DL-Lite without UNA

A. Artale,[1] D. Calvanese,[1] R. Kontchakov,[2] and M. Zakharyaschev[2]

[1] KRDB Research Centre
Free University of Bozen-Bolzano
I-39100 Bolzano, Italy
lastname@inf.unibz.it

[2] School of Comp. Science and Inf. Sys.
Birkbeck College
London WC1E 7HX, UK
{roman,michael}@dcs.bbk.ac.uk

1 Introduction

Description logics (DLs) have recently been used to provide access to large amounts of data through a high-level conceptual interface, which is of relevance to both data integration and ontology-based data access. The fundamental inference service in this case is answering queries by taking into account the axioms in the TBox and the data stored in the ABox. The key property for such an approach to be viable in practice is the efficiency of query evaluation. To address these needs a series of description logics has been proposed and investigated in [6–8, 16] (the so-called *DL-Lite* family) and in [1, 2]. The significance of the *DL-Lite* logics is testified by the fact that they form the basis of OWL 2 QL, one of the three profiles of the web ontology language OWL 2.[1] According to the current version of the official W3C profiles document, the purpose of OWL 2 QL is to be the language of choice for applications that use very large amounts of data and where query answering is the most important reasoning task.

A very important difference between description logics and OWL is the status of the unique name assumption (UNA), which is commonly made in DL but not adopted in OWL. Instead, the OWL syntax provides explicit means for stating which object names are supposed to denote the same individual and which of them should be interpreted differently (in OWL, these constructs are called `sameAs` and `differentFrom`).

Until recently, it had been assumed that the *DL-Lite* logics are interpreted under the UNA. The role of the UNA (for *DL-Lite*$_{\mathcal{A}}$) is discussed in [4, 5], where it is shown that if the UNA is dropped, instance checking becomes NLogSpace-hard for data complexity. The aim of this paper is to investigate in depth the impact of dropping the UNA on both the combined and data complexity of reasoning in the *DL-Lite* family and extensions, classified according to the following features:

(1) the presence or absence of role inclusion assertions;
(2) the form of the allowed concept inclusions, where we consider four classes *core*, *Krom*, *Horn*, and *Bool* exhibiting different computational properties;
(3) the form of the allowed numeric constraints, ranging from none, to global functionality constraints only, and to arbitrary number restrictions.

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

In a nutshell, the obtained results can be summarized as follows. Without any kind of number restrictions, the above logics do not 'feel' the UNA. However, equality between object names increases the data complexity for core and Horn logics, $DL\text{-}Lite^{\mathcal{R}}_{core}$ and $DL\text{-}Lite^{\mathcal{R}}_{horn}$, from AC^0 to LOGSPACE. (The inequality constraints do not affect the complexity.) In the presence of functionality constraints, dropping the UNA increases the combined complexity of satisfiability for $DL\text{-}Lite^{\mathcal{F}}_{core}$ and $DL\text{-}Lite^{\mathcal{F}}_{krom}$ from NLOGSPACE to P, and the data complexity of query answering (or instance checking) for $DL\text{-}Lite^{\mathcal{F}}_{core}$ and $DL\text{-}Lite^{\mathcal{F}}_{horn}$ from AC^0 to P. With arbitrary number restrictions the price is even higher: e.g., the data complexity of query answering (or instance checking) for $DL\text{-}Lite^{\mathcal{N}}_{core}$ increases from AC^0 to coNP if the UNA is not adopted.

Needless to say that in all these cases we loose the important property of first-order rewritability of (positive existential) queries, and so cannot use the standard database query engines in a straightforward manner. Since the OWL 2 profiles are defined as syntactic restrictions without changing the basic semantic assumptions, it was chosen not to include in the OWL 2 QL profile any construct that interferes with the UNA and which, in the absence of the UNA, would cause higher complexity. That is why OWL 2 QL does not include any form of number restrictions and the `sameAs` constructor.

As for the matching upper complexity bounds, we show that, without UNA, the logics of the form $DL\text{-}Lite^{(\mathcal{RF})}_{\alpha}$ and $DL\text{-}Lite^{(\mathcal{RN})}_{\alpha}$ with restricted interaction between number restrictions and role inclusions (similar to that in $DL\text{-}Lite_{\mathcal{A}}$ [16]) behave precisely in the same way as $DL\text{-}Lite^{\mathcal{F}}_{\alpha}$ and $DL\text{-}Lite^{\mathcal{N}}_{\alpha}$, respectively. (If this interaction is not restricted then even the logic $DL\text{-}Lite^{\mathcal{R},\mathcal{F}}_{core}$ is EXPTIME-hard for combined complexity and P-hard for data complexity [13].)

2 *DL-Lite* Logics

We begin by defining the description logic $DL\text{-}Lite^{\mathcal{R},\mathcal{N}}_{bool}$, which can be regarded as the *supremum* of the original *DL-Lite* family [6–8] in the lattice of description logics. The language of $DL\text{-}Lite^{\mathcal{R},\mathcal{N}}_{bool}$ contains *object names* $a_0, a_1, \dots,$ *concept names* $A_0, A_1, \dots,$ and *role names* $P_0, P_1, \dots.$ Complex *roles* R and *concepts* C are defined as follows:

$$\begin{array}{lcllllll} R & ::= & P_i & \mid & P_i^-, & & \\ B & ::= & \bot & \mid & A_i & \mid & \geq q\, R, \\ C & ::= & B & \mid & \neg C & \mid & C_1 \sqcap C_2, \end{array}$$

where q is a positive integer. The concepts of the form B are called *basic*.

A $DL\text{-}Lite^{\mathcal{R},\mathcal{N}}_{bool}$ *TBox*, $\mathcal{T}$, is a finite set of *concept* and *role inclusions* of the form $C_1 \sqsubseteq C_2$ and $R_1 \sqsubseteq R_2$, respectively. An *ABox*, $\mathcal{A}$, is a finite set of assertions of the form $A_k(a_i), \neg A_k(a_i),\ P_k(a_i, a_j), \neg P_k(a_i, a_j)$. Taken together, $\mathcal{T}$ and $\mathcal{A}$ constitute the $DL\text{-}Lite^{\mathcal{R},\mathcal{N}}_{bool}$ *knowledge base* (KB) $\mathcal{K} = (\mathcal{T}, \mathcal{A})$. In the following, we denote by *role*($\mathcal{K}$) the set of role names occurring in $\mathcal{T}$ and $\mathcal{A}$, by

$\mathit{role}^{\pm}(\mathcal{K})$ the set $\{P_k, P_k^- \mid P_k \in \mathit{role}(\mathcal{K})\}$, and by $\mathit{ob}(\mathcal{A})$ the set of object names in $\mathcal{A}$. For a role R, we set $\mathit{inv}(R) = P_k^-$ if $R = P_k$, and $\mathit{inv}(R) = P_k$ if $R = P_k^-$.

