and their domains are pairwise disjoint. Let $\mathcal{I}'$ be the union of $\mathcal{I}_0$ with all $\mathcal{C}_d$, $d \in \Delta^{\mathcal{I}_0}$. Since each negative atom of $\boldsymbol{q}$ contains a loose variable, $\mathcal{I}'$ does not violate any negative inclusions of $\mathcal{T}_G$, that is, (24) and (25). Thus, $\mathcal{I}' \models (\mathcal{T} \cup \mathcal{T}', \mathcal{A})$. Finally, for the sake of contradiction, suppose $\mathcal{I}' \models \boldsymbol{q}'$. Then there is a match π for $\boldsymbol{q}'$ in $\mathcal{I}'$. By the definition of $\boldsymbol{q}'$, $\pi(y)$ must be the element in one of the $\mathcal{C}_d$ introduced to witness the existential restriction in (17). By (18), atoms corresponding to one of the components, say $\boldsymbol{q}_i$, of $\boldsymbol{q}$ must be matched in the part of the original model $\mathcal{I}_0$, contrary to $\mathcal{I}_0 \not\models \boldsymbol{q}_i$, for all i, $1 \leq i \leq n$. ❑

Consider now the UCQ$^{\neg s}$ and the TBox obtained in the proof of Corollary 5 from the query and the TBox in the proofs of Theorem 3 and Lemma 4. It can be verified that the components of the UCQ$^{\neg s}$ are tree-shaped and satisfy the conditions of Lemma 7. Thus, we obtain undecidability of CQ$^{\neg s}$ answering over *DL-Lite*$^{\mathcal{H}}_{\text{core}}$ KBs.

Theorem 8. *There exist a Boolean CQ*$^{\neg s}$ $\boldsymbol{q}$ *and a DL-Lite*$^{\mathcal{H}}_{\text{core}}$ *TBox* $\mathcal{T}$ *such that* CertainAnswers$(\boldsymbol{q}, \mathcal{T})$ *is undecidable.*

This solves the open problem of decidability of CQ$^{\neg s}$ answering over *DL-Lite*$^{\mathcal{H}}_{\text{core}}$ (Rosati, 2007). However, since role inclusions are required in the transformation in Lemma 7, the decidability of the CQ$^{\neg s}$ answering problem over *DL-Lite*$_{\text{core}}$ remains open. On the other hand, by Corollary 5, answering *unions* of CQs$^{\neg s}$ over *DL-Lite*$_{\text{core}}$ is undecidable. The number of queries in the union constructed in the proof of Corollary 5 depends, however, on the size of the alphabet and the number of states of the universal Turing machine (more precisely, it is $(2 \cdot |Q| + 1) \cdot |\Gamma| + 4)$. We can strengthen the negative result to a union of only three queries.

Theorem 9. *There exist a union* $\boldsymbol{q}$ *of three Boolean CQs*$^{\neg_1 s}$ *and a DL-Lite*$_{\text{core}}$ *TBox* $\mathcal{T}$ *such that* CertainAnswers$(\boldsymbol{q}, \mathcal{T})$ *is undecidable.*

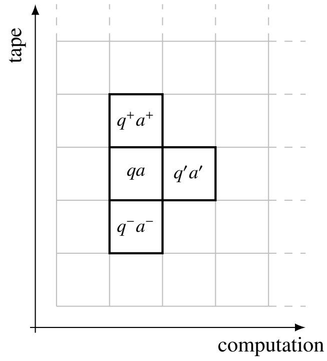


Figure 4: Quadruples $\tau = (q^-a^-, qa, q^+a^+, q'a')$.

Proof. The proof again is by reduction of the halting problem for deterministic Turing machines. Let $M = (\Gamma, Q, q_0, q_1, \delta)$ be a deterministic Turing machine; see the proof of Theorem 3.

Similarly to the construction in the proof of Theorem 3, we represent computations of M in a two-dimensional grid, where role T points to the representation of the next cell on the tape and role S to the representation of the same cell in the successor configuration. However, we now use a role E to relate the representation of a cell containing $a \in \Gamma$ in a configuration with state $q \in Q$ and the head positioned over the cell to an individual e_{qa}; if the head is not over the cell then its representation is E-related to $e_{\emptyset a}$, where $\emptyset$ is a no-head marker; the representation of the cells in the initial configuration beyond the input word is E-related to a special individual $e_{*\sqcup}$, where $*$ is a tape initialisation marker. We abbreviate pairs $(q, a) \in (Q \cup \{\emptyset, *\}) \times \Gamma$ simply as qa and say that a cell contains such qa if it contains a and either it is under the head in the state $q \in Q$ or it is not under the head and $q \in \{\emptyset, *\}$.

Consider a set $\mathfrak{T}_M$ of quadruples of the form

$$(q^-a^-,\ qa,\ q^+a^+,\ q'a')$$

that are defined by the transition function δ: if cells $i-1$, i and $i+1$ contain pairs q^-a^-, qa and q^+a^+, respectively, then the cell i contains pair $q'a'$ in the successive configuration; see Fig. 4. Note that, since M is deterministic, the pair $q'a'$ is determined uniquely. We also include special quadruples in $\mathfrak{T}_M$ for initialisation of the tape beyond the input word: for $a, a' \in \Gamma$,

$$(\emptyset a', \emptyset a, *_\sqcup, \emptyset a), \quad (\emptyset a, *_\sqcup, *_\sqcup, \emptyset_\sqcup), \quad (*_\sqcup, *_\sqcup, *_\sqcup, \emptyset_\sqcup).$$

We assume that the input word contains at least three symbols, and so none of the first three cells of the tape contain $*_\sqcup$.

In addition to individual names e_{qa} for the pairs qa, take an individual name e_τ for each quadruple $\tau \in \mathfrak{T}_M$. Let P_-, P, P_+ and P' be role names and let ABox $\mathcal{A}_M$ contain assertions

$$P_-(e_{q^-a^-}, e_\tau), \quad P(e_{qa}, e_\tau), \quad P_+(e_{q^+a^+}, e_\tau), \quad P'(e_\tau, e_{q'a'}),$$

for each quadruple $\tau = (q^-a^-, qa, q^+a^+, q'a')$ in $\mathfrak{T}_M$. Also, the ABox $\mathcal{A}_M$ uses a fresh concept name N to mark all the pairs with the accepting state $q_1 \in Q$ and contains

$$N(e_{q_1 a}), \qquad \text{for all } a \in \Gamma.$$

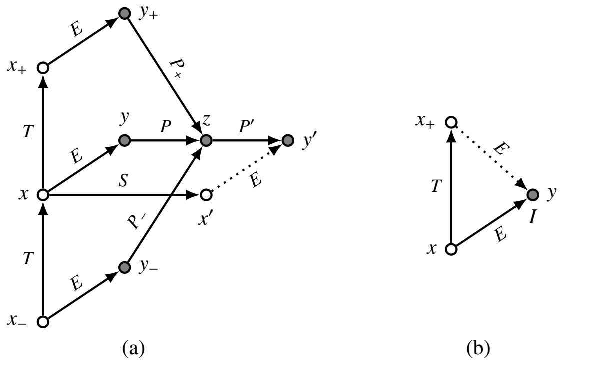


Figure 5: The last two components of the query in the proof of Theorem 9.

Another ABox, $\mathcal{A}_w$, encodes an input $w = a_1, \dots, a_n \in \Gamma^*$ on the tape as follows:

$$T(c_0, c_0),\ \ E(c_0, e_{\emptyset\textvisiblespace}),\qquad T(c_0, c_1),\ \ E(c_1, e_{q_0 a_1}),$$
$$T(c_{i-1}, c_i),\ \ E(c_i, e_{\emptyset a_i}),\ \text{for } 1 < i \le n,$$
$$T(c_n, c_{n+1}),\ \ E(c_{n+1}, e_{*\textvisiblespace}),\ \ I(e_{*\textvisiblespace}),$$

where $c_1, \dots, c_n$ are fresh individual names, corresponding to the cells of the input, c_0 and c_{n+1} are special individuals placed 'before' and 'after' the input word in the initial configuration of the tape, and I is a fresh concept name for initialisation of the tape beyond the input (note that there is a T-loop in c_0).

Consider now a union q of the following three CQs$^{\neg 1 s}$ given in negated form (see Fig. 1 for the first and Fig. 5 for the last two):

$$S(x, y) \land T(x, z) \land S(z, u) \ \to\ T(y, u), \tag{26}$$

$$\begin{aligned} &E(x, y) \land P(y, z) \land S(x, x') \land P'(z, y') \land \\ &\quad T(x_-, x) \land E(x_-, y_-) \land P_-(y_-, z) \land \\ &\quad T(x, x_+) \land E(x_+, y_+) \land P_+(y_+, z) \ \to\ E(x', y'), \end{aligned} \tag{27}$$

$$T(x, x_+) \land E(x, y) \land I(y) \ \to\ E(x_+, y). \tag{28}$$

Let TBox $\mathcal{T}$ contain

$$\exists T \sqsubseteq \exists S, \qquad \exists T^- \sqsubseteq \exists T, \qquad \exists E^- \sqcap N \sqsubseteq \bot.$$

We claim that $(\mathcal{T}, \mathcal{A}_M \cup \mathcal{A}_w) \not\models q$ if and only if M does not accept w.

Consider a model $\mathcal{I}$ of $(\mathcal{T}, \mathcal{A}_M \cup \mathcal{A}_w)$ with $\mathcal{I} \not\models q$. Then there exists an infinite sequence of (not necessarily distinct) domain elements $d_0, d_1, d_2, \dots$ that encode the initial configuration in the sense that $(d_0, d_0) \in T^{\mathcal{I}}$, $(d_i, d_{i+1}) \in T^{\mathcal{I}}$ for all $i \ge 0$, and each element is connected by the interpretation of E to the element of the corresponding pair, that is, $E^{\mathcal{I}}$ contains $(d_0, e^{\mathcal{I}}_{\emptyset\textvisiblespace})$, $(d_1, e^{\mathcal{I}}_{q_0 a_1})$, all $(d_i, e^{\mathcal{I}}_{\emptyset a_i})$, for $1 < i \le n$, and all $(d_i, e^{\mathcal{I}}_{*\textvisiblespace})$, for $i > n$. Note that $d_0 = c_0^{\mathcal{I}}$ is an auxiliary element before the tape, whose role is to match the (positive part of the) second component of q for the representation of the first cell, and $e_{*\textvisiblespace}$ serves as a substitute for $e_{\emptyset\textvisiblespace}$, which is necessary, along with concept I and the third component of q, to initialise the tape beyond the input. By the first TBox inclusion, there exists a sequence of elements $d'_0, d'_1, d'_2, \dots$ such that $(d_i, d'_i) \in S^{\mathcal{I}}$. By the first component of q, they form another T-connected sequence, that is, $(d'_i, d'_{i+1}) \in T^{\mathcal{I}}$ for all i. Moreover, since d_0 has a $T^{\mathcal{I}}$-loop, d'_0 also has a $T^{\mathcal{I}}$-loop. By $\mathcal{A}_M$ and the second component of q, the sequence represents the second configuration of the computation in the same way, except that now $e_{*\textvisiblespace}$ is not used: instead, by the tape initialisation quadruples, all the cells beyond the working space are $E^{\mathcal{I}}$-connected to $e_{\emptyset\textvisiblespace}$. Note that d'_0 is also $E^{\mathcal{I}}$-connected to $e_{\emptyset\textvisiblespace}$. By the same argument, there exists a sequence of elements for each configuration of the computation. Finally, the negative concept inclusion in $\mathcal{T}$ and assertions in $\mathcal{A}_M$ guarantee that the accepting state never occurs in the computation, and so, M does not accept w.

Conversely, if M has a non-accepting computation on w then it is routine to construct an infinite two-dimensional grid-like interpretation $\mathcal{I}$ satisfying $(\mathcal{T}, \mathcal{A}_M \cup \mathcal{A}_w)$ but not q (all domain elements in the bottom row of the grid have a $T^{\mathcal{I}}$-loop). ❑

We note in passing that the query q in the proof of Theorem 9 is not tree-shaped, and therefore Lemma 7 is not applicable.

3.3. Guarded Negation: Decidability

In this section we narrow down the class of CQs with safe negation and concentrate on guarded negation. As follows from the results by Bárány et al. (2012), answering unions of GNCQs over ontologies in the language of the so-called *frontier-guarded tuple-generating dependencies* (*fg-tgds*) is decidable and in coNP in data complexity; moreover, it is in P in data complexity if each GNCQ in the union contains at most one negated atom. Observe that (*i*) $\mathcal{ELI}$ concept and role inclusions are a particular form of frontier-guarded tgds, and that (*ii*) negative concept and role inclusions can be viewed as negated CQs. Therefore, the upper complexity bounds also apply to $\mathcal{ELI}_\bot$ and *DL-Lite*$^{\mathcal{H}}_{\text{core}}$ KBs. We establish the matching lower complexity bounds even for a TBox $\mathcal{T}_0$ containing a single negative concept inclusion

$$V \sqcap F \sqsubseteq \bot$$

(by definition, $\mathcal{T}_0$ is in both $\mathcal{EL}_\bot$ and *DL-Lite*$_{\text{core}}$).

Lemma 10. *There exists a Boolean GNCQ q with one negated atom such that the problem* CertainAnswers$(q, \mathcal{T}_0)$ *is* P*-hard.*

Proof. The proof is by reduction of the complement of Horn-3SAT, the satisfiability problem for Horn clauses with at most three literals, which is known to be P-complete; see, e.g., (Papadimitriou, 1994). Suppose we are given a conjunction ψ of Horn clauses of the form p, $\neg p$ and $p_1 \land p_2 \to p$, where p, p_1 and p_2 are propositional variables. Consider a Boolean GNCQ q with the following negated form:

$$N_1(x_1, y) \land V(x_1) \land N_2(x_2, y) \land V(x_2) \land R(y, z) \ \to\ V(z);$$

see Fig. 6 (a). Note that q does not depend on ψ.