As usual, an *interpretation*, $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, consists of a *domain* $\Delta^{\mathcal{I}} \neq \emptyset$ and an interpretation function $\cdot^{\mathcal{I}}$ that assigns to each a_i an element $a_i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, to A_i a subset $A_i^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ of the domain, and to each P_i a binary relation $P_i^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation of the concept constructs, the concept and role inclusions, and the ABox assertions is also standard. For example,

$$(\geq q\, R)^{\mathcal{I}} \;=\; \big\{x \in \Delta^{\mathcal{I}} \mid \sharp\{y \in \Delta^{\mathcal{I}} \mid (x,y) \in R^{\mathcal{I}}\} \geq q\big\}.$$

A KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is *satisfiable* if there is an interpretation $\mathcal{I}$ satisfying all the members of $\mathcal{T}$ and $\mathcal{A}$, in which case we write $\mathcal{I} \models \mathcal{K}$ and say that $\mathcal{I}$ is a *model of* $\mathcal{K}$.

The *unique name assumption* (UNA) is the following requirement imposed on interpretations $\mathcal{I}$: $a_i^{\mathcal{I}} \neq a_j^{\mathcal{I}}$ for all $i \neq j$. As was mentioned in the introduction, in this paper we *do not* make this assumption.

The languages we consider here are obtained by imposing various syntactic restrictions on $\mathit{DL\text{-}Lite}_{bool}^{\mathcal{R},\mathcal{N}}$. A $\mathit{DL\text{-}Lite}_{bool}^{\mathcal{R},\mathcal{N}}$ TBox $\mathcal{T}$ is called a *Krom TBox* if its concept inclusions are of the form

$$B_1 \;\sqsubseteq\; B_2, \qquad B_1 \;\sqsubseteq\; \neg B_2 \qquad \text{or} \qquad \neg B_1 \;\sqsubseteq\; B_2 \tag{Krom}$$

(here and below all the B_i and B are basic concepts). $\mathcal{T}$ is called a *Horn TBox* if its concept inclusions are of the form

$$\bigsqcap_k B_k \;\sqsubseteq\; B \tag{Horn}$$

(by definition, the empty conjunction is just $\top$). Finally, $\mathcal{T}$ is a *core TBox* if its concept inclusions are restricted to

$$B_1 \;\sqsubseteq\; B_2 \qquad \text{or} \qquad B_1 \;\sqsubseteq\; \neg B_2. \tag{core}$$

As $B_1 \sqsubseteq \neg B_2$ is equivalent to $B_1 \sqcap B_2 \sqsubseteq \bot$, core TBoxes can be regarded as sitting in the intersection of Krom and Horn TBoxes. The fragments of $\mathit{DL\text{-}Lite}_{bool}^{\mathcal{R},\mathcal{N}}$ with Krom, Horn and core TBoxes are denoted by $\mathit{DL\text{-}Lite}_{krom}^{\mathcal{R},\mathcal{N}}$, $\mathit{DL\text{-}Lite}_{horn}^{\mathcal{R},\mathcal{N}}$ and $\mathit{DL\text{-}Lite}_{core}^{\mathcal{R},\mathcal{N}}$, respectively.

For $\alpha \in \{\mathit{core}, \mathit{krom}, \mathit{horn}, \mathit{bool}\}$, we denote by $\mathit{DL\text{-}Lite}_{\alpha}^{\mathcal{R},\mathcal{F}}$ the fragment of $\mathit{DL\text{-}Lite}_{\alpha}^{\mathcal{R},\mathcal{N}}$ in which, out of all number restrictions $\geq q\, R$, we are allowed to use existential concepts (i.e., $\exists R ::= \geq 1\, R$) and only those $\geq 2\, R$ that occur in concept inclusions of the form $\geq 2\, R \sqsubseteq \bot$ (known as *global functionality constraints*). If no number restrictions $\geq q\, R$ with $q \geq 2$ are allowed, then we obtain the fragments denoted by $\mathit{DL\text{-}Lite}_{\alpha}^{\mathcal{R}}$. And if role inclusions are excluded from the language then the resulting fragments are denoted by $\mathit{DL\text{-}Lite}_{\alpha}^{\mathcal{N}}$ (with arbitrary number restrictions), $\mathit{DL\text{-}Lite}_{\alpha}^{\mathcal{F}}$ (with functionality constraints and existential concepts $\exists R$), and $\mathit{DL\text{-}Lite}_{\alpha}$ (without number restrictions different from $\exists R$).

As shown in [13], the logics $\mathit{DL\text{-}Lite}_{core}^{\mathcal{R},\mathcal{F}}$, $\mathit{DL\text{-}Lite}_{core}^{\mathcal{R},\mathcal{N}}$ and their extensions turn out to be computationally rather costly, with satisfiability being ExpTime-hard for combined complexity and instance checking P-hard or even NP-hard for data complexity. On the other hand, for the purpose of conceptual modeling one may need both concept and role inclusions. A compromise can be found by artificially limiting the interplay between role inclusions and number restrictions in a way similar to the logic $\mathit{DL\text{-}Lite}_{\mathcal{A}}$ [16].

For a TBox $\mathcal{T}$, let $\sqsubseteq^*_{\mathcal{T}}$ be the reflexive and transitive closure of the relation $\big\{(R, R'), (\mathit{inv}(R), \mathit{inv}(R')) \mid R \sqsubseteq R' \in \mathcal{T}\big\}$. Say that R' is a *proper sub-role of* R in $\mathcal{T}$ if $R' \sqsubseteq^*_{\mathcal{T}} R$ and $R' \not\sqsubseteq^*_{\mathcal{T}} R$.