Next, we construct an ABox $\mathcal{A}_\psi$ such that ψ is satisfiable iff $(\mathcal{T}_0, \mathcal{A}_\psi) \not\models q$. The ABox $\mathcal{A}_\psi$ uses an individual name c_p for each variable p in ψ and an individual name c_γ for each clause

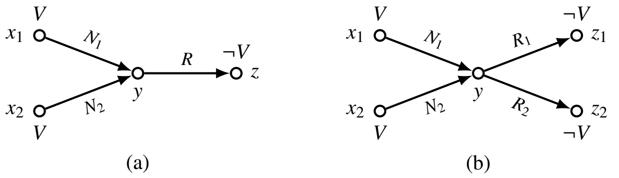


Figure 6: GNCQs in the proofs of Lemmas 10 and 11.

γ of the form $p_1 \land p_2 \to p$ in ψ. For every clause γ, the ABox $\mathcal{A}_\psi$ contains the following assertions:

$$\begin{aligned} V(c_p), &\quad \text{if } \gamma = p,\\ F(c_p), &\quad \text{if } \gamma = \neg p,\\ N_1(c_{p_1}, c_\gamma),\ N_2(c_{p_2}, c_\gamma),\ R(c_\gamma, c_p), &\quad \text{if } \gamma = p_1 \land p_2 \to p. \end{aligned}$$

Suppose first there is a model $\mathcal{I}$ of $(\mathcal{T}_0, \mathcal{A}_\psi)$ with $\mathcal{I} \not\models \boldsymbol{q}$. We show that ψ is satisfiable. Observe that, for each clause γ of ψ of the form $p_1 \land p_2 \to p$, if both $c_{p_1}^{\mathcal{I}} \in V^{\mathcal{I}}$ and $c_{p_2}^{\mathcal{I}} \in V^{\mathcal{I}}$ then $c_p \in V^{\mathcal{I}}$. Thus, we can define a satisfying assignment $\mathfrak{a}$ for ψ by taking $\mathfrak{a}(p)$ true iff $c_p^{\mathcal{I}} \in V^{\mathcal{I}}$.

Conversely, if ψ is satisfiable then we can evidently construct a model $\mathcal{I}$ of $(\mathcal{T}_0, \mathcal{A}_\psi)$ with $\mathcal{I} \not\models \boldsymbol{q}$. ❑

Lemma 11. *There exists a Boolean GNCQ* $\boldsymbol{q}$ *with two negated atoms such that* CertainAnswers$(\boldsymbol{q}, \mathcal{T}_0)$ *is* coNP*-hard.*

Proof. The proof is by reduction of the complement of 2+2SAT, the satisfiability problem for clauses with two negative and two positive literals, which is known to be NP-complete (Schaerf, 1993). Suppose we are given a conjunction ψ of clauses of the form $\neg p_1 \lor \neg p_2 \lor p_1' \lor p_2'$, where each p_i and p_i' is either a propositional variable or one of the two propositional constants, *true* and *false*. Consider a Boolean GNCQ $\boldsymbol{q}$ with the following negated form:

$$\begin{aligned} N_1(x_1, y) \land V(x_1) \land N_2(x_2, y) \land V(x_2) \land&\\ R_1(y, z_1) \land R_2(y, z_2) \quad &\to \quad V(z_1) \lor V(z_2); \end{aligned}$$

see Fig. 6 (b). Observe that the query is similar to the one in the proof of Lemma 10 except that now we have two R_i-atoms instead of one R-atom. Note again that $\boldsymbol{q}$ does not depend on ψ.

Next, we construct an ABox $\mathcal{A}_\psi$ such that ψ is satisfiable iff $(\mathcal{T}_0, \mathcal{A}_\psi) \not\models \boldsymbol{q}$. The ABox $\mathcal{A}_\psi$ uses individual names c_{true} and c_{false} for the two constants, an individual name c_p for each variable p in ψ and an individual name c_γ for each clause γ in ψ. It contains $V(c_{true})$, $F(c_{false})$ and the following assertions, for every clause γ of the form $\neg p_1 \lor \neg p_2 \lor p_1' \lor p_2'$ in ψ:

$$N_1(c_{p_1}, c_\gamma), \quad N_2(c_{p_2}, c_\gamma), \quad R_1(c_\gamma, c_{p_1'}), \quad R_2(c_\gamma, c_{p_2'}).$$

Suppose first there is a model $\mathcal{I}$ of $(\mathcal{T}_0, \mathcal{A}_\psi)$ with $\mathcal{I} \not\models \boldsymbol{q}$. We show that ψ is satisfiable. Observe that, for each clause of ψ of the form $\neg p_1 \lor \neg p_2 \lor p_1' \lor p_2'$, if both $c_{p_1}^{\mathcal{I}} \in V^{\mathcal{I}}$ and $c_{p_2}^{\mathcal{I}} \in V^{\mathcal{I}}$ then *either* $c_{p_1'}^{\mathcal{I}} \in V^{\mathcal{I}}$ or $c_{p_2'}^{\mathcal{I}} \in V^{\mathcal{I}}$. Since we have $c_{true}^{\mathcal{I}} \in V^{\mathcal{I}}$ and $c_{false}^{\mathcal{I}} \notin V^{\mathcal{I}}$, a satisfying assignment $\mathfrak{a}$ for ψ can be defined by taking $\mathfrak{a}(p)$ true iff $c_p^{\mathcal{I}} \in V^{\mathcal{I}}$.

Conversely, if ψ is satisfiable then we can evidently construct a model $\mathcal{I}$ of $(\mathcal{T}_0, \mathcal{A}_\psi)$ with $\mathcal{I} \not\models \boldsymbol{q}$. ❑

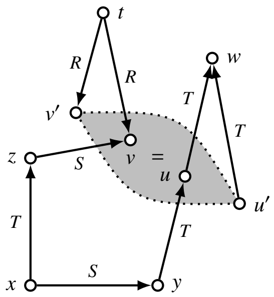


Figure 7: CQ$^{\neq}$ in the proof of Theorem 13.

Summing up, we obtain the following result.

Theorem 12. *The problems of answering GNCQs and unions of GNCQs over DL-Lite*$_{\text{core}}$, *DL-Lite*$_{\text{core}}^{\mathcal{H}}$, $\mathcal{EL}_\bot$ *and* $\mathcal{ELI}_\bot$ *KBs are* coNP*-complete in data complexity. The problems are* P*-complete if the GNCQ and each component in the union, respectively, have at most one negation.*

4. Answering CQs with Inequalities

In this section we first prove that CQ$^{\neq}$ answering over *DL-Lite*$_{\text{core}}^{\mathcal{H}}$ is undecidable, even if only one inequality may be used. Over *DL-Lite*$_{\text{core}}$, we show undecidability for unions of three CQs$^{\neq}$, as well as P- and coNP-hardness for CQs$^{\neq}$. We then observe that one of the reasons for undecidability is applying inequalities to the non-ABox elements in interpretations and identify a class of CQs$^{\neq}$, *local CQs*$^{\neq}$, that require at least one of the arguments in any inequality to be an ABox element. We show that this restriction guarantees decidability of the query answering problem.

4.1. CQs with Inequalities over DL-Lite$_{\text{core}}^{\mathcal{H}}$: Undecidability

We begin by establishing undecidability of CQ$^{\neq}$ answering over *DL-Lite*$_{\text{core}}^{\mathcal{H}}$. In principle, the technique of Lemma 7 could be adapted to queries with inequalities and by using, e.g., a modification of the proof of Theorem 1 in (Gutiérrez-Basulto et al., 2012), 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 $\mathcal{EL}_\bot$ (Klenke, 2010) and show that even one inequality suffices for *DL-Lite*$_{\text{core}}^{\mathcal{H}}$.

Theorem 13. *There exist a Boolean CQ*$^{\neq}$ $\boldsymbol{q}$ *with one inequality and a DL-Lite*$_{\text{core}}^{\mathcal{H}}$ *TBox* $\mathcal{T}$ *such that* CertainAnswers$(\boldsymbol{q}, \mathcal{T})$ *is undecidable.*

Proof. Similarly to the proof of Theorem 3, we reduce the halting problem for deterministic Turing machines to CertainAnswers$(\boldsymbol{q}, \mathcal{T})$. We also use a two-dimensional grid formed by roles T and S. This time, however, the grid is established (along with functionality of certain roles) by means of a Boolean CQ$^{\neq}$ $\boldsymbol{q}$ with the following negated form:

$$\begin{aligned} S(x, y) \land T(x, z) \land S(z, v) \land T(y, u) \land&\\ T(u, w) \land T(u', w) \land R(t, v) \land R(t, v')&\\ &\to (u' = v'). \end{aligned}$$

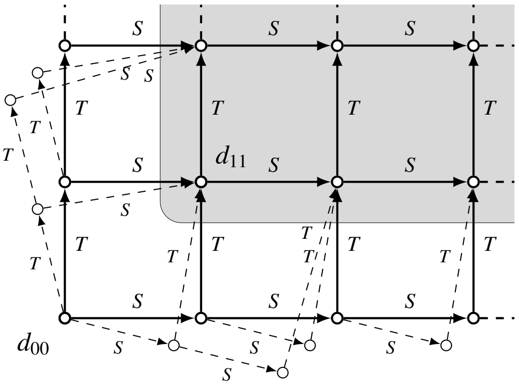


Figure 8: The grid structure in the proof of Theorem 13.

Note that this sentence, in fact, implies $v = v' = u' = u$; see the shaded area in Fig. 7.

We present the construction of the TBox $\mathcal{T}$ in a series of steps. 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 $\mathcal{I}$, in addition, is its model then $\mathcal{I}$ enjoys certain structural properties. We say that the interpretation $P^{\mathcal{I}}$ of a role P is *functional in* $d \in \Delta^{\mathcal{I}}$ if $d' = d''$ whenever both (d, d') and (d, d'') are in $P^{\mathcal{I}}$. We also denote the composition of binary relations by $\circ$, for example:

$$S^{\mathcal{I}} \circ T^{\mathcal{I}} = \{ (d, d'') \mid (d, d') \in S^{\mathcal{I}}, (d', d'') \in T^{\mathcal{I}} \}.$$

Let the first part, $\mathcal{T}_G$, of the TBox contain the following concept inclusions:

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

We claim that if $\mathcal{I} \models \mathcal{T}_G$ and $\mathcal{I} \models \exists T \sqsubseteq \exists S$ then the fragment of $\mathcal{I}$ rooted in element $d_{11} \in (\exists S^-.\exists T^-)^{\mathcal{I}}$ has a grid structure of the shaded area in Fig. 8 (each domain element in $(\exists S^-)^{\mathcal{I}}$ also has an $R^{\mathcal{I}}$-predecessor, which is not shown). Note that $\mathcal{T}_G$ ensures that domain elements in $(\exists S^-)^{\mathcal{I}}$ only have $T^{\mathcal{I}}$- and $(R^-)^{\mathcal{I}}$-successors but not necessarily $S^{\mathcal{I}}$-successors (existence of $S^{\mathcal{I}}$-successors will be guaranteed by concept and role inclusions (31)–(33), (41), (42) and $\mathcal{T}_F$ to be defined below).

More formally, the domain elements in the shaded area enjoy the following property.

Claim 13.1. *If $\mathcal{I} \models \mathcal{T}_G$ and $\mathcal{I} \not\models \boldsymbol{q}$ then, for every $d \in \Delta^{\mathcal{I}}$ with an $S^{\mathcal{I}}$-successor and a $T^{\mathcal{I}} \circ S^{\mathcal{I}}$-successor,*

(a) *$S^{\mathcal{I}}$ is functional in any $T^{\mathcal{I}}$-successor of d,*

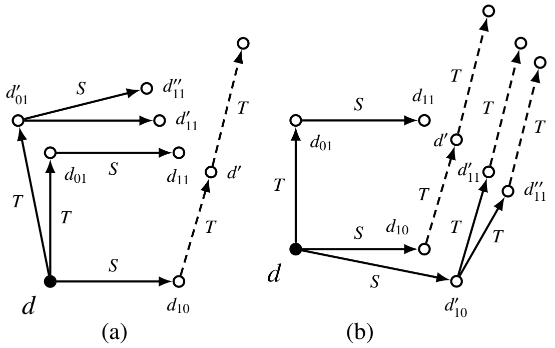
(b) *$T^{\mathcal{I}}$ is functional in any $S^{\mathcal{I}}$-successor of d,*

(c) *all $T^{\mathcal{I}} \circ S^{\mathcal{I}}$- and $S^{\mathcal{I}} \circ T^{\mathcal{I}}$-successors of d coincide,*

(d) *$(T^-)^{\mathcal{I}}$ is functional in any $T^{\mathcal{I}} \circ S^{\mathcal{I}} \circ T^{\mathcal{I}}$-successor of d,*

(e) *$R^{\mathcal{I}}$ is functional in any $T^{\mathcal{I}} \circ S^{\mathcal{I}} \circ (R^-)^{\mathcal{I}}$-successor of d.*

Proof of claim. There are domain elements d_{10}, d_{01}, d_{11} such that $(d, d_{10}) \in S^{\mathcal{I}}$, $(d, d_{01}) \in T^{\mathcal{I}}$ and $(d_{01}, d_{11}) \in S^{\mathcal{I}}$.



(a) (b)

Figure 9: Proof of Claim 13.1.

(a) Let $(d, d'_{01}) \in T^{\mathcal{I}}$ and (d'_{01}, d'_{11}), $(d'_{01}, d''_{11}) \in S^{\mathcal{I}}$; see Fig. 9 (a). Since $\mathcal{I} \models \mathcal{T}_G$, the element d_{10} has a $T^{\mathcal{I}}$-successor d', which in turn has a $T^{\mathcal{I}}$-successor too; each of d_{11}, d'_{11} and d''_{11} has an $R^{\mathcal{I}}$-predecessor (not shown in Fig. 9 (a)). As $\mathcal{I} \not\models \boldsymbol{q}$, each of d_{11}, d'_{11} and d''_{11} coincides with d' and thus, $S^{\mathcal{I}}$ is functional in any $T^{\mathcal{I}}$-successor of d.

(b) Let $(d, d'_{10}) \in S^{\mathcal{I}}$ and (d'_{10}, d'_{11}), $(d'_{10}, d''_{11}) \in T^{\mathcal{I}}$; see Fig. 9 (b). Since $\mathcal{I} \models \mathcal{T}_G$, the element d_{10} has a $T^{\mathcal{I}}$-successor d', which in turn has a $T^{\mathcal{I}}$-successor too; also, d_{11} has an $R^{\mathcal{I}}$-predecessor (not shown in Fig. 9 (b)); and both d'_{11} and d''_{11} have $T^{\mathcal{I}}$-successors. As $\mathcal{I} \not\models \boldsymbol{q}$, each of d', d'_{11} and d''_{11} coincides with d_{11}. So, $T^{\mathcal{I}}$ is functional in any $S^{\mathcal{I}}$-successor of d.

(c) Is not difficult to see now that all $T^{\mathcal{I}} \circ S^{\mathcal{I}}$-successors and all $S^{\mathcal{I}} \circ T^{\mathcal{I}}$-successors coincide. Denote this element by d'.

(d) and (e) By item (c), $(T^-)^{\mathcal{I}}$ is functional in any $T^{\mathcal{I}}$-successor of d' and $R^{\mathcal{I}}$ is functional in any $R^{\mathcal{I}}$-predecessor of d'. ∎

So, $S^{\mathcal{I}}$ and $T^{\mathcal{I}}$ are functional in all domain elements in the shaded area. However, $S^{\mathcal{I}}$ does not have to be functional in the bottom row and $T^{\mathcal{I}}$ in the left column (see Fig. 8); $(T^-)^{\mathcal{I}}$ is functional in all domain elements in the shaded area except its bottom row but it does not have to be functional elsewhere; $R^{\mathcal{I}}$ does not have to be functional anywhere but in $R^{\mathcal{I}}$-predecessors of the domain elements in the shaded area; finally, $(S^-)^{\mathcal{I}}$ 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 domain elements outside the grid. Besides this, we require role R to be functional not only in $R^{\mathcal{I}}$-predecessors of the grid elements but also in the grid elements themselves. To this end, we use a technique similar to the proof of Lemma 7.

Claim 13.2. *Let $\mathcal{I} \models \mathcal{T}_G$ and $\mathcal{I} \not\models \boldsymbol{q}$.*

(a) *If $\mathcal{I}$ satisfies*

$$E \sqsubseteq \exists T^-.\exists S \tag{29}$$

then $S^{\mathcal{I}}$ is functional in every $d \in E^{\mathcal{I}}$.

(b) *If $\mathcal{I}$ satisfies*

$$D \sqsubseteq \exists R.\exists S^-.\exists T^-.\exists S \tag{30}$$

then $R^{\mathcal{I}}$ is functional in every $d \in D^{\mathcal{I}}$.

Proof of claim. (a) Let $d \in E^{\mathcal{I}}$ have an $S^{\mathcal{I}}$-successor. Then d has a $T^{\mathcal{I}}$-predecessor d_1, which, in turn, has an $S^{\mathcal{I}}$-successor and a $T^{\mathcal{I}} \circ S^{\mathcal{I}}$-successor (the $S^{\mathcal{I}}$-successor of d). Thus, by Claim 13.1 (a) applied to d_1, we obtain functionality of $S^{\mathcal{I}}$ in d.

(b) The argument is essentially the same as in (a) but we apply Claim 13.1 (e) instead. ❚

We now describe the part of the TBox that encodes computations of a given Turing machine. Let $M = (\Gamma, Q, q_0, q_1, \delta)$ be a deterministic Turing machine (see the proof of Theorem 3) with a *two-symbol* tape alphabet $\Gamma = \{1, ␣\}$.

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, $S_␣$ and S_1, for the two symbols of the alphabet Γ to encode cell contents: the range of S_a represents cells containing $a \in \Gamma$.

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 $H_q \sqcap \exists S_a^- \sqsubseteq \exists S_{a'} \sqcap \exists S_{q'\sigma}$, where $S_{q'\sigma}$ is also a sub-role of S (recall that the latter is functional in the grid). Alas, *DL-Lite*$^{\mathcal{H}}_{core}$ does not allow conjunction on the left-hand side of concept inclusions. The following construction simulates the required inclusions by using functionality of just two roles, R and S. Let $\mathcal{T}_F$ contain (29), (30) and the following concept and role inclusions with fresh role names R_q, L_a and P_{qa}, for each $q \in Q \cup \{\emptyset\}$ and $a \in \Gamma$:

$$\begin{array}{lll}
\exists S_a^- \sqsubseteq D, & H_q \sqsubseteq \exists R_q, & \exists S_a^- \sqsubseteq \exists L_a, \\
 & R_q \sqsubseteq R, & L_a \sqsubseteq R, \\
 & & \\
\exists R_q^- \sqsubseteq E, & \exists R_q^- \sqsubseteq \exists P_{q␣}, & \exists R_q^- \sqsubseteq \exists P_{q1}, \\
 & P_{q␣} \sqsubseteq R, & P_{q1} \sqsubseteq S, \\
 & L_␣^- \sqsubseteq R, & L_1^- \sqsubseteq S.
\end{array}$$

Claim 13.3. *If $\mathcal{I} \models \mathcal{T}_G \cup \mathcal{T}_F$ and $\mathcal{I} \not\models \boldsymbol{q}$ then, for each $a \in \Gamma$ and $q \in Q \cup \{\emptyset\}$, we have*

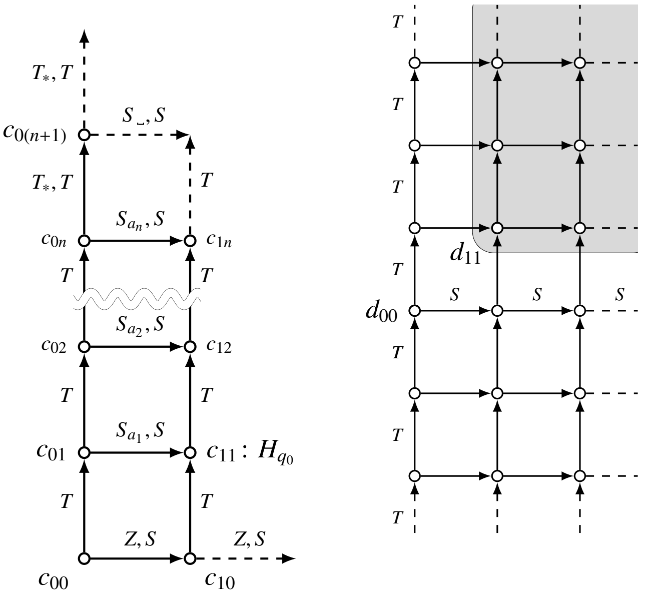
$$d \in (\exists P_{qa}^-)^{\mathcal{I}} \quad \textit{whenever} \quad d \in H_q^{\mathcal{I}} \cap (\exists S_a^-)^{\mathcal{I}},$$

for any d such that $R^{\mathcal{I}}$ is functional in any $R^{\mathcal{I}}$-predecessor of d.

Proof of claim. Let $d \in H_q^{\mathcal{I}} \cap (\exists S_a^-)^{\mathcal{I}}$. Then d has an $R_q^{\mathcal{I}}$-successor and an $L_a^{\mathcal{I}}$-successor, which coincide because, by Claim 13.2 (b), $R^{\mathcal{I}}$ is functional in $d \in D^{\mathcal{I}}$. Let d' be the $R^{\mathcal{I}}$-successor of d.

If $a = 1$ then the *inverse* of L_1 is 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 $(d', d'') \in S^{\mathcal{I}}$. Since $d' \in E^{\mathcal{I}}$, by Claim 13.2 (a), $S^{\mathcal{I}}$ is functional in d', whence $d = d''$. Therefore, $d \in (\exists P_{q1}^-)^{\mathcal{I}}$.

If $a = ␣$ then the argument is similar with R replacing S as the super-role of both $L_␣^-$ and $P_{q␣}$. As $R^{\mathcal{I}}$ is functional in any $R^{\mathcal{I}}$-predecessor of d, in particular in d', we obtain $d \in (\exists P_{q␣}^-)^{\mathcal{I}}$. ❚



(a)

(b)

Figure 10: (a) ABox $\mathcal{A}_w$ and (b) the three-way infinite grid in the proof of Theorem 13.

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

$$\exists P_{qa}^- \sqsubseteq \exists S_{a'} \sqcap \exists S_{q'\sigma}, \qquad \text{for } \delta(q, a) = (q', a', \sigma), \tag{31}$$

$$S_a \sqsubseteq S, \qquad \text{for } a \in \Gamma, \tag{32}$$

$$S_{q\sigma} \sqsubseteq S, \qquad \text{for } q \in Q \text{ and } \sigma \in \{-1, +1\}, \tag{33}$$

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

$$\exists P_{\emptyset a}^- \sqsubseteq \exists S_a, \qquad \text{for } a \in \Gamma. \tag{34}$$

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

$$\exists S_{q\sigma}^- \sqsubseteq \exists T_{q\sigma}, \qquad \text{for } q \in Q \text{ and } \sigma \in \{-1, +1\}, \tag{35}$$

$$\exists T_{q\sigma}^- \sqsubseteq H_q, \qquad \text{for } q \in Q \text{ and } \sigma \in \{-1, +1\}, \tag{36}$$

$$T_{q,+1} \sqsubseteq T \quad \text{and} \quad T_{q,-1} \sqsubseteq T^-, \qquad \text{for } q \in Q, \tag{37}$$

where $T_{q,+1}$ and $T_{q,-1}$ are used to propagate the head in the state q along the tape (recall that, by Claim 13.1, both $T^{\mathcal{I}}$ and $(T^-)^{\mathcal{I}}$ are functional in the grid); finally, the following concept inclusions are required to propagate the no-head marker $H_\emptyset$:

$$H_q \sqsubseteq \exists T_{\emptyset,+1} \quad \text{and} \quad H_q \sqsubseteq \exists T_{\emptyset,-1}, \qquad \text{for } q \in Q, \tag{38}$$

$$T_{\emptyset,+1} \sqsubseteq T \quad \text{and} \quad T_{\emptyset,-1} \sqsubseteq T^-, \tag{39}$$

$$\exists T_{\emptyset\sigma}^- \sqsubseteq \exists T_{\emptyset\sigma} \sqcap H_\emptyset, \qquad \text{for } \sigma \in \{-1, +1\}. \tag{40}$$

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

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

where Z is a fresh role name to start off an infinite sequence of configurations and T_* a fresh role name to fill the rest of the tape in the initial configuration by blanks:

$$\exists Z^- \sqsubseteq \exists Z, \qquad Z \sqsubseteq S, \tag{41}$$
$$\exists T_*^- \sqsubseteq \exists S_{\text{␣}} \sqcap \exists T_*, \qquad T_* \sqsubseteq T; \tag{42}$$

see Fig. 10 (a). Finally, the following concept inclusion ensures that the accepting state $q_1 \in Q$ never occurs in a computation:

$$H_{q_1} \sqsubseteq \bot. \tag{43}$$

Let $\mathcal{T}_M$ contain (31)–(43) encoding transitions of M and let $\mathcal{T} = \mathcal{T}_G \cup \mathcal{T}_F \cup \mathcal{T}_M$. 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, by Claims 13.1 and 13.3, we can extract from $\mathcal{I}$ a computation of M that does not accept $\boldsymbol{w}$ (for a similar argument, see the proofs of Theorems 3 and 9).

Conversely, if M does not accept $\boldsymbol{w}$ then we can construct a model $\mathcal{I}$ of $(\mathcal{T}, \mathcal{A}_{\boldsymbol{w}})$ with $\mathcal{I} \not\models \boldsymbol{q}$ as follows. First, it is routine to construct a model $\mathcal{J}_0$ of $\mathcal{T}_G$ such that

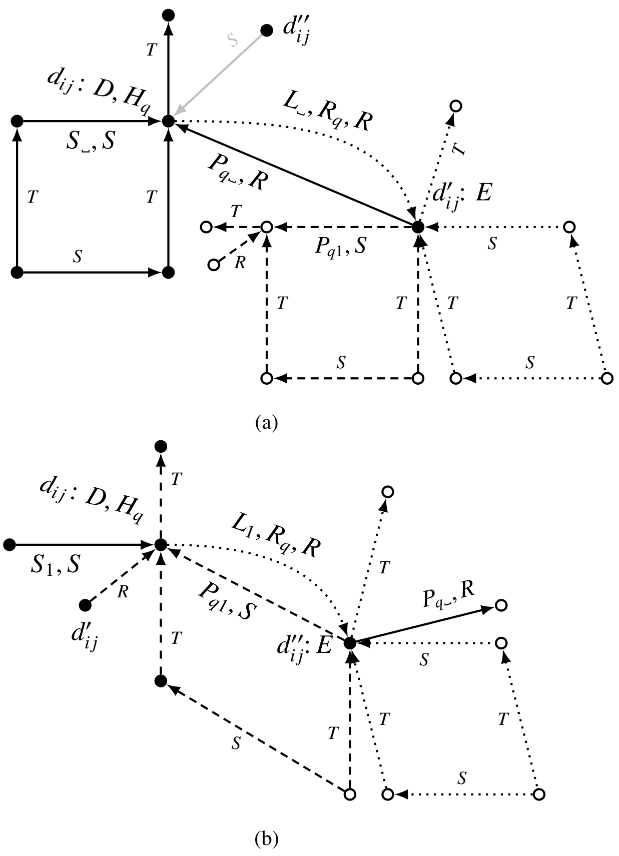
$$\Delta^{\mathcal{J}_0} = \{\, d_{ij} \mid i \geq 0 \text{ and } j \in \mathbb{Z} \,\} \cup \{\, d'_{ij},\ d''_{ij} \mid i > 0 \text{ and } j \in \mathbb{Z} \,\},$$

the d_{ij} form a three-way infinite grid structure on roles S and T (see Fig. 10 (b)), each d'_{ij} is an $R^{\mathcal{J}_0}$-predecessor of d_{ij} and each d''_{ij} is an $S^{\mathcal{J}_0}$-predecessor of d_{ij} (note that if $i > 0$ then d_{ij} has another $S^{\mathcal{J}_0}$-predecessor, $d_{(i-1)j}$, and it is important that $S^{\mathcal{J}_0}$ is not functional in d_{ij}). The resulting $\mathcal{J}_0$ is clearly a model of $\mathcal{T}_G$ and $\mathcal{J}_0 \not\models \boldsymbol{q}$.