Consider now the language $\mathit{DL\text{-}Lite}_{\mathit{bool}}^{(\mathcal{RN})}$ obtained from $\mathit{DL\text{-}Lite}_{\mathit{bool}}^{\mathcal{R},\mathcal{N}}$ by imposing the following syntactic restriction on its TBoxes $\mathcal{T}$:

(inter) if R has a proper sub-role in $\mathcal{T}$ then $\mathcal{T}$ contains no *negative occurrences*[2] of number restrictions $\geq q\, R$ or $\geq q\, \mathit{inv}(R)$ with $q \geq 2$.

Without spoiling its computational properties, we can allow in this language *positive occurrences* of *qualified number restrictions* $\geq q\, R.C$ in TBoxes $\mathcal{T}$ provided that the following condition is satisfied:

(exists) if $\geq q\, R.C$ occurs in $\mathcal{T}$ then $\mathcal{T}$ does not contain negative occurrences of $\geq q'\, R$ or $\geq q'\, \mathit{inv}(R)$, for $q' \geq 2$.

Moreover, we can also allow in $\mathit{DL\text{-}Lite}_{\mathit{bool}}^{(\mathcal{RN})}$ *role disjointness*, *reflexivity*, *irreflexivity*, *symmetry* and *asymmetry constraints*. The languages $\mathit{DL\text{-}Lite}_{\mathit{horn}}^{(\mathcal{RN})}$, $\mathit{DL\text{-}Lite}_{\mathit{krom}}^{(\mathcal{RN})}$ and $\mathit{DL\text{-}Lite}_{\mathit{core}}^{(\mathcal{RN})}$ are defined as the corresponding fragments of $\mathit{DL\text{-}Lite}_{\mathit{bool}}^{(\mathcal{RN})}$ (we only note that a concept C occurring in some $\geq q\, R.C$ can be any concept allowed on the right-hand side of concept inclusions in the respective language or a conjunction thereof).

We also define the languages $\mathit{DL\text{-}Lite}_{\alpha}^{(\mathcal{RF})}$ as sub-languages of $\mathit{DL\text{-}Lite}_{\alpha}^{(\mathcal{RN})}$ in which only number restrictions of the form $\exists R$, $\exists R.C$ and functionality constraints $\geq 2\, R \sqsubseteq \bot$ are allowed—provided, of course, that they satisfy **(inter)** and **(exists)**; in particular, $\exists R.C$ is not allowed if R is functional.

Finally, if the UNA is not adopted, it is standard to include in the language *equality* and *inequality constraints* of the form $a_i \approx a_j$ and $a_i \not\approx a_j$ (which are supposed to belong to the ABox part of a KB) with their obvious semantics.

We will concentrate on three fundamental reasoning tasks for the logics $\mathcal{L}$ of the resulting family: satisfiability, instance checking and query answering. The *KB satisfiability problem* is to check, given an $\mathcal{L}$-KB $\mathcal{K}$, whether there is a model of $\mathcal{K}$. The *instance checking problem* is to decide, given an object name a, an $\mathcal{L}$-concept C and an $\mathcal{L}$-KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, whether $\mathcal{K} \models C(a)$, that is, $a^{\mathcal{I}} \in C^{\mathcal{I}}$, for every model $\mathcal{I}$ of $\mathcal{K}$. Instance checking and satisfiability are (AC^0-) reducible to the complement of each other.

Finally, we remind the reader that a *positive existential query* q is any first-order formula constructed by means of $\land$, $\lor$ and $\exists y$ starting from atoms of the form $A(t)$ and $P(t_1, t_2)$, where A is a concept name, P a role name, and t, t_1, t_2

[2] An occurrence of a concept on the right-hand (resp., left-hand) side of a concept inclusion is called *negative* if it is in the scope of an odd (resp., even) number of negations $\neg$; otherwise the occurrence is called *positive*.

Languages	Complexity		
	Combined complexity	Data complexity	
	Satisfiability	Instance checking	Query answering
$DL\text{-}Lite_{core}^{[\ \mid \mathcal{R}]}$	NLogSpace	LogSpace [a)]	LogSpace [a)]
$DL\text{-}Lite_{krom}^{[\ \mid \mathcal{R}]}$	NLogSpace [2]	LogSpace [a)]	coNP [18]
$DL\text{-}Lite_{horn}^{[\ \mid \mathcal{R}]}$	P [2]	LogSpace [a)]	LogSpace [a)] $\leq$ [Th.5]
$DL\text{-}Lite_{bool}^{[\ \mid \mathcal{R}]}$	NP [2]	LogSpace [a)] $\leq$ [Th.1]	coNP
$DL\text{-}Lite_{core/horn}^{[\mathcal{F} \mid (\mathcal{RF})]}$	P $\leq$ [Cor.1] $\geq$ [Th.4]	P $\geq$ [Th.4]	P
$DL\text{-}Lite_{krom}^{[\mathcal{F} \mid (\mathcal{RF})]}$	P $\leq$ [Cor.1]	P	coNP
$DL\text{-}Lite_{bool}^{[\mathcal{F} \mid (\mathcal{RF})]}$	NP	P $\leq$ [Cor.1]	coNP
$DL\text{-}Lite_{core/horn}^{[\mathcal{N} \mid (\mathcal{RN})]}$	NP $\geq$ [Th.2]	coNP $\geq$ [Th.2]	coNP
$DL\text{-}Lite_{krom/bool}^{[\mathcal{N} \mid (\mathcal{RN})]}$	NP $\leq$ [Th.3]	coNP	coNP

[a)] in AC^0 for KBs without equality constraints.

Table 1. Tight complexity results for *DL-Lite* logics without the UNA.
$DL\text{-}Lite_{\alpha}^{[\beta_1 \mid \beta_2]}$ means $DL\text{-}Lite_{\alpha}^{\beta_1}$ or $DL\text{-}Lite_{\alpha}^{\beta_2}$
$DL\text{-}Lite_{core/horn}^{\beta}$ means $DL\text{-}Lite_{core}^{\beta}$ or $DL\text{-}Lite_{horn}^{\beta}$ (likewise for $DL\text{-}Lite_{krom/bool}^{\beta}$)

are *terms* taken from the list of variables $y_0, y_1, \dots$ and the list of object names $a_0, a_1, \dots$. The free variables of q are called *distinguished variables*; we write $\mathsf{q}(x_1, \dots, x_n)$ for a query with distinguished variables $x_1, \dots, x_n$. Given a query $\mathsf{q}(\boldsymbol{x})$ with $\boldsymbol{x} = x_1, \dots, x_n$ and an n-tuple $\boldsymbol{a}$ of object names, we write $\mathsf{q}(\boldsymbol{a})$ for the result of replacing every occurrence of x_i in $\mathsf{q}(\boldsymbol{x})$ with the ith member of $\boldsymbol{a}$. Queries containing no distinguished variables will be called *ground*. The *satisfaction relation* $\mathcal{I} \models^{\mathfrak{a}} \mathsf{q}(\boldsymbol{a})$ between an interpretation $\mathcal{I}$ and a query q under an assignment $\mathfrak{a}$ for the variables y_i in $\Delta^{\mathcal{I}}$ is defined in the usual way (e.g., $\mathcal{I} \models^{\mathfrak{a}} \exists y_i\, \varphi$ iff $\mathcal{I} \models^{\mathfrak{b}} \varphi$, for some assignment $\mathfrak{b}$ in $\Delta^{\mathcal{I}}$ that may differ from $\mathfrak{a}$ only on y_i). For a ground query $\mathsf{q}(\boldsymbol{a})$, the relation $\models^{\mathfrak{a}}$ does not depend on $\mathfrak{a}$, and so we write $\mathcal{I} \models \mathsf{q}(\boldsymbol{a})$ instead of $\mathcal{I} \models^{\mathfrak{a}} \mathsf{q}(\boldsymbol{a})$. The answer to such a query is either 'yes' or 'no.'

For a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, we say that a tuple $\boldsymbol{a}$ of object names from $\mathcal{A}$ is a *certain answer* to $\mathsf{q}(\boldsymbol{x})$ w.r.t. $\mathcal{K}$ and write $\mathcal{K} \models \mathsf{q}(\boldsymbol{a})$, if $\mathcal{I} \models \mathsf{q}(\boldsymbol{a})$ whenever $\mathcal{I} \models \mathcal{K}$. The *query answering problem* is to decide, given a tuple $\boldsymbol{a}$, whether $\mathcal{K} \models \mathsf{q}(\boldsymbol{a})$. Note that instance checking is a special case of query answering.

The complexity results obtained in this paper for the three reasoning problems and both combined and data complexity are summarized in Table 1. Some of them will be proved in the remaining part of the paper.

3 Satisfiability: Combined and Data Complexity

As shown in [2], under the UNA, the satisfiability problem for combined complexity is NLogSpace-complete for the logics of the form $DL\text{-}Lite_{core}^{(\mathcal{RN})}$ and

$DL\text{-}Lite^{(\mathcal{RN})}_{krom}$, P-complete for $DL\text{-}Lite^{(\mathcal{RN})}_{horn}$ and NP-complete for $DL\text{-}Lite^{(\mathcal{RN})}_{bool}$. For data complexity, satisfiability (as well as instance checking) is in AC^0 for all of the above logics and their fragments.

Logics of the form $DL\text{-}Lite^{\mathcal{R}}_{\alpha}$, for $\alpha \in \{core, krom, horn, bool\}$, without equality constraints do not feel whether the UNA is adopted or not: for combined complexity, satisfiability is NLogSpace-complete for $DL\text{-}Lite^{\mathcal{R}}_{core}$ and $DL\text{-}Lite^{\mathcal{R}}_{krom}$, P-complete for $DL\text{-}Lite^{\mathcal{R}}_{horn}$ and NP-complete for $DL\text{-}Lite^{\mathcal{R}}_{bool}$ [2].

However, for data complexity, dropping the UNA results in a slightly higher complexity because of the equality constraints. More precisely, we have:

Theorem 1. *Without the UNA and with equality constraints, the satisfiability and instance checking problems for DL-Lite$_\alpha$ and DL-Lite$^{\mathcal{R}}_{\alpha}$, with $\alpha \in \{core, krom, horn, bool\}$, are* LogSpace*-complete for data complexity.*

The proof of this theorem is based on the following reduction:

Lemma 1. *For every KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, one can construct in* LogSpace *in the size of $\mathcal{A}$ a KB $\mathcal{K}' = (\mathcal{T}, \mathcal{A}')$ without equality constraints such that $\mathcal{I} \models \mathcal{K}$ iff $\mathcal{I} \models \mathcal{K}'$, for every interpretation $\mathcal{I}$.*

Proof. Let $G = (V, E)$ be the symmetric graph with

$$V = ob(\mathcal{A}), \qquad E = \big\{(a_i, a_j) \mid a_i \approx a_j \in \mathcal{A} \ \text{ or } \ a_j \approx a_i \in \mathcal{A}\big\},$$

and $[a_i]$ the set of all vertices of G that are reachable from a_i. Define $\mathcal{A}'$ by removing all the equality constraints from $\mathcal{A}$ and replacing every a_i with $a_j \in [a_i]$ with minimal j. Note that this minimal j can be computed in LogSpace: just enumerate the object names a_j w.r.t. the order of their indices j and check whether the current a_j is reachable from a_i in G. It remains to recall that reachability in undirected graphs is SLogSpace-complete and that SLogSpace = LogSpace [17].

In view of the reduction in the proof of Lemma 1 and the fact that AC^0 is a proper subclass of LogSpace, the upper bound in Theorem 1 cannot be lowered to AC^0. However, the AC^0 data complexity can be regained if we refrain from using the equality constraints; see Section 4.

Let us consider now the logics of the form $DL\text{-}Lite^{(\mathcal{RN})}_{\alpha}$ and $DL\text{-}Lite^{(\mathcal{RF})}_{\alpha}$, together with their fragments.

3.1 $DL\text{-}Lite^{(\mathcal{RN})}_{\alpha}$: Arbitrary Number Restrictions

In this section, we show that the interaction between number restrictions and the possibility of identifying objects in the ABox results in a higher complexity. Indeed, it turns out that, for both combined complexity and data complexity, the satisfiability problem for the logics $DL\text{-}Lite^{\mathcal{N}}_{\alpha}$ and $DL\text{-}Lite^{(\mathcal{RN})}_{\alpha}$, $\alpha \in \{core, krom, horn, bool\}$, without the UNA is NP-complete. This is quite different from the case when the UNA is adopted, where satisfiability is in AC^0 for any of the logics $DL\text{-}Lite^{(\mathcal{RN})}_{\alpha}$ for data complexity, and is tractable for the logics $DL\text{-}Lite^{(\mathcal{RN})}_{horn}$, $DL\text{-}Lite^{(\mathcal{RN})}_{krom}$ and $DL\text{-}Lite^{(\mathcal{RN})}_{core}$ under combined complexity [2].