Next, we extend $\mathcal{J}_0$ to a model $\mathcal{J}$ of $\mathcal{T}_M$ and $\mathcal{A}_{\boldsymbol{w}}$ by choosing the interpretation of concepts and roles in $\mathcal{T}_M$ on the domain of $\mathcal{J}_0$ in such a way that the part of $\mathcal{J}$ rooted in d_{11} encodes the computation of M on $\boldsymbol{w}$ (which is uniquely defined because M is deterministic). Specifically, we set $c_{ij}^{\mathcal{J}} = d_{ij}$ for all c_{ij} in $\mathcal{A}_{\boldsymbol{w}}$. Role Z follows the infinite chain of $S^{\mathcal{J}}$-successors from d_{00} and role T_* the infinite chain of $T^{\mathcal{J}}$-successors from d_{0n}. Then, the interpretation of H_q, S_a and $S_{q\sigma}$, for $q \in Q$, $a \in \Gamma$ and $\sigma \in \{-1, +1\}$, is determined by the computation assuming that the d_{ij} with $j \leq 0$ represent the blank cells (containing ␣) of the infinite extension of the tape 'before' the input, which is never visited by the head. It then should be clear how to interpret $H_\emptyset$ and $T_{q\sigma}$, for $q \in Q \cup \{\emptyset\}$ and $\sigma \in \{-1, +1\}$. As the final step of the construction of $\mathcal{J}$, we define $P_{qa}^{\mathcal{J}}$ and extend $R^{\mathcal{J}}$ as follows:

$$(d'_{ij}, d_{ij}) \in P_{q\text{␣}}^{\mathcal{J}} \text{ and } (d_{ij}, d'_{ij}) \in R^{\mathcal{J}} \quad \text{if} \quad d_{ij} \in H_q^{\mathcal{J}} \cap (\exists S_{\text{␣}}^-)^{\mathcal{J}},$$
$$(d''_{ij}, d_{ij}) \in P_{q1}^{\mathcal{J}} \text{ and } (d_{ij}, d''_{ij}) \in R^{\mathcal{J}} \quad \text{if} \quad d_{ij} \in H_q^{\mathcal{J}} \cap (\exists S_1^-)^{\mathcal{J}}.$$

It remains to show that $\mathcal{J}$ can be extended by new domain elements to satisfy $\mathcal{T}_F$ in such a way that the interpretation of concepts and roles of $\mathcal{T}_G \cup \mathcal{T}_M$ *on the domain of* $\mathcal{J}$ remains unchanged.



(a)

(b)

Figure 11: Extending $\mathcal{J}$ to $\mathcal{I}$.

Claim 13.4. *$\mathcal{J}$ can be extended to a model $\mathcal{I}$ of $\mathcal{T}_F$ so that*

(a) *$d_{ij} \in H_q^{\mathcal{I}} \cap (\exists S_a^-)^{\mathcal{I}}$ if $d_{ij} \in (\exists P_{qa}^-)^{\mathcal{I}}$, for every d_{ij};*

(b) *$A^{\mathcal{I}} \cap \Delta^{\mathcal{J}} = A^{\mathcal{J}}$ for all concept names A other than D;*

(c) *$P^{\mathcal{I}} \cap (\Delta^{\mathcal{J}} \times \Delta^{\mathcal{J}}) = P^{\mathcal{J}}$ for all role names P but R_q and L_a.*

Proof of claim. The cases of $P_{q\text{␣}}$ and P_{q1} are illustrated in Figs. 11 (a) and 11 (b), 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 in Fig. 11 (a) correspond to the three sets of positive atoms of $\boldsymbol{q}$ so that the negated inequality atom, $(u' = v')$, 'identifies' certain domain elements of the pattern. Similarly, the two black (dashed and dotted) patterns of edges in Fig. 11 (b) correspond to the two sets of positive atoms of $\boldsymbol{q}$ that 'identify' certain domain elements.

Black nodes are in the domain of $\mathcal{J}$, whereas white nodes are in the domain of $\mathcal{I}$ proper. It can be seen that the domain elements d_{ij} in $\mathcal{J}$ are subject only to the following modifications: each d_{ij}, for $i > 0$, is added to $D^{\mathcal{I}}$ and, depending on the a in the role S_a with $d_{ij} \in (\exists S_a^-)^{\mathcal{J}}$, either (d_{ij}, d'_{ij}) or (d_{ij}, d''_{ij}) is added to both $R_q^{\mathcal{I}}$ and $L_a^{\mathcal{I}}$ (which do not occur anywhere but 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

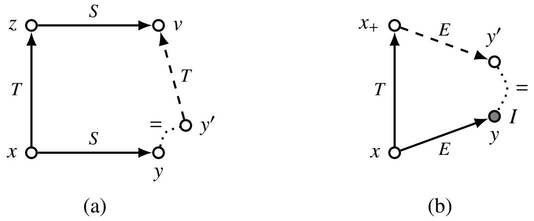


Figure 12: CQs$^{\neq}$ in the proof of Theorem 14.

that accepts $\boldsymbol{w}$ iff the empty input is accepted by the Turing machine encoded by $\boldsymbol{w}$. This finishes the proof of Theorem 13. ❑

4.2. Hardness of CQs with Inequalities over *DL-Lite*$_{\text{core}}$

In the previous section we established undecidability of CQ$^{\neq}$ answering over *DL-Lite*$^{\mathcal{H}}_{\text{core}}$. The reduction, however, essentially uses role inclusions. Leaving decidability of CQ$^{\neq}$ answering over *DL-Lite*$_{\text{core}}$ as an open problem, we establish undecidability of answering unions of three CQs$^{\neq}$, as well as P- and coNP-hardness of answering single CQs$^{\neq}$.

Theorem 14. *There exist a union of three Boolean CQs$^{\neq}$ $\boldsymbol{q}$ with one inequality each and a DL-Lite$_{\text{core}}$ TBox $\mathcal{T}$ such that* CertainAnswers$(\boldsymbol{q}, \mathcal{T})$ *is undecidable.*

Proof. We adapt the ideas of the proof of Theorem 9 to the case of inequalities and provide here a sketch of the reduction of the halting problem for deterministic Turing machines.

Let $M = (\Gamma, Q, q_0, q_1, \delta)$ be a deterministic Turing machine; see the proof of Theorem 3. Similarly to the proof of Theorem 9, we associate with a computation a two-dimensional grid on roles S and T, where representations of the cells on the tape are related by role E to individuals e_{qa}, for $(q, a) \in (Q \cup \{\emptyset, *\}) \times \Gamma$ (recall that $\emptyset$ is a no-head marker and $*$ is a marker for initialising the tape beyond the input). We use the same ABox as in Theorem 9, comprising $\mathcal{A}_M$ to encode the instructions of M (via quadruples $\mathfrak{T}_M$) and $\mathcal{A}_{\boldsymbol{w}}$ to encode an input $\boldsymbol{w} = a_1, \dots, a_n \in \Gamma^*$.

Consider a union $\boldsymbol{q}$ of the following three CQs$^{\neq}$ given in negated form (see Fig. 12 for the first and the third; the second is similar to the one in Fig. 5 (a)):

$$S(x, y) \land T(x, z) \land S(z, v) \land T(y', v) \quad \to \quad (y = y'),$$

$$\begin{aligned} & E(x, y) \land P(y, z) \land S(x, x') \land P'(z, y') \land E(x', y'') \land \\ & \qquad T(x_-, x) \land E(x_-, y_-) \land P_-(y_-, z) \land \\ & \qquad T(x, x_+) \land E(x_+, y_+) \land P_+(y_+, z) \quad \to \quad (y' = y''), \end{aligned}$$

$$T(x, x_+) \land E(x, y) \land I(y) \land E(x_+, y') \quad \to \quad (y = y').$$

Observe that queries (26)–(28) from the proof of Theorem 9 are all similarly transformed as follows: in (26), for example, the conclusion of the implication, $T(y, v)$, is moved into the premise, then one of its variables, y, is replaced with a fresh copy, y', and an equality between the variable and its copy, $y = y'$, is placed in the conclusion. The resulting queries (if viewed in negated form) can 'identify' certain points in an interpretation but require an extended TBox to achieve the effect of queries (26)–(28) with safe negation. To this end, let TBox $\mathcal{T}$ contain

$$\begin{aligned} & \exists S^- \sqsubseteq \exists T^-, \qquad \exists T \sqsubseteq \exists E, \\ & \qquad\qquad \exists T \sqsubseteq \exists S, \qquad \exists T^- \sqsubseteq \exists T, \qquad \exists E^- \sqcap N \sqsubseteq \bot. \end{aligned}$$

The first two concept inclusions allow the components of query $\boldsymbol{q}$ to play the role of (26)–(28) in Theorem 9: they enforce any model to contain matches for the atoms moved from the conclusions to the premises, and then the (negated) inequalities reconnect the other ends in the model (these atoms are indicated by the dashed arrows in Fig. 12). Finally, note that the last three concept inclusions are the same as in the proof of Theorem 9.

It can be verified that $(\mathcal{T}, \mathcal{A}_M \cup \mathcal{A}_{\boldsymbol{w}}) \not\models \boldsymbol{q}$ iff M does not accept $\boldsymbol{w}$. We just note that, in any model $\mathcal{I}$ with $\mathcal{I} \not\models \boldsymbol{q}$, the relation $(T^-)^{\mathcal{I}}$ is functional in all points with an $(S^-)^{\mathcal{I}} \circ T^{\mathcal{I}} \circ S^{\mathcal{I}}$-predecessor but $T^{\mathcal{I}}$ does not have to be functional anywhere (in fact, c_0 has a T-loop and another T-successor, c_1, in $\mathcal{A}_M$). ❑

Theorem 15. *There exist a Boolean CQ$^{\neq}$ $\boldsymbol{q}$ with one inequality and a DL-Lite$_{\text{core}}$ TBox $\mathcal{T}$ such that the problem* CertainAnswers$(\boldsymbol{q}, \mathcal{T})$ *is* P*-hard.*

Proof. We first show how the proof of Lemma 10, which shows P-hardness of answering GNCQs with one negated atom over *DL-Lite*$_{\text{core}}$, can be also adapted for the case of inequalities. Recall that the proof is by reduction of the complement of Horn-3SAT, the satisfiability problem for Horn clauses with at most three literals.

Suppose we are given a conjunction ψ of Horn clauses of the form p, $\neg p$ and $p_1 \land p_2 \to p$, where p, p_1 and p_2 are propositional variables. Consider the following Boolean CQ$^{\neq}$ $\boldsymbol{q}_1$ in negated form:

$$\begin{aligned} & N_1(x_1, y) \land E(x_1, v) \land N_2(x_2, y) \land E(x_2, v) \land V(v) \land \\ & \qquad\qquad\qquad\qquad R(y, z) \land E(z, v') \quad \to \quad (v = v'). \end{aligned}$$

This query follows the pattern of the GNCQ in the proof of Lemma 10, where unary predicate V served as a marker for variables p that are true in all models of ψ. In this case, we use *binary* predicate E to connect all such variables p to a single fixed domain element in V, which represents *truth* (as, e.g., in the proof of Theorem 14). So, we take $\mathcal{T}_1$ that contains

$$\exists R^- \sqsubseteq \exists E \qquad \text{and} \qquad V \sqcap F \sqsubseteq \bot,$$

and let $\mathcal{A}_{\psi,1}$ consist of $V(e_{true})$, $F(e_{false})$ and, for each clause γ in ψ, the following assertions:

$$\begin{aligned} E(c_p, e_{true}), &\quad \text{if } \gamma = p, \\ E(c_p, e_{false}), &\quad \text{if } \gamma = \neg p, \\ N_1(c_{p_1}, c_\gamma),\ N_2(c_{p_2}, c_\gamma),\ R(c_\gamma, c_p), &\quad \text{if } \gamma = p_1 \land p_2 \to p, \end{aligned}$$

where c_p and c_γ are individual names for every p and γ, respectively, and e_{true} and e_{false} are the individual names for *truth* and *falsum*. (Without loss of generality, we assume that ψ does not contain both p and $\neg p$, for the same variable p.) It can be verified that $(\mathcal{T}_1, \mathcal{A}_{\psi,1}) \not\models \boldsymbol{q}_1$ iff ψ is satisfiable. Note that, if the

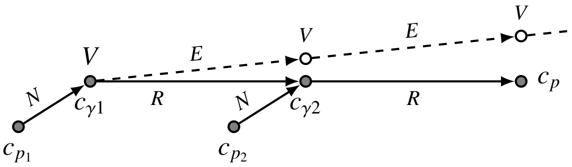


Figure 13: Proof of Theorem 15.

UNA is adopted, then the negative concept inclusion in $\mathcal{T}_1$ is not required.