Theorem 2. *Without the UNA, satisfiability of DL-Lite$^{\mathcal{N}}_{core}$ KBs (even without equality and inequality constraints) is* NP*-hard for both combined and data complexity.*

Proof. The proof is by reduction of the following variant of the 3SAT problem—called *monotone one-in-three 3SAT*—which is known to be NP-complete [10]: given a *positive* 3CNF formula

$$\varphi \quad = \quad \bigwedge_{k=1}^{n} \big(a_{k,1} \lor a_{k,2} \lor a_{k,3}\big),$$

where each $a_{k,j}$ is one of the propositional variables $a_1, \dots, a_m$, decide whether there is an assignment for the variables a_j such that *exactly one variable* is true in each of the clauses in φ. To encode this problem in the language of DL-Lite$^{\mathcal{N}}_{core}$, we need object names a_i^k, for $1 \le k \le n$, $1 \le i \le m$, and c_k and t^k, for $1 \le k \le n$, role names S and P, and concept names A_1, A_2, A_3. Let $\mathcal{A}_\varphi$ be the ABox containing the following assertions:

$$\begin{array}{ll}
S(a_i^1, a_i^2), \ \ \dots, \ \ S(a_i^{n-1}, a_i^n), \ \ S(a_i^n, a_i^1), & \text{for } 1 \le i \le m,\\
S(t^1, t^2), \ \ \dots, \ \ S(t^{n-1}, t^n), \ \ S(t^n, t^1), & \\
P(c_k, t^k), \qquad \text{for } 1 \le k \le n, & \\
P(c_k, a_{k,j}^k), \ \ A_j(a_{k,j}^k), \qquad \text{for } 1 \le k \le n, \ \ 1 \le j \le 3, &
\end{array}$$

and let $\mathcal{T}$ be the TBox with the axioms:

$$A_1 \sqsubseteq \neg A_2, \quad A_2 \sqsubseteq \neg A_3, \quad A_3 \sqsubseteq \neg A_1, \qquad \ge 2\, S \ \sqsubseteq \ \bot, \qquad \ge 4\, P \ \sqsubseteq \ \bot.$$

Clearly, $(\mathcal{T}, \mathcal{A}_\varphi)$ is a DL-Lite$^{\mathcal{N}}_{core}$ KB and $\mathcal{T}$ does not depend on φ (so that we cover both combined and data complexity). We claim that the answer to the monotone one-in-three 3SAT problem is positive iff $(\mathcal{T}, \mathcal{A}_\varphi)$ is satisfiable without the UNA.

($\Rightarrow$) Let $\mathfrak{a}$ be an assignment satisfying the requirements of the problem. Take some a_{i_0} with $\mathfrak{a}(a_{i_0}) = \mathbf{t}$ (clearly, such an i_0 exists, for otherwise $\mathfrak{a}(\varphi) = \mathbf{f}$) and construct an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ by taking:

- $\Delta^{\mathcal{I}} = \big\{y_k, z^k \mid 1 \le k \le n\big\} \cup \big\{x_i^k \mid \mathfrak{a}(a_i) = \mathbf{f},\ 1 \le i \le m, 1 \le k \le n\big\}$,
- $c_k^{\mathcal{I}} = y_k$ and $(t^k)^{\mathcal{I}} = z^k$, for $1 \le k \le n$,
- $(a_i^k)^{\mathcal{I}} = \begin{cases} x_i^k, & \text{if } \mathfrak{a}(a_i) = \mathbf{f},\\ z^k, & \text{if } \mathfrak{a}(a_i) = \mathbf{t}, \end{cases}$ for $1 \le i \le m$, $1 \le k \le n$,
- $S^{\mathcal{I}} = \big\{((a_i^1)^{\mathcal{I}}, (a_i^2)^{\mathcal{I}}), \dots, ((a_i^{n-1})^{\mathcal{I}}, (a_i^n)^{\mathcal{I}}), ((a_i^n)^{\mathcal{I}}, (a_i^1)^{\mathcal{I}}) \mid 1 \le i \le m\big\}$,
- $P^{\mathcal{I}} = \big\{(c_k^{\mathcal{I}}, (t^k)^{\mathcal{I}}) \mid 1 \le k \le n\big\} \cup \big\{(c_k^{\mathcal{I}}, (a_{k,j}^k)^{\mathcal{I}}) \mid 1 \le k \le n,\ 1 \le j \le 3\big\}$.

It is readily checked that $\mathcal{I} \models (\mathcal{T}, \mathcal{A}_\varphi)$.

($\Leftarrow$) Suppose $\mathcal{I} \models \mathcal{K}$. Define an assignment $\mathfrak{a}$ by taking $\mathfrak{a}(a_i) = \mathbf{t}$ iff $(a_i^1)^{\mathcal{I}} = (t^1)^{\mathcal{I}}$. Our aim is to show that $\mathfrak{a}(a_{k,j}) = \mathbf{t}$ for exactly one $j \in \{1, 2, 3\}$, for each k, $1 \le k \le n$. We have $P^{\mathcal{I}}(c_k^{\mathcal{I}}, (a_{k,j}^k)^{\mathcal{I}})$ for all $j = 1, 2, 3$. Moreover, $(a_{k,i}^k)^{\mathcal{I}} \neq (a_{k,j}^k)^{\mathcal{I}}$

for $i \neq j$. As $c_k^{\mathcal{I}} \in (\leq 3\, P)^{\mathcal{I}}$ and $P^{\mathcal{I}}(c_k^{\mathcal{I}}, (t^k)^{\mathcal{I}})$, we then must have $(a_{k,j}^k)^{\mathcal{I}} = (t^k)^{\mathcal{I}}$ for some unique $j \in \{1, 2, 3\}$. It follows from functionality of S that, for each $1 \leq k \leq n$, we have $(a_{k,j}^1)^{\mathcal{I}} = (t^1)^{\mathcal{I}}$ for exactly one $j \in \{1, 2, 3\}$. ☐

The next theorem establishes a matching upper bound:

Theorem 3. *Without the UNA, satisfiability of DL-Lite*$_\alpha^{\mathcal{N}}$ *and DL-Lite*$_\alpha^{(\mathcal{RN})}$ *KBs with equality and inequality constraints is* NP*-complete for both combined complexity and data complexity and any* $\alpha \in \{\mathit{core}, \mathit{krom}, \mathit{horn}, \mathit{bool}\}$*.*

Proof. The upper bound can proved using the following non-deterministic algorithm. Given a *DL-Lite*$_{bool}^{(\mathcal{RN})}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, we

- guess an equivalence relation $\sim$ over $ob(\mathcal{A})$;
- select in each equivalence class $a_i/_\sim$ a representative, say a_i, and replace every occurrence of $a_{i'} \in a_i/_\sim$ in $\mathcal{A}$ with a_i;
- fail if the equalities and inequalities are violated in the resulting ABox—i.e., if it contains $a_i \not\approx a_i$ or $a_i \approx a_j$, for $i \neq j$;
- otherwise, remove the equality and inequality constraints from the ABox and denote the result by $\mathcal{A}'$;
- use an NP satisfiability checking algorithm for *DL-Lite*$_{bool}^{\mathcal{N}}$ to decide whether the KB $\mathcal{K}' = (\mathcal{T}, \mathcal{A}')$ is consistent under the UNA.

Clearly, if the algorithm returns 'yes,' then $\mathcal{I}' \models \mathcal{K}'$, for some $\mathcal{I}'$ respecting the UNA, and we can construct a model $\mathcal{I}$ of $\mathcal{K}$ (not necessarily respecting the UNA) by extending $\mathcal{I}'$ with the following interpretation of object names: $a^{\mathcal{I}} = a_i^{\mathcal{I}'}$, whenever a_i is the representative of $a/_\sim$ ($\mathcal{I}$ coincides with $\mathcal{I}'$ on all other symbols). Conversely, if $\mathcal{I} \models \mathcal{K}$ then we take the equivalence relation $\sim$ defined by $a_i \sim a_j$ iff $a_i^{\mathcal{I}} = a_j^{\mathcal{I}}$. Let $\mathcal{I}'$ be constructed from $\mathcal{I}$ by removing the interpretations of all object names that are not representatives of the equivalence classes for $\sim$. It follows that $\mathcal{I}'$ respects the UNA and is a model of $\mathcal{K}'$, so the algorithm returns 'yes.' ☐

3.2 *DL-Lite*$_\alpha^{(\mathcal{RF})}$: Functionality Constraints

Let us consider now *DL-Lite*$_{bool}^{(\mathcal{RF})}$ and its fragments. In the absence of the UNA, the necessity to identify pairs of objects due to functionality constraints also causes an increase in complexity. However, this increase is less dramatic as the procedure of identifying objects is deterministic: without the UNA, the satisfiability problem for the logics of the form *DL-Lite*$_\alpha^{\mathcal{F}}$ and *DL-Lite*$_\alpha^{(\mathcal{RF})}$, $\alpha \in \{\mathit{core}, \mathit{krom}, \mathit{horn}, \mathit{bool}\}$, is P-complete for data complexity (under the UNA, it is in AC^0 [2]). For the combined complexity, satisfiability in *DL-Lite*$_\alpha^{\mathcal{F}}$ and *DL-Lite*$_\alpha^{(\mathcal{RF})}$, $\alpha \in \{\mathit{core}, \mathit{krom}\}$, becomes P-complete rather than NLogSpace-complete as it is under the UNA (and remains the same for $\alpha \in \{\mathit{horn}, \mathit{bool}\}$).

The following lemma shows that for all these logics reasoning without the UNA can be reduced in polynomial time in the size of the ABox to reasoning under the UNA.

Lemma 2. *For every DL-Lite$_{bool}^{(\mathcal{RF})}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ with equality and inequality constraints, one can construct in polynomial time in $|\mathcal{A}|$ a DL-Lite$_{bool}^{(\mathcal{RF})}$ KB $\mathcal{K}' = (\mathcal{T}, \mathcal{A}')$ such that $\mathcal{A}'$ contains neither equalities nor inequalities, and $\mathcal{K}$ is satisfiable without the UNA iff $\mathcal{K}'$ is satisfiable under the UNA.*

Proof. In what follows by *identifying a_j with a_k in $\mathcal{A}$* we mean replacing each occurrence of a_k in $\mathcal{A}$ with a_j. We construct $\mathcal{A}'$ by first identifying a_j with a_k, for each $a_j \approx a_k \in \mathcal{A}$, and removing the equality from $\mathcal{A}$, and then exhaustively applying the following procedure to $\mathcal{A}$:

– if $\geq 2\, R \sqsubseteq \bot \in \mathcal{T}$, $R_1 \sqsubseteq_{\mathcal{T}}^* R$, $R_2 \sqsubseteq_{\mathcal{T}}^* R$, and $R_1(a_i, a_j), R_2(a_i, a_k) \in \mathcal{A}$, for distinct a_j and a_k, then identify a_j with a_k.

If the resulting ABox contains $a_i \not\approx a_i$, for some a_i, then, clearly, $\mathcal{K}$ is not satisfiable, so we add $A(a_i)$ and $\neg A(a_i)$ to the ABox, for some concept name A. Finally, we remove all inequalities from the ABox and denote the result by $\mathcal{A}'$. It should be clear that $\mathcal{A}'$ is computed from $\mathcal{A}$ in polynomial time and that, without the UNA, $\mathcal{K}$ is satisfiable iff $\mathcal{K}'$ is satisfiable. So it suffices to show that $\mathcal{K}'$ is satisfiable without the UNA iff it is satisfiable under the UNA. The implication ($\Leftarrow$) is trivial. To prove ($\Rightarrow$), observe that every model $\mathcal{I}$ for $\mathcal{K}'$ not respecting the UNA can be transformed into a model $\mathcal{I}$ of $\mathcal{K}'$ respecting the UNA by starting from the set of all object names, which are interpreted as distinct domain elements, and applying the unraveling procedure to cure the defects in the interpretation (for details see Lemmas 8.4 and 5.14 in [2]). ☐

The reduction above cannot be done better than in P, as shown by the next theorem.