Next, we provide an alternative proof of this theorem, which uses a shorter query. It is also by reduction of the complement of Horn-3SAT. Given a conjunction ψ as above, fix a TBox $\mathcal{T}$ containing

$$V \sqsubseteq \exists E, \qquad \exists E^- \sqsubseteq V, \qquad V \sqcap F \sqsubseteq \bot,$$

and a Boolean CQ$^{\neq}$ $\boldsymbol{q}$ with negated form

$$V(x) \land N(x,y) \land R(y,z) \land E(y,z') \;\;\to\;\; (z = z').$$

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 $p_1 \land p_2 \to p$ in ψ, and contains the following assertions, for every clause γ in ψ:

$$\begin{aligned} V(c_p), &\quad \text{if } \gamma = p,\\ F(c_p), &\quad \text{if } \gamma = \neg p,\\ N(c_{p_1}, c_{\gamma 1}),\ R(c_{\gamma 1}, c_{\gamma 2}),\ V(c_{\gamma 1}), &\\ N(c_{p_2}, c_{\gamma 2}),\ R(c_{\gamma 2}, c_p), &\quad \text{if } \gamma = p_1 \land p_2 \to p. \end{aligned}$$

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 $p_1 \land p_2 \to p$, the model $\mathcal{I}$ contains a configuration depicted in Fig. 13 (the grey nodes represent ABox individuals and the white ones—anonymous individuals generated by the TBox). If $c_{p_1}^{\mathcal{I}} \in V^{\mathcal{I}}$ then the $E^{\mathcal{I}}$- and $R^{\mathcal{I}}$-successors of $c_{\gamma 1}^{\mathcal{I}}$ coincide, whence $c_{\gamma 2}^{\mathcal{I}} \in V^{\mathcal{I}}$, which triggers the second 'application' of the query to identify $c_p^{\mathcal{I}}$ with the $E^{\mathcal{I}}$-successor of $c_{\gamma 2}^{\mathcal{I}}$ resulting in $c_p^{\mathcal{I}} \in V^{\mathcal{I}}$ but *only if* $c_{p_2}^{\mathcal{I}} \in V^{\mathcal{I}}$. So, as follows from the argument above, we can define a satisfying assignment $\mathfrak{a}$ for ψ by taking $\mathfrak{a}(p)$ true iff $c_p^{\mathcal{I}} \in V^{\mathcal{I}}$.

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

Proof. We begin with a remark that we could follow the lines of the first proof of Theorem 15 and adapt the proof of Lemma 11, which is by reduction of 2+2SAT, the satisfiability problem for clauses with two negative and two positive literals. This would require the following query in negated form:

$$\begin{aligned} N_1(x_1,y) \land E(x_1,v) \land N_2(x_2,y) \land E(x_2,v) \land V(v) \land{}&\\ R_1(y,z_1) \land E(z_1,v_1) \land R_2(y,z_2) \land E(z_2,v_2)&\\ \to\;\; (v = v_1) \;\lor\; (v = v_2),& \end{aligned}$$

and the following TBox:

$$\exists R_i^- \sqsubseteq \exists E, \quad \text{for } i = 1,2, \qquad \text{and} \qquad V \sqcap F \sqsubseteq \bot.$$

Instead, we provide an alternative proof with a larger TBox but a shorter query.

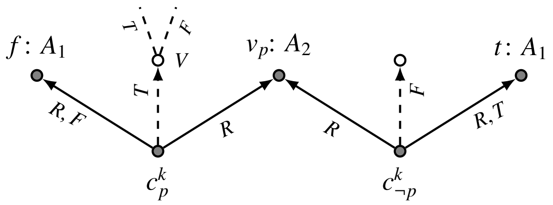


Figure 14: Proof of Theorem 16.

The proof is by reduction of the complement of 3SAT, which is known to be coNP-complete; see e.g., (Papadimitriou, 1994). Suppose we are given a conjunction ψ of clauses of the form $\ell_1 \lor \ell_2 \lor \ell_3$, 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:

$$\begin{aligned} V \sqsubseteq \exists T \sqcap \exists F, \qquad \exists T^- \sqsubseteq V,&\\ \exists T^- \sqcap \exists F^- \sqsubseteq \bot, \qquad A_1 \sqcap A_2 \sqsubseteq \bot,& \end{aligned}$$

and a Boolean CQ$^{\neq}$ $\boldsymbol{q}$ with the following negated form:

$$\begin{aligned} V(x) \land R(x,y) \land T(x,y_1) \land F(x,y_2)&\\ \to\;\; (y = y_1) \;\lor\; (y = y_2).& \end{aligned}$$

Claim 16.1. *Let $\mathcal{I}$ be a model of $\mathcal{T}$ with $\mathcal{I} \not\models \boldsymbol{q}$. If $d \in V^{\mathcal{I}}$ and $(d,d_1),(d,d_2) \in R^{\mathcal{I}}$ with $d_1 \neq d_2$ 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. Since $\mathcal{I} \not\models \boldsymbol{q}$, each pair (d,d_k) belongs either to $T^{\mathcal{I}}$ or $F^{\mathcal{I}}$. To prove the claim, suppose to the contrary that $(d,d_k) \in T^{\mathcal{I}}$ for both $k = 1,2$ (the other case, with both pairs in $F^{\mathcal{I}}$, is similar). Consider a map π with $\pi(x) = d$, $\pi(y) = d_1$, $\pi(y_1) = d_2$ and an $F^{\mathcal{I}}$-successor of d as $\pi(y_2)$. Since π cannot be a match for $\boldsymbol{q}$ in $\mathcal{I}$ but $d_1 \neq d_2$, we must have $y = y_2$, 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 five individuals v_p, $c^k_{\neg p}$ and c^k_p:

$$\begin{aligned} &A_2(v_p), \qquad && R(c^k_p, v_p),\ R(c^k_p, f),\ F(c^k_p, f),\\ & && R(c^k_{\neg p}, v_p),\ R(c^k_{\neg p}, t),\ T(c^k_{\neg p}, t),\end{aligned}$$

where the c^k_p and $c^k_{\neg p}$ represent the literals p and $\neg p$, respectively, see Fig. 14.

Let $\mathcal{I}$ be a model of $(\mathcal{T}, \mathcal{A}_\psi)$ with $\mathcal{I} \not\models \boldsymbol{q}$. Observe that $v^\mathcal{I}_p \neq t^\mathcal{I}$. By Claim 16.1, if $(c^k_{\neg p})^\mathcal{I} \in V^\mathcal{I}$ then $v^\mathcal{I}_p \in (\exists F^-)^\mathcal{I}$, that is, if the literal $\neg p$ is chosen (by means of V) then p must be false. Conversely, if $\neg p$ is not chosen (that is, $(c^k_{\neg p})^\mathcal{I} \notin V^\mathcal{I}$) then $v^\mathcal{I}_p$ does not have to be in $(\exists F^-)^\mathcal{I}$ and p can be either true or false. Similarly for $(c^k_p)^\mathcal{I}$ with $v^\mathcal{I}_p \in (\exists T^-)^\mathcal{I}$.

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

$$\begin{aligned} &V(c_{\gamma 1}), \qquad && R(c_{\gamma 1}, c^1_{\ell_1}),\ A_1(c^1_{\ell_1}),\ R(c_{\gamma 1}, c_{\gamma 2}),\ A_2(c_{\gamma 2}),\\ & && R(c_{\gamma 2}, c^1_{\ell_2}),\ A_1(c^1_{\ell_2}),\ R(c_{\gamma 2}, c^2_{\ell_3}),\ A_2(c^2_{\ell_3}).\end{aligned}$$

It can be verified 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 16.1 and the observation above, we can construct a satisfying assignment $\mathfrak{a}$ for ψ by taking $\mathfrak{a}(p)$ true iff $v^\mathcal{I}_p \in V^\mathcal{I}$. The converse direction is straightforward.

Note that the construction can be simplified if the UNA is adopted: in this case, there is no need for A_1, A_2 and the two copies of the individuals c^k_ℓ, for $k = 1, 2$, representing literals. ❑

4.3. Local CQs$^{\neq}$ over DL-Lite$^{\mathcal{H}}_{\text{core}}$: Decidability

In this section we identify a restriction on CQs$^{\neq}$ and *DL-Lite*$^{\mathcal{H}}_{\text{core}}$ TBoxes with decidable query answering problem. In a nutshell, decidability is attained by ensuring that each inequality has a term that can only be matched by ABox individuals.

Let $\mathcal{T}$ be a *DL-Lite*$^{\mathcal{H}}_{\text{core}}$ TBox. A basic concept B is said to be *$\mathcal{T}$-local* if there is no existential restriction $\exists R$ occurring on the right-hand side of a concept inclusion in $\mathcal{T}$ such that

$$\mathcal{T} \models \exists R^- \sqsubseteq B.$$

Intuitively, this condition guarantees that B contains only individuals in the canonical interpretation.

Definition 17. *A CQ$^{\neq}$ $\boldsymbol{q}$ is $\mathcal{T}$-*local *(or* local *when $\mathcal{T}$ is clear from the context) if, for each inequality $y_1 \neq y_2$ between existentially quantified variables y_1 and y_2 in $\boldsymbol{q}$, the query also contains either $B(y_1)$ or $B(y_2)$ such that B is a $\mathcal{T}$-local basic concept.*

Recall that we say that $\boldsymbol{q}$ contains $B(y)$, for $B = \exists R$, if it contains $R(y, t)$, for some term t. Remarkably, local CQs$^{\neq}$ can express quite complex patterns: see the proofs of Theorems 15 and 16; on the other hand, the first component of the union in the proof of Theorem 14 is not local (but the other two components are).

To establish decidability of query answering we require the following notions. Given two interpretations $\mathcal{J}$ and $\mathcal{I}$, we say that $\mathcal{J}$ is a *sub-interpretation* of $\mathcal{I}$ and write $\mathcal{J} \subseteq \mathcal{I}$ if $\Delta^\mathcal{J} \subseteq \Delta^\mathcal{I}$ and $\cdot^\mathcal{J}$ is the restriction of $\cdot^\mathcal{I}$ onto $\Delta^\mathcal{J}$; in particular, $c^\mathcal{J} = c^\mathcal{I} \in \Delta^\mathcal{J}$, for all individuals c.

Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a *DL-Lite*$^{\mathcal{H}}_{\text{core}}$ knowledge base. The set of interpretations d_c of individuals c in the canonical interpretation $\mathcal{C}_\mathcal{K}$ of $\mathcal{K}$ is denoted by $\mathsf{ind}_\mathcal{K}$. A *branch* $\mathfrak{b}$ is a (finite or infinite) sequence $d_c, d_{cR_1}, d_{cR_1R_2}, \dots$ of elements in $\Delta^{\mathcal{C}_\mathcal{K}}$ such that it cannot be extended to a longer sequence of this form in $\mathcal{C}_\mathcal{K}$. A *trim* of the canonical interpretation $\mathcal{C}_\mathcal{K}$ is an interpretation $\mathcal{J} \subseteq \mathcal{C}_\mathcal{K}$ whose domain $\Delta^\mathcal{J}$ is closed in the following sense: $d_w \in \Delta^\mathcal{J}$ whenever $d_{wR} \in \Delta^\mathcal{J}$. Observe that, on the one hand, the first element of every branch is in $\mathsf{ind}_\mathcal{K}$; on the other hand, by the definition of the sub-interpretation, the domain $\Delta^\mathcal{J}$ contains $\mathsf{ind}_\mathcal{K}$. Hence, the first element of every branch belongs to $\mathcal{J}$. A branch $\mathfrak{b}$ is said to be *complete in $\mathcal{J}$* if each element of $\mathfrak{b}$ is in $\Delta^\mathcal{J}$. The number of elements of $\mathfrak{b}$ in $\Delta^\mathcal{J}$, which may be infinite, is denoted by $|\mathfrak{b}|_\mathcal{J}$; if $\mathfrak{b}$ is complete in $\mathcal{J}$ then $|\mathfrak{b}|_\mathcal{J}$ is its length.

The *image $h(\mathcal{J})$* of a trim $\mathcal{J}$ under a mapping h from the domain of $\mathcal{J}$ is an interpretation defined by taking

$$\begin{aligned} &\Delta^{h(\mathcal{J})} = \{\, h(d) \mid d \in \Delta^\mathcal{J} \,\}, && \\ &c^{h(\mathcal{J})} = h(c^\mathcal{J}), && \text{for individual names } c,\\ &A^{h(\mathcal{J})} = \{\, h(d) \mid d \in A^\mathcal{J} \,\}, && \text{for concept names } A,\\ &P^{h(\mathcal{J})} = \{\, (h(d), h(d')) \mid (d, d') \in P^\mathcal{J} \,\}, && \text{for role names } P.\end{aligned}$$

Let $\mathcal{I}$ be the image $h(\mathcal{J})$ of $\mathcal{J}$ under a mapping h. By definition, h is a surjective homomorphism from $\mathcal{J}$ onto $\mathcal{I}$, and so we often write $h\colon \mathcal{J} \to \mathcal{I}$ to indicate that $\mathcal{I}$ is the image of $\mathcal{J}$ under h. We say that h is an *identification* if each $d \in \Delta^\mathcal{I} \setminus h(\mathsf{ind}_\mathcal{K})$ has at most one pre-image. Note that only interpretations of individuals, that is, elements in $h(\mathsf{ind}_\mathcal{K})$, can have multiple pre-images in an identification h. It is readily verified that, for every identification $h\colon \mathcal{J} \to \mathcal{I}$, we have the following partial converse of the homomorphism condition:

(id) if $(d_1, d_2) \in R^\mathcal{I}$, for a role R, and $d_1 \notin h(\mathsf{ind}_\mathcal{K})$ then there is a unique d_w in $\mathcal{J}$ such that either

$$\begin{aligned} &d_1 = h(d_w),\ d_2 = h(d_{wS}) \ \text{ and } \ \mathcal{T} \models S \sqsubseteq R,\\ \text{or } &d_1 = h(d_{wS}),\ d_2 = h(d_w) \ \text{ and } \ \mathcal{T} \models S \sqsubseteq R^-,\end{aligned}$$

for some role S.

Let $k > 0$ and $h\colon \mathcal{J} \to \mathcal{I}$ be an identification for a trim $\mathcal{J}$. We define the equivalence relation $\sim^h_k$ on elements of $\mathcal{J}$ by taking $d_{w'} \sim^h_k d_{w''}$ iff the following two conditions hold for every w with $|w| \leq k$:

(eq-t) $d_{w'w}$ is in $\mathcal{J}$ iff $d_{w''w}$ is in $\mathcal{J}$;

(eq-c) if $d_{w'w}$ is in $\mathcal{J}$ then either $h(d_{w'w}) = h(d_{w''w}) \in h(\mathsf{ind}_\mathcal{K})$ or $h(d_{w'w}), h(d_{w''w}) \notin h(\mathsf{ind}_\mathcal{K})$.

A pair $(d_{w_1}, d_{w_1w_2})$ of distinct elements in $\mathcal{J}$ is called a *k-block under h* in case $d_{w_1} \sim^h_k d_{w_1w_2}$ and $d_{w'_1} \not\sim^h_k d_{w'_1w'_2}$, for any distinct proper prefixes w'_1 and $w'_1w'_2$ of w_1w_2. It should be clear that each equivalence class is determined by a tree of depth k

and branching factor of at most $|\mathcal{T}|$, each element of which indicates that it does not belong to $\mathcal{J}$, or it belongs to $\mathcal{J}$ but its h-image is not in $\mathsf{ind}_{\mathcal{K}}$, or it belongs to $\mathcal{J}$ and its h-image coincides with one of the $\mathsf{ind}_{\mathcal{K}}$. This gives rise to at most $(2+|\mathcal{A}|)^{|\mathcal{T}|^k}$ equivalence classes. Therefore, under any identification, every sufficiently long branch of the canonical interpretation has a k-block simply because some equivalence class will have to appear twice on the branch.

Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a consistent *DL-Lite*$^{\mathcal{H}}_{\text{core}}$ KB and $\boldsymbol{q}$ a $\mathcal{T}$-local Boolean CQ$^{\neq}$. (Recall that queries can contain individual names, and so, without loss of generality, we may assume that the query does not have answer variables.) An interpretation $\mathcal{I}$ is called a *k-certificate for $\boldsymbol{q}$ and $\mathcal{K}$* if

- $\mathcal{I} \not\models \boldsymbol{q}$,
- $\mathcal{I}$ satisfies all negative inclusions in $\mathcal{T}$,
- there is a trim $\mathcal{J}$ of $C_{\mathcal{K}}$ and an identification $h\colon \mathcal{J} \to \mathcal{I}$ such that, for each branch $\mathfrak{b}$ in $C_{\mathcal{K}}$,
  - ($\mathbf{b}_1$) if $\mathfrak{b}$ is complete in $\mathcal{J}$ and contains a k-block $(d_{w_1}, d_{w_1 w_2})$ under h then $|\mathfrak{b}|_{\mathcal{J}} \leq |w_1 w_2| + k$;
  - ($\mathbf{b}_2$) if $\mathfrak{b}$ is incomplete in $\mathcal{J}$ then it contains a k-block $(d_{w_1}, d_{w_1 w_2})$ under h and $|\mathfrak{b}|_{\mathcal{J}} = |w_1 w_2| + k$.

Note that the trim $\mathcal{J}$ in the definition is finite because every branch has a k-block and the trim contains at most $|\mathcal{T}|^k$ elements beyond each k-block. It follows that any k-certificate is finite by definition.

Having these definitions at hand, we are ready to state and prove two key lemmas of this section.

Lemma 18. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a consistent DL-Lite$^{\mathcal{H}}_{\text{core}}$ KB, $\boldsymbol{q}$ a $\mathcal{T}$-local Boolean CQ$^{\neq}$ and $k > 0$. If $\mathcal{K} \not\models \boldsymbol{q}$ then there exists a k-certificate for $\boldsymbol{q}$ and $\mathcal{K}$.*

Proof. Let $\mathcal{K} \not\models \boldsymbol{q}$. Then there exists a model $\mathcal{I}_0$ of $\mathcal{K}$ such that $\mathcal{I}_0 \not\models \boldsymbol{q}$. Let h_0 be a homomorphism from the canonical interpretation $C_{\mathcal{K}}$ to $\mathcal{I}_0$ (without loss of generality we assume that the domain of $\mathcal{I}_0$ is disjoint from the domain of $C_{\mathcal{K}}$). The homomorphism h_0 can be represented as a composition $h' \circ h$ of two mappings such that h agrees with h_0 on all elements that are merged with images of individuals but is the identity on all other elements:

$$h(d) = \begin{cases} h_0(d), & \text{if } h_0(d) \in h_0(\mathsf{ind}_{\mathcal{K}}), \\ d, & \text{otherwise;} \end{cases}$$

it follows that h' is the identity on the interpretations of individuals and agrees with h_0 on all other elements. Let $\mathcal{I} = h(C_{\mathcal{K}})$. By definition, h and h' are homomorphisms from $C_{\mathcal{K}}$ to $\mathcal{I}$ and from $\mathcal{I}$ to $\mathcal{I}_0$, respectively; moreover, $h\colon C_{\mathcal{K}} \to \mathcal{I}$ is an identification. We have $\mathcal{I} \not\models \boldsymbol{q}$ for otherwise $\mathcal{I} \models \boldsymbol{q}$ would imply $\mathcal{I}_0 \models \boldsymbol{q}$ because h' is a homomorphism that does not identify anything with the interpretations of individuals and $\boldsymbol{q}$ is $\mathcal{T}$-local.

Consider the (finite) trim $\mathcal{J}$ of $C_{\mathcal{K}}$ to all the elements d_w such that $|w| \leq |w_1 w_2| + k$ for all k-blocks $(d_{w_1}, d_{w_1 w_2})$ under h with $w_1 w_2$ being a prefix of w (in particular, d_w is included if there is no such k-block). Let $\mathcal{I}_* = h(\mathcal{J})$. We claim that $\mathcal{I}_*$ is a k-certificate for $\boldsymbol{q}$ and $\mathcal{K}$. Indeed, since $\mathcal{I}_* \subseteq \mathcal{I}$, we have $\mathcal{I}_* \not\models \boldsymbol{q}$ and $\mathcal{I}_*$ satisfies all negative inclusions in $\mathcal{T}$. On the other hand, all the k-blocks under h are also k-blocks under the restriction of h onto $\mathcal{J}$: indeed, $\mathcal{J}$ contains all the elements within the distance of k from k-blocks, therefore satisfying **(eq-t)** (and **(eq-c)** is inherited from $\mathcal{I}$). ❑

Lemma 19. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a consistent DL-Lite$^{\mathcal{H}}_{\text{core}}$ KB and $\boldsymbol{q}$ a $\mathcal{T}$-local Boolean CQ$^{\neq}$. Let k be the size of $\boldsymbol{q}$. If there exists a k-certificate for $\boldsymbol{q}$ and $\mathcal{K}$ then $\mathcal{K} \not\models \boldsymbol{q}$.*

Proof. Let $\mathcal{I}_0$ be a k-certificate for $\boldsymbol{q}$ and $\mathcal{K}$. Although $\mathcal{I}_0 \not\models \boldsymbol{q}$, the interpretation $\mathcal{I}_0$ may not be a model of $\mathcal{K}$. We show how to extend $\mathcal{I}_0$ to a model of $\mathcal{K}$ without introducing a match for $\boldsymbol{q}$.

Since $\mathcal{I}_0$ is a k-certificate, there is a trim $\mathcal{J}_0$ of the canonical interpretation $C_{\mathcal{K}}$ and an identification $h_0\colon \mathcal{J}_0 \to \mathcal{I}_0$ satisfying ($\mathbf{b}_1$) and ($\mathbf{b}_2$). In the sequel, for the sake of simplifying the presentation, we will often refer to k-blocks under h_0 simply as k-blocks.

For $\ell > 0$, denote by $\mathcal{J}_\ell$ the trim of $C_{\mathcal{K}}$ to all the elements $d_{ww'}$ such that $d_w \in \Delta^{\mathcal{J}_0}$ and $|w'| \leq \ell$ (the trim $\mathcal{J}_\ell$ extends all branches of $\mathcal{J}_0$ by at most ℓ elements).

Claim 19.1. *Let $(d_{w_1}, d_{w_1 w_2})$ be a k-block under h_0 and let $\ell > 0$. If $d_{w_1 w_2 w}$ belongs to $\mathcal{J}_\ell$ then $d_{w_1 w}$ belongs to $\mathcal{J}_{\ell-1}$.*

Proof of claim. By the definition of the canonical interpretation, since $d_{w_1 w_2 w}$ belongs to $\mathcal{J}_\ell \subseteq C_{\mathcal{K}}$, the element $d_{w_1 w}$ also belongs to $C_{\mathcal{K}}$. If $d_{w_1 w}$ belongs to $\mathcal{J}_0$ then it clearly belongs to $\mathcal{J}_{\ell-1}$. Otherwise, all the branches containing $d_{w_1 w}$ are incomplete in $\mathcal{J}_0$. Consider any of these branches. By ($\mathbf{b}_2$), there exists a k-block $(d_{w'_1}, d_{w'_1 w'_2})$ on this branch. We know that $(d_{w_1}, d_{w_1 w_2})$ is the first pair with $d_{w_1} \sim^{h_0}_{k} d_{w_1 w_2}$ on any branch containing $d_{w_1 w_2 w}$ and so, w_1 is a proper prefix of $w'_1 w'_2$, whence $|w_1| < |w'_1 w'_2|$. On the other hand, $d_{w_1 w_2 w}$ belongs to $\mathcal{J}_\ell$ and so, $|w| \leq k + \ell$. Thus, $|w_1 w| < |w'_1 w'_2| + k + \ell$, or equivalently, $|w_1 w| \leq |w'_1 w'_2| + (k + \ell - 1)$. However, by ($\mathbf{b}_1$) and ($\mathbf{b}_2$), the trim $\mathcal{J}_0$ contains all k-blocks together with all the elements within the distance of k from the k-blocks. Therefore, $\mathcal{J}_{\ell-1}$ contains all elements of $C_{\mathcal{K}}$ that are within $k + \ell - 1$ steps from any k-block. In particular, $\mathcal{J}_{\ell-1}$ contains $d_{w_1 w}$. ❚

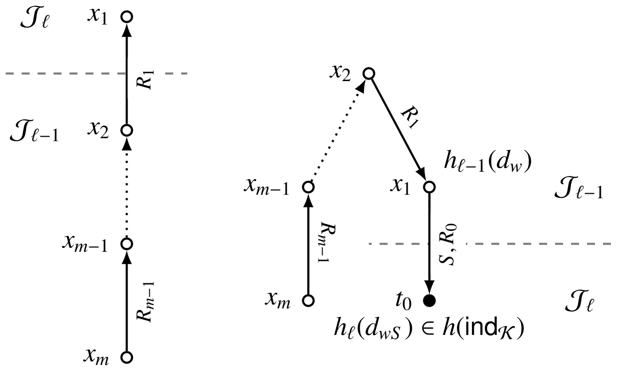
We construct a sequence of interpretations

$$\mathcal{I}_0 \subseteq \mathcal{I}_1 \subseteq \cdots \subseteq \mathcal{I}_\ell \subseteq \cdots$$

with identifications $h_\ell\colon \mathcal{J}_\ell \to \mathcal{I}_\ell$ and show by induction that, for all $\ell \geq 0$, the interpretation $\mathcal{I}_\ell$ satisfies all negative inclusions in $\mathcal{T}$ and $\mathcal{I}_\ell \not\models \boldsymbol{q}$.

The basis of induction, $\ell = 0$, is by the definition of k-certificate: $\mathcal{I}_0 = h_0(\mathcal{J}_0)$. Let $\ell > 0$, and suppose that $\mathcal{J}_{\ell-1}$ and $\mathcal{I}_{\ell-1} = h_{\ell-1}(\mathcal{J}_{\ell-1})$ have been constructed. To obtain h_ℓ, we extend $h_{\ell-1}$ to the elements $d_{w_1 w_2 w}$ in $\mathcal{J}_\ell$ that are not in $\mathcal{J}_{\ell-1}$ as follows:

$$h_\ell(d_{w_1 w_2 w}) = \begin{cases} h_{\ell-1}(d_{w_1 w}), & \text{if } h_{\ell-1}(d_{w_1 w}) \in h_{\ell-1}(\mathsf{ind}_{\mathcal{K}}), \\ \text{a fresh element}, & \text{otherwise.} \end{cases}$$



(a) (b)

Figure 15: Definition of Θ.

By Claim 19.1, the definition is correct. It also follows from the definition that $h_\ell(\mathsf{ind}_{\mathcal{K}}) = h_{\ell-1}(\mathsf{ind}_{\mathcal{K}})$ and therefore, we will use $h(\mathsf{ind}_{\mathcal{K}})$ for this set in the sequel.