Theorem 4. *Without the UNA, satisfiability of DL-Lite$_{core}^{\mathcal{F}}$ KBs (even without equality and inequality constraints) is* P*-hard for both combined and data complexity.*

Proof. The proof is by reduction of the entailment problem for Horn-CNF, which is known to be P-complete (see, e.g., [3, Exercise 2.2.4]). Let

$$\varphi \quad = \quad \bigwedge_{k=1}^{n} \big(a_{k,1} \land a_{k,2} \to a_{k,3}\big) \quad \land \quad \bigwedge_{l=1}^{p} a_{l,0}$$

be a Horn-CNF formula, where each $a_{k,j}$ and each $a_{l,0}$ is one of the propositional variables $a_1, \dots, a_m$ and $a_{k,1}$, $a_{k,2}$, $a_{k,3}$ are all distinct, for each k, $1 \leq k \leq n$. To encode the P-complete problem '$\varphi \models a_i$?' in the language of *DL-Lite*$_{core}^{\mathcal{F}}$ we need object names a_i^k, for $1 \leq k \leq n$, $1 \leq i \leq m$, f_k and g_k, for $1 \leq k \leq n$ and t and role names P, Q, S and T. The ABox $\mathcal{A}$ contains the following assertions

$$S(a_i^1, a_i^2), \quad \dots, \quad S(a_i^{n-1}, a_i^n), \quad S(a_i^n, a_i^1), \qquad \text{for } 1 \leq i \leq m,$$
$$P(a_{k,1}^k, f_k), \quad P(a_{k,2}^k, g_k), \quad Q(g_k, a_{k,3}^k), \quad Q(f_k, a_{k,1}^k), \qquad \text{for } 1 \leq k \leq n,$$
$$T(t, a_{l,0}^1), \qquad \text{for } 1 \leq l \leq p,$$

and the TBox $\mathcal{T}$ asserts that all of the roles are functional:

$$\geq 2\,P \sqsubseteq \bot, \qquad \geq 2\,Q \sqsubseteq \bot, \qquad \geq 2\,S \sqsubseteq \bot \qquad \text{and} \qquad \geq 2\,T \sqsubseteq \bot.$$

Clearly, $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is a *DL-Lite*$^{\mathcal{F}}_{core}$ KB and $\mathcal{T}$ does not depend on φ. We claim that $\varphi \models a_j$ iff $\mathcal{K}' = (\mathcal{T}, \mathcal{A} \cup \{\neg T(t, a^1_j)\})$ is not satisfiable without the UNA. It suffices to prove that $\varphi \models a_j$ iff $\mathcal{I} \models T(t, a^1_j)$ in every model $\mathcal{I}$ of $\mathcal{K}$.

($\Rightarrow$) Let $\mathcal{I} \models \mathcal{K}$ and suppose $\varphi \models a_j$. Then we can derive a_j from φ using the following inference rules:

– $\varphi \models a_{l,0}$ for each l, $1 \leq l \leq p$;
– if $\varphi \models a_{k,1}$ and $\varphi \models a_{k,2}$, for some k, $1 \leq k \leq n$, then $\varphi \models a_{k,3}$.

We show the claim by induction on the length of the derivation of a_j from φ. The basis of induction is trivial. So assume that $a_j = a_{k,3}$, $\varphi \models a_{k,1}$, $\varphi \models a_{k,2}$, for some k, $1 \leq k \leq n$, and that $\mathcal{I} \models T(t, a^1_{k,1}) \wedge T(t, a^1_{k,2})$. Since T is functional, we have $(a^1_{k,1})^{\mathcal{I}} = (a^1_{k,2})^{\mathcal{I}}$. Since S is functional, $(a^{k'}_{k,1})^{\mathcal{I}} = (a^{k'}_{k,2})^{\mathcal{I}}$, for all k', $1 \leq k' \leq n$, and in particular, for $k' = k$. Then, since P is functional, $f^{\mathcal{I}}_k = g^{\mathcal{I}}_k$, from which, by functionality of Q, $(a^k_{k,3})^{\mathcal{I}} = (a^k_{k,1})^{\mathcal{I}}$. Finally, since S is functional, $(a^{k'}_{k,3})^{\mathcal{I}} = (a^{k'}_{k,1})^{\mathcal{I}}$, for all k', $1 \leq k' \leq n$, and in particular, for $k' = 1$. Thus, $\mathcal{I} \models T(t, a^1_j)$.

($\Leftarrow$) Suppose that $\varphi \not\models a_j$. Then there is an assignment $\mathfrak{a}$ such that $\mathfrak{a}(\varphi) = \mathbf{t}$ and $\mathfrak{a}(a_j) = \mathbf{f}$. Construct an interpretation $\mathcal{I}$ by taking

– $\Delta^{\mathcal{I}} = \bigl\{x^k_i \mid \mathfrak{a}(a_i) = \mathbf{f},\ 1 \leq k \leq n, 1 \leq i \leq m\bigr\} \cup \bigl\{z^k, u_k, v_k \mid 1 \leq k \leq n\bigr\} \cup \bigl\{w\bigr\}$,
– $(a^k_i)^{\mathcal{I}} = \begin{cases} x^k_i, & \text{if } \mathfrak{a}(a_i) = \mathbf{f}, \\ z^k, & \text{if } \mathfrak{a}(a_i) = \mathbf{t}, \end{cases}$ for $1 \leq k \leq n$ and $1 \leq i \leq m$,
– $t^{\mathcal{I}} = w$, $\quad T^{\mathcal{I}} = \bigl\{(w, z^1)\bigr\}$,
– $S^{\mathcal{I}} = \bigl\{((a^1_i)^{\mathcal{I}}, (a^2_i)^{\mathcal{I}}), \dots, ((a^{n-1}_i)^{\mathcal{I}}, (a^n_i)^{\mathcal{I}}), ((a^n_i)^{\mathcal{I}}, (a^1_i)^{\mathcal{I}}) \mid 1 \leq i \leq m\bigr\}$,
– $f^{\mathcal{I}}_k = u_k$ and $g^{\mathcal{I}}_k = \begin{cases} v_k, & \text{if } \mathfrak{a}(a_{k,2}) = \mathbf{f}, \\ u_k, & \text{if } \mathfrak{a}(a_{k,2}) = \mathbf{t}, \end{cases}$ for $1 \leq k \leq n$,
– $P^{\mathcal{I}} = \bigl\{((a^k_{k,1})^{\mathcal{I}}, f^{\mathcal{I}}_k), ((a^k_{k,2})^{\mathcal{I}}, g^{\mathcal{I}}_k) \mid 1 \leq k \leq n\bigr\}$,
– $Q^{\mathcal{I}} = \bigl\{(g^{\mathcal{I}}_k, (a^k_{k,3})^{\mathcal{I}}), (f^{\mathcal{I}}_k, (a^k_{k,1})^{\mathcal{I}}) \mid 1 \leq k \leq n\bigr\}$.