Claim 19.2. *Let $(d_{w_1}, d_{w_1w_2})$ be a k-block under h_0. Then, for every $d_{w_1w_2w}$ in $\mathcal{J}_\ell$, we have*

$$h_\ell(d_{w_1w_2w}) \in h(\mathsf{ind}_{\mathcal{K}}) \quad \textit{iff} \quad h_\ell(d_{w_1w_2w}) = h_{\ell-1}(d_{w_1w}).$$

Proof of claim. If $|w| \leq k$ then the claim is immediate from **(eq-c)** and the definition of h_0. If $|w| > k$ then, by **(b$_1$)** and **(b$_2$)**, $d_{w_1w_2w}$ does not belong to $\mathcal{J}_0$. By the definition of h_ℓ, either $h_\ell(d_{w_1w_2w})$ and $h_\ell(d_{w_1w})$ are equal and in $h(\mathsf{ind}_{\mathcal{K}})$ or $h_\ell(d_{w_1w_2w})$ is a fresh element, which, in particular, cannot be equal to $h_{\ell-1}(d_{w_1w})$. ❙

Let $\mathcal{I}_\ell = h_\ell(\mathcal{J}_\ell)$. Clearly, $\mathcal{I}_{\ell-1} \subseteq \mathcal{I}_\ell$ and h_ℓ is an identification. We show that $\mathcal{I}_\ell \not\models \boldsymbol{q}$. Suppose for the sake of contradiction that there is a match π for $\boldsymbol{q}$ in $\mathcal{I}_\ell$. We then construct a match π' for $\boldsymbol{q}$ in $\mathcal{I}_{\ell-1}$. To this end we require a set Θ of all variables in sequences $x_1, \dots, x_m$, $m \geq 1$, such that $\pi(x_i) \notin h(\mathsf{ind}_{\mathcal{K}})$ for $i \leq m$, $R_i(x_{i+1}, x_i) \in \boldsymbol{q}$ for $i < m$ and either

– $\pi(x_1) = h_\ell(d)$, for some d in $\mathcal{J}_\ell$ but not in $\mathcal{J}_{\ell-1}$, or

– $\boldsymbol{q}$ contains $R_0(x_1, t_0)$ with $\pi(x_1) = h_\ell(d_w)$, $\mathcal{T} \models S \sqsubseteq R_0$, $\pi(t_0) = h_\ell(d_{wS}) \in h(\mathsf{ind}_{\mathcal{K}})$ and d_{wS} in $\mathcal{J}_\ell$ but not in $\mathcal{J}_{\ell-1}$.

Intuitively, the set Θ contains exactly those variables whose images under π are reachable from the new part in $\mathcal{I}_\ell$ through anonymous elements by a chain of (images of) atoms in the query; see Figs. 15 (a) and (b) for the two cases.

Claim 19.3. *For each $x \in \Theta$, there are a unique k-block $(d_{w_1}, d_{w_1w_2})$ under h_0 and a unique non-empty w such that $d_{w_1w_2w}$ is in $\mathcal{J}_\ell$ and $\pi(x) = h_\ell(d_{w_1w_2w})$.*

Proof of claim. Let $x_1, \dots, x_m$ be a sequence of elements of Θ such that $\pi(x_i) \notin h(\mathsf{ind}_{\mathcal{K}})$, $R_i(x_{i+1}, x_i) \in \boldsymbol{q}$, for all i, and $x_m = x$.

Suppose first that $\pi(x_1) = h_\ell(d_1)$ for some d_1 in $\mathcal{J}_\ell$ but not in $\mathcal{J}_{\ell-1}$. Since $\pi(x_1) \notin h(\mathsf{ind}_{\mathcal{K}})$ and h_ℓ is an identification, such a d_1 is uniquely defined. By **(b$_2$)**, there are a unique k-block $(d_{w_1}, d_{w_1w_2})$ and a unique w^1 such that $d_1 = d_{w_1w_2w^1}$. We show by (finite) induction that, for each x_i, there is a unique w^i with

$$\pi(x_i) = h_\ell(d_{w_1w_2w^i}) \quad \text{and} \quad |w^i| \geq k + \ell - (i-1). \tag{44}$$

Since i ranges from 1 to m, it does not exceed the size of $\boldsymbol{q}$, which in turn does not exceed k and thus, $i \leq k$. For the basis of induction, $i = 1$, the unique w^1 is constructed above; moreover, since d_1 is in $\mathcal{J}_\ell$ but not in $\mathcal{J}_{\ell-1}$, we have $|w^1| = k + \ell$. For the induction step suppose that (44) holds for some $i < m$. As $\pi(x_{i+1}) \notin h(\mathsf{ind}_{\mathcal{K}})$ and $(\pi(x_{i+1}), \pi(x_i)) \in R_i^{\mathcal{I}_\ell}$, by **(id)**, there is a unique d_{i+1} with $\pi(x_{i+1}) = h_\ell(d_{i+1})$. Since $i < m \leq k$, w^i is non-empty. Hence, $d_{i+1} = d_{w_1w_2w^{i+1}}$ with either $w^{i+1} = w^iS$ or $w^{i+1}S = w^i$, for some S. Thus, $|w^{i+1}| \geq |w^i| - 1$ and (44) follows. Finally, we use (44) with $i = m$ to obtain $|w^m| > \ell$.

Suppose now that $\boldsymbol{q}$ contains $R_0(x_1, t_0)$ with $\pi(x_1) = h_\ell(d_w)$, $\pi(t_0) = h_\ell(d_{wS}) \in h(\mathsf{ind}_{\mathcal{K}})$ and d_{wS} in $\mathcal{J}_\ell$ but not in $\mathcal{J}_{\ell-1}$. The argument and the construction are identical to the case above except that now $|w^i| \geq k + \ell - i$, and thus $|w^m| \geq \ell > 0$. ❙

The mapping π' from the terms t of $\boldsymbol{q}$ to the domain of $\mathcal{I}_{\ell-1}$ is constructed as follows.

– If $t \in \Theta$ then t is a variable. By Claim 19.3, we have $\pi(t) = h_\ell(d_{w_1w_2w})$, for a k-block $(d_{w_1}, d_{w_1w_2})$ and some w. By Claim 19.1, d_{w_1w} is in $\mathcal{J}_{\ell-1}$; so, let $\pi'(t) = h_{\ell-1}(d_{w_1w})$, which is in $\Delta^{\mathcal{I}_{\ell-1}}$.

– If $t \notin \Theta$ then $\pi(t)$ is in $\Delta^{\mathcal{I}_{\ell-1}}$ (for otherwise t is in Θ); let $\pi'(t) = \pi(t)$.

We claim that π' is a match for $\boldsymbol{q}$ in $\mathcal{I}_{\ell-1}$ and prove it by showing that the image of every atom in $\boldsymbol{q}$ under π' is true in $\mathcal{I}_{\ell-1}$.

1. Suppose $(\pi(s), \pi(t)) \in R^{\mathcal{I}_\ell}$. We show $(\pi'(s), \pi'(t)) \in R^{\mathcal{I}_{\ell-1}}$. There are four cases, depending on the way $\pi'(s)$ and $\pi'(t)$ are constructed.

Case 1.1: $s, t \in \Theta$, that is, $\pi(s) = h_\ell(d_{w_1w_2w})$, $\pi'(s) = h_\ell(d_{w_1w})$, $\pi(t) = h_\ell(d_{w_1'w_2'w'})$ and $\pi'(t) = h_\ell(d_{w_1'w'})$. By Claim 19.3, both w and w' are non-empty and uniquely defined. Moreover, since neither $\pi(s)$ nor $\pi(t)$ is in $h(\mathsf{ind}_{\mathcal{K}})$, by **(id)**, we obtain $w_1 = w_1'$, $w_2 = w_2'$ and either $w' = wS$ with $\mathcal{T} \models S \sqsubseteq R$ or $w = w'S$ with $\mathcal{T} \models S \sqsubseteq R^-$. By Claim 19.1, both d_{w_1w} and $d_{w_1w'}$ belong to $\mathcal{J}_{\ell-1}$, and so, in either case, $(d_{w_1w}, d_{w_1w'}) \in R^{\mathcal{J}_{\ell-1}}$, whence $(\pi'(s), \pi'(t)) \in R^{\mathcal{I}_{\ell-1}}$.

Case 1.2: $s \in \Theta$ and $t \notin \Theta$, that is, $\pi(s) = h_\ell(d_{w_1w_2w})$ and $\pi'(s) = h_{\ell-1}(d_{w_1w})$ but $\pi'(t) = \pi(t)$. We have $\pi(t) \in h(\mathsf{ind}_{\mathcal{K}})$, for otherwise we would include t in Θ by considering a sequence ending in t. By Claim 19.3, w is non-empty and uniquely defined and so, by **(id)**, we have $\pi(t) = h_\ell(d_{w_1w_2w'})$ with either $w' = wS$ and $\mathcal{T} \models S \sqsubseteq R$ or $w'S = w$ and $\mathcal{T} \models S \sqsubseteq R^-$. By Claim 19.1, both d_{w_1w} and $d_{w_1w'}$ are in $\mathcal{J}_{\ell-1}$ and so, in either case, $(d_{w_1w}, d_{w_1w'}) \in R^{\mathcal{J}_{\ell-1}}$. By Claim 19.2, $\pi'(t) = h_{\ell-1}(d_{w_1w'})$, whence $(\pi'(s), \pi'(t)) \in R^{\mathcal{I}_{\ell-1}}$.

Case 1.3: $s \notin \Theta$ and $t \in \Theta$ is the mirror image of Case 1.2.

Case 1.4: $s, t \notin \Theta$, that is, $\pi(s) = \pi'(s)$ and $\pi(t) = \pi'(t)$. We have $(\pi(s), \pi(t)) \in R^{\mathcal{I}_\ell}$. Consider first the case when at least one of these elements is not in $h(\mathsf{ind}_{\mathcal{K}})$. Suppose that $\pi(s) \notin$

$h(\mathsf{ind}_{\mathcal{K}})$ (the other case is symmetric). By **(id)**, either $\pi(s) = h(d_w)$ and $\pi(t) = h(d_{wS})$ with $\mathcal{T} \models S \sqsubseteq R$ or $\pi(s) = h(d_{wS})$ and $\pi(t) = h(d_w)$ with $\mathcal{T} \models S \sqsubseteq R^-$. We claim that in either case d_{wS} (and so d_w) belongs to $\mathcal{J}_{\ell-1}$. Indeed, in the former case $d_{wS} = \pi(t)$ cannot be outside $\mathcal{J}_{\ell-1}$ for otherwise Θ would contain s. For the same reason, $d_{wS} = \pi(s)$ cannot be outside $\mathcal{J}_{\ell-1}$ in the latter case. So, both $\pi(s)$ and $\pi(t)$ are in $\mathcal{J}_{\ell-1}$, and we obtain $(\pi'(s), \pi'(t)) \in R^{\mathcal{I}_{\ell-1}}$.

Otherwise, both $\pi(s)$ and $\pi(t)$ are in $h(\mathsf{ind}_{\mathcal{K}})$. Suppose for the sake of contradiction that $(\pi'(s), \pi'(t)) \notin R^{\mathcal{I}_{\ell-1}}$. Then, since $\mathcal{I}_\ell = h_\ell(\mathcal{J}_\ell)$, $\mathcal{I}_{\ell-1} = h_{\ell-1}(\mathcal{J}_{\ell-1})$ and both $\mathcal{J}_\ell$, $\mathcal{J}_{\ell-1}$ are trims of $C_{\mathcal{K}}$, there are some $d_{w_1w_2w}$ and $d_{w_1w_2w'}$, for a k-block $(d_{w_1}, d_{w_1w_2})$, with one of them in $\mathcal{J}_\ell$ but not in $\mathcal{J}_{\ell-1}$ such that $\pi(s) = h_\ell(d_{w_1w_2w})$ and $\pi(t) = h_\ell(d_{w_1w_2w'})$ and either $w' = wS$ with $\mathcal{T} \models S \sqsubseteq R$ or $w = w'S$ with $\mathcal{T} \models S \sqsubseteq R^-$. By Claim 19.1, d_{w_1w} and $d_{w_1w'}$ are in $\mathcal{J}_{\ell-1}$, and so, in either case, $(d_{w_1w}, d_{w_1w'}) \in R^{\mathcal{J}_{\ell-1}}$, whence $(h_{\ell-1}(d_{w_1w}), h_{\ell-1}(d_{w_1w'})) \in R^{\mathcal{I}_{\ell-1}}$. By Claim 19.2, $h_\ell(d_{w_1w_2w}) = h_{\ell-1}(d_{w_1w})$ and $h_\ell(d_{w_1w_2w'}) = h_{\ell-1}(d_{w_1w'})$. So, $(\pi'(s), \pi'(t)) \in R^{\mathcal{I}_{\ell-1}}$ contrary to the assumption.

2. Next, suppose $\pi(s) \in A^{\mathcal{I}_\ell}$. We show $\pi'(s) \in A^{\mathcal{I}_{\ell-1}}$. There are two cases.

Case 2.1: $s \in \Theta$, that is, $\pi(s) = h_\ell(d_{w_1w_2w})$ and $\pi'(s) = h_{\ell-1}(d_{w_1w})$. By Claim 19.3, $d_{w_1w_2w}$ is uniquely defined. By the definition of Θ, $h_\ell(d_{w_1w_2w}) \notin h(\mathsf{ind}_{\mathcal{K}})$ and so, as h_ℓ is an identification, we obtain $d_{w_1w_2w} \in A^{\mathcal{J}_\ell}$. By Claim 19.1, d_{w_1w} belongs to $\mathcal{J}_{\ell-1}$, and so, by the definition of the canonical interpretation, $d_{w_1w} \in A^{\mathcal{J}_{\ell-1}}$, whence $\pi'(s) \in A^{\mathcal{I}_{\ell-1}}$.

Case 2.2: $s \notin \Theta$, that is, $\pi(s) = \pi'(s)$. If $\pi(s) \notin h(\mathsf{ind}_{\mathcal{K}})$ then, since h_ℓ is an identification, there is a unique d in $\mathcal{J}_\ell$ such that $\pi(s) = h_\ell(d)$. By the first item in the definition of Θ, d is in fact in $\mathcal{J}_{\ell-1}$. Since $\mathcal{I}_{\ell-1} = h_{\ell-1}(\mathcal{J}_{\ell-1})$, we obtain $h_{\ell-1}(d) \in A^{\mathcal{I}_{\ell-1}}$, whence $\pi'(s) \in A^{\mathcal{I}_{\ell-1}}$. If $\pi(s) \in h(\mathsf{ind}_{\mathcal{K}})$ then suppose, for the sake of contradiction, that $\pi(s) \notin A^{\mathcal{I}_{\ell-1}}$. As $\mathcal{I}_\ell = h_\ell(\mathcal{J}_\ell)$ and $\mathcal{I}_{\ell-1} = h_{\ell-1}(\mathcal{J}_{\ell-1})$ and both $\mathcal{J}_\ell$, $\mathcal{J}_{\ell-1}$ are trims of $C_{\mathcal{K}}$, there is $d_{w_1w_2w}$, for a k-block $(d_{w_1}, d_{w_1w_2})$, in $\mathcal{J}_\ell$ but not in $\mathcal{J}_{\ell-1}$ such that $\pi(s) = h_\ell(d_{w_1w_2w})$. By Claim 19.1, d_{w_1w} belongs to $\mathcal{J}_{\ell-1}$, and so, $d_{w_1w} \in A^{\mathcal{J}_{\ell-1}}$. Hence, $\pi(s) \in A^{\mathcal{I}_{\ell-1}}$ contrary to the assumption.

3. Finally, suppose $\pi(s) \neq \pi(t)$, for an inequality $s \neq t$ in $\boldsymbol{q}$. We show $\pi'(s) \neq \pi'(t)$. Since $\boldsymbol{q}$ is $\mathcal{T}$-local, either $\pi(s)$ or $\pi(t)$ must be in $h(\mathsf{ind}_{\mathcal{K}})$, and therefore either s or t is not in Θ, which leaves the following three cases possible.

Case 3.1: $s \in \Theta$ and $t \notin \Theta$, that is, $\pi(s) = h_\ell(d_{w_1w_2w})$ and $\pi'(s) = h_{\ell-1}(d_{w_1w})$ but $\pi'(t) = \pi(t)$. By the definition of Θ, $\pi(s) \notin h(\mathsf{ind}_{\mathcal{K}})$, whence, by Claim 19.2, $\pi'(s) \notin h(\mathsf{ind}_{\mathcal{K}})$. On the other hand, by the observation above, $\pi'(t) = \pi(t) \in h(\mathsf{ind}_{\mathcal{K}})$. So, $\pi'(s) \neq \pi'(t)$.

Case 3.2: $s \notin \Theta$ and $t \in \Theta$ is the mirror image of Case 3.1.

Case 3.3: $s, t \notin \Theta$, that is, $\pi(s) = \pi'(s)$ and $\pi(t) = \pi'(t)$, which, by the assumption, implies $\pi'(s) \neq \pi'(t)$.

By induction hypothesis, $\mathcal{I}_{\ell-1} \not\models \boldsymbol{q}$, and so $\mathcal{I}_\ell \not\models \boldsymbol{q}$. Moreover, by repeating the same argument, one can show that $\mathcal{I}_\ell$ satisfies all negative inclusions in $\mathcal{T}$ (the negation of a negative inclusion can be regarded as a Boolean CQ with two atoms and at most three variables, that is, as a $\mathcal{T}$-local CQ$^{\neq}$ of special form).

To complete the proof, let $\mathcal{J}$ be the union of the $\mathcal{J}_\ell$ and h be the union of the h_ℓ. It should be clear that in fact $\mathcal{J} = C_{\mathcal{K}}$. Consider $\mathcal{I} = h(\mathcal{J})$. By definition, $\mathcal{I}$ satisfies the assertions of the ABox $\mathcal{A}$ and all positive inclusions in $\mathcal{T}$. Since, by construction, each $\mathcal{I}_\ell$ satisfies all negative inclusions in $\mathcal{T}$, we can conclude that $\mathcal{I}$ is a model of $\mathcal{K}$ (note, however, that $\mathcal{I}_\ell$ may not necessarily be a model of $\mathcal{K}$, for any ℓ). Finally, by our inductive argument, $\mathcal{I} \not\models \boldsymbol{q}$. ❑

Combining Lemmas 18 and 19 and observing that the size of a k-certificate can be bounded by an exponential function (in $|\mathcal{A}|$), we obtain the following theorem.

Theorem 20. *For any DL-Lite$^{\mathcal{H}}_{\mathrm{core}}$ TBox $\mathcal{T}$ and any $\mathcal{T}$-local CQ$^{\neq}$ $\boldsymbol{q}$, the problem* CertainAnswers$(\boldsymbol{q}, \mathcal{T})$ *is decidable.*

The exponential bound on the size of k-certificates means that the problem CertainAnswers$(\boldsymbol{q}, \mathcal{T})$ for a *DL-Lite*$^{\mathcal{H}}_{\mathrm{core}}$ TBox $\mathcal{T}$ and a $\mathcal{T}$-local CQ$^{\neq}$ $\boldsymbol{q}$ is in fact in coNExpTime in data complexity, which leaves an exponential gap with the coNP-hardness established in Theorem 16. In case of a single inequality, a k-certificate of exponential size can be constructed by a deterministic algorithm. This results in the ExpTime upper data complexity bound, which is again exponentially harder than the P-hardness in Theorem 15.

Finally, we remark that the arguments in the proofs of Lemmas 18 and 19 can be transferred to *unions* of $\mathcal{T}$-local CQs$^{\neq}$, so Theorem 20 also holds for this extended class of queries.

5. Conclusions and Future Work

Our investigation in the OBDA paradigm has made further steps towards a clearer understanding of the impact of extending CQs with different forms of negation. We have shown that in general these extensions lead to a surprisingly significant increase even in the data complexity: e.g., from AC^0 for answering CQs to undecidability when safe negations are allowed. In order to find a way of having efficient query answering in the presence of negation, we have also explored various syntactic restrictions. For example, we have identified a novel class of CQs$^{\neq}$, local CQs$^{\neq}$, with decidable query answering over *DL-Lite*$^{\mathcal{H}}_{\mathrm{core}}$.

Our investigation leaves open some important problems for future work, e.g., decidability of answering CQs$^{\neg s}$ and CQs$^{\neq}$ over *DL-Lite*$_{\mathrm{core}}$, as well as of answering CQs$^{\neg s}$ and local CQs$^{\neq}$ over $\mathcal{EL}_\bot$. It also remains open to establish the exact complexity for local CQs$^{\neq}$ over *DL-Lite*$^{\mathcal{H}}_{\mathrm{core}}$.

Another interesting problem is to investigate whether the notions of guardedness and locality can be relaxed to increase the expressivity. We note that CQs$^{\neq}$ are not *finite controllable* for ontology languages with inverses, such as *DL-Lite*$_{\mathrm{core}}$, *DL-Lite*$^{\mathcal{H}}_{\mathrm{core}}$ and $\mathcal{ELI}$, and that our undecidability proofs rely on the encoding of infinite structures. Therefore, our techniques do not apply directly to the finite case. Finally, we believe that other problems, such as query containment, are also worth studying for the ontology languages with decidable query answering.

Acknowledgements. The first author was supported by the M8-Project TS-OBDA, the second author by the SFB/TR8 *Spatial Cognition* and the fourth author by the UK EPSRC grant EP/J017728 (SOCIAM project).

References

Abiteboul, S., Duschka, O. M., 1999. Complexity of answering queries using materialized views. Tech. Rep. Gemo Report 383, INRIA Saclay.

Andréka, H., Németi, I., van Benthem, J., 1998. Modal languages and bounded fragments of predicate logic. J. Philos. Logic 27, 217–274.

Arenas, M., Barceló, P., Reutter, J. L., 2011. Query languages for data exchange: Beyond unions of conjunctive queries. Theory Comput. Syst. 49 (2), 489–564.

Artale, A., Calvanese, D., Kontchakov, R., Zakharyaschev, M., 2009. The DL-Lite family and relations. J. Artif. Intell. Res. (JAIR) 36, 1–69.

Baader, F., Brandt, S., Lutz, C., 2005. Pushing the EL envelope. In: Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005). Professional Book Center, pp. 364–369.

Bárány, V., ten Cate, B., Otto, M., 2012. Queries with guarded negation. PVLDB 5 (11), 1328–1339.

Bárány, V., ten Cate, B., Segoufin, L., 2011. Guarded negation. In: Proc. of the 38th Int. Colloquium on Automata, Languages and Programming (ICALP 2011). Vol. 6756 of LNCS. Springer, pp. 356–367.

Bienvenu, M., Calvanese, D., Ortiz, M., Šimkus, M., 2014. Nested regular path queries in description logics. In: Proc. of the 14th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2014). AAAI Press.

Bienvenu, M., Ortiz, M., Šimkus, M., 2013. Conjunctive regular path queries in lightweight description logics. In: Proc. of the 23rd Int. Joint Conf. on Artificial Intelligence (IJCAI 2013). IJCAI/AAAI.

Calì, A., Gottlob, G., Orsi, G., Pieris, A., 2012. On the interaction of existential rules and equality constraints in ontology querying. In: Correct Reasoning: Essays on Logic-Based AI in Honour of Vladimir Lifschitz. Vol. 7265 of LNCS. Springer, pp. 117–133.

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R., 2007a. EQL-Lite: Effective first-order query processing in description logics. In: Proc. of the 20th Int. Joint Conf. on Artificial Intelligence (IJCAI 2007). pp. 274–279.

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R., 2007b. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. J. Autom. Reasoning 39 (3), 385–429.

Calvanese, D., De Giacomo, G., Lenzerini, M., 1998. On the decidability of query containment under constraints. In: Proc. of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'98). ACM Press, pp. 149–158.

Calvanese, D., De Giacomo, G., Lenzerini, M., 2008a. Conjunctive query containment and answering under description logic constraints. ACM Trans. Comput. Log. 9 (3), 22.

Calvanese, D., Kharlamov, E., Nutt, W., Thorne, C., 2008b. Aggregate queries over ontologies. In: Proc. of the 2nd Int. Workshop on Ontologies and Information Systems for the Semantic Web (ONISW 2008). ACM, pp. 97–104.

Deutsch, A., Nash, A., Remmel, J. B., 2008. The chase revisited. In: Proc. of the 27th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS 2008). ACM Press, pp. 149–158.

Fagin, R., Kolaitis, P. G., Miller, R. J., Popa, L., 2005. Data exchange: semantics and query answering. Theor. Comput. Sci. 336 (1), 89–124.

Grädel, E., Walukiewicz, I., 1999. Guarded fixed point logic. In: Proc. of the 14th Annual IEEE Symposium on Logic in Computer Science (LICS'99). IEEE Computer Society, pp. 45–54.

Gutiérrez-Basulto, V., Ibáñez-García, Y., Kontchakov, R., 2012. An update on query answering with restricted forms of negation. In: Proc. of the 6th Int. Conf. on Web Reasoning and Rule Systems (RR 2012). Vol. 7497 of LNCS. Springer, pp. 75–89.

Gutiérrez-Basulto, V., Ibáñez-García, Y. A., Kontchakov, R., Kostylev, E. V., 2013. Conjunctive queries with negation over DL-Lite: A closer look. In: Proc. of the 7th Int. Conf. on Web Reasoning and Rule Systems (RR 2013). Vol. 7994 of LNCS. Springer, pp. 109–122.

Hernich, A., Kupke, C., Lukasiewicz, T., Gottlob, G., 2013. Well-founded semantics for extended datalog and ontological reasoning. In: Proc. of the 32nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS 2013). ACM Press, pp. 225–236.

Kikot, S., Kontchakov, R., Zakharyaschev, M., 2012. Conjunctive query answering with OWL 2 QL. In: Proc. of the 13th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2012). AAAI Press, pp. 275–285.

Klenke, T., 2010. Über die Entscheidbarkeit von konjunktiv Anfragen mit Ungleichheit in der Beschreibungslogik $\mathcal{EL}$. Master's thesis, Universität Bremen.

Klug, A., 1988. On conjunctive queries containing inequalities. J. ACM 35 (1), 146–160.

Kontchakov, R., Lutz, C., Toman, D., Wolter, F., Zakharyaschev, M., 2010. The combined approach to query answering in DL-Lite. In: Proc. of the 12th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2010). AAAI Press, pp. 247–257.

Kostylev, E. V., Reutter, J. L., 2013. Answering counting aggregate queries over ontologies of the DL-Lite family. In: Proc. of the 27th AAAI Conf. on Artificial Intelligence (AAAI 2013). AAAI Press, pp. 534–540.

Kostylev, E. V., Reutter, J. L., Vrgoc, D., 2015. XPath for DL ontologies. In: Proc. of the 29th AAAI Conf. on Artificial Intelligence (AAAI 2015). AAAI Press, pp. 1525–1531.

Lutz, C., Seylan, I., Toman, D., Wolter, F., 2013. The combined approach to OBDA: Taming role hierarchies using filters. In: Proc. of the 12th Int. Semantic Web Conf. (ISWC 2013). Vol. 8218 of LNCS. Springer, pp. 314–330.

Lutz, C., Toman, D., Wolter, F., 2009. Conjunctive query answering in the description logic EL using a relational database system. In: Proc. of the 21st Int. Joint Conference on Artificial Intelligence (IJCAI 09). pp. 2070–2075.

Ortiz, M., Calvanese, D., Eiter, T., 2006. Characterizing data complexity for conjunctive query answering in expressive description logics. In: Proc. of the 21st Nat. Conf. on Artificial Intelligence (AAAI 2006). AAAI Press, pp. 275–280.

Papadimitriou, C. H., 1994. Computational complexity. Addison-Wesley.

Rodríguez-Muro, M., Kontchakov, R., Zakharyaschev, M., 2013. Ontology-based data access: ontop of databases. In: Proc. of the 12th Int. Semantic Web Conf. (ISWC 2013). Vol. 8218 of LNCS. Springer, pp. 558–573.

Rosati, R., 2006. On the decidability and finite controllability of query processing in databases with incomplete information. In: Proc. of the 25th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 2006). ACM Press, pp. 356–365.

Rosati, R., 2007. The limits of querying ontologies. In: Proc. of the 11th Int. Conf. on Database Theory (ICDT 2007). Vol. 4353 of LNCS. Springer, pp. 164–178.

Rossman, B., 2008. Homomorphism preservation theorems. J. ACM 55 (3).

Schaerf, A., 1993. On the complexity of the instance checking problem in concept languages with existential quantification. J. Intell. Inf. Syst. 2 (3), 265–278.

Vardi, M. Y., 1982. The complexity of relational query languages (extended abstract). In: Proc. of the 14th Annual ACM Symposium on Theory of Computing (STOC'82). ACM, pp. 137–146.