It is readily checked that $\mathcal{I} \models \mathcal{K}$ and $\mathcal{I} \not\models T(t, a^1_j)$. $\square$

The above result strengthens the NLogSpace lower bound for instance checking in *DL-Lite*$^{\mathcal{F}}_{core}$ given in [5].

Corollary 1. *Without the UNA, the satisfiability problems for DL-Lite*$^{\mathcal{F}}_{\alpha}$ *and DL-Lite*$^{(\mathcal{RF})}_{\alpha}$ *KBs,* $\alpha \in \{core, krom, horn\}$*, with equalities and inequalities are* P*-complete for both combined complexity and data complexity.*

Without the UNA, satisfiability of DL-Lite$^{\mathcal{F}}_{bool}$ *and DL-Lite*$^{(\mathcal{RF})}_{bool}$ *KBs with equalities and inequalities is* NP*-complete for combined complexity and* P*-complete for data complexity.*

Proof. The upper bounds follow from Lemma 2 and the corresponding upper bounds for the UNA case. The NP lower bound for combined complexity is obvious and the P lower bounds follow from Theorem 4. ◻

4 Query Answering: Data Complexity

The P and coNP upper bounds for data complexity of query answering in $DL\text{-}Lite_{horn}^{\mathcal{R},\mathcal{F}}$ and $DL\text{-}Lite_{bool}^{\mathcal{R},\mathcal{N}}$ without the UNA follow immediately from the results for Horn-$\mathcal{SHIQ}$ [12, 9] and $\mathcal{SHIQ}$ [14, 15, 11], respectively. We show now that the maximal *DL-Lite* logic for which query answering remains in AC^0 without the UNA is the logic $DL\text{-}Lite_{horn}^{\mathcal{R}}$ extended with role constraints as specified in the following theorem:

Theorem 5. *Without the UNA, the positive existential query answering problem for $DL\text{-}Lite_{horn}^{\mathcal{R}}$ KBs with disjointness, (a)symmetry, (ir)reflexivity role constraints and inequalities (but without equalities) is in AC^0 for data complexity. It is* LogSpace*-complete for data complexity and KBs with equality constraints.*

Proof. The proof follows the lines of the proof of [2, Theorem 7.1] given for the UNA case and uses the observation that models without the UNA produce no more answers than their unravelled counterparts. More precisely, let $\mathcal{K}' = (\mathcal{T}', \mathcal{A}')$ be a *consistent* $DL\text{-}Lite_{horn}^{\mathcal{R}}$ KB, and $\mathsf{q}(\boldsymbol{x})$ a positive existential query. Then [2, Lemma 5.17] provides a $DL\text{-}Lite_{horn}^{\mathcal{R}}$ KB $\mathcal{K}$ that has no role constraints but may still have inequality constraints. The following lemma shows that one can simply ignore the inequality constraints in $\mathcal{K}'$ and thus reduces the query answering problem without the UNA to query answering under the UNA:

Lemma 3. *For every tuple $\boldsymbol{a}$ of object names in $\mathcal{K}'$, we have $\mathcal{K}' \models \mathsf{q}(\boldsymbol{a})$ (in models without the UNA) iff $\mathcal{I} \models \mathsf{q}(\boldsymbol{a})$ for all models $\mathcal{I}$ of $\mathcal{K}$ respecting the UNA.*

Proof. ($\Rightarrow$) Suppose that $\mathcal{K}' \models \mathsf{q}(\boldsymbol{a})$ and $\mathcal{I}$ is a model of $\mathcal{K}$ respecting the UNA. In view of satisfiability of $\mathcal{K}'$, we have $\mathcal{I} \models \mathcal{K}'$, and thus $\mathcal{I} \models \mathsf{q}(\boldsymbol{a})$.

($\Leftarrow$) Take any $\mathcal{I}'$ with $\mathcal{I}' \models \mathcal{K}'$. We construct an interpretation $\mathcal{J}'$ respecting the UNA as follows. Let $\Delta^{\mathcal{J}'}$ be the disjoint union of $\Delta^{\mathcal{I}'}$ and $ob(\mathcal{A})$. Define a function $h\colon \Delta^{\mathcal{J}'} \to \Delta^{\mathcal{I}'}$ by taking $h(a) = a^{\mathcal{I}'}$, for each $a \in ob(\mathcal{A})$, and $h(w) = w$, for each $w \in \Delta^{\mathcal{I}'}$, and let

$$a^{\mathcal{J}'} = a, \quad A^{\mathcal{J}'} = \{u \mid h(u) \in A^{\mathcal{I}'}\} \quad \text{and} \quad P^{\mathcal{J}'} = \{(u, v) \mid (h(x), h(v)) \in P^{\mathcal{I}'}\},$$

for all object, concept and role names a, A, P. Clearly, $\mathcal{J}'$ respects the UNA and $\mathcal{J}' \models \mathcal{K}'$. It also follows that h is a homomorphism. By [2, Lemma 5.17], there is a model $\mathcal{I}$ of $\mathcal{K}$ with the same domain as $\mathcal{J}'$ that coincides with $\mathcal{J}'$ on all symbols in $\mathcal{K}'$. As $\mathcal{I} \models \mathsf{q}(\boldsymbol{a})$, we must then have $\mathcal{J}' \models \mathsf{q}(\boldsymbol{a})$, and since h is a homomorphism, $\mathcal{I}' \models \mathsf{q}(\boldsymbol{a})$. Therefore, $\mathcal{K}' \models \mathsf{q}(\boldsymbol{a})$ in models without the UNA, as required. ◻

The remaining part of the proof of the theorem is exactly as in [2, Theorem 7.1], as now we may assume that $\mathcal{K}$ is a $DL\text{-}Lite_{horn}^{\mathcal{R}}$ KB containing neither inequality nor role constraints. ◻

References

1. A. Artale, D. Calvanese, R. Kontchakov, and M. Zakharyaschev. DL-Lite in the light of first-order logic. In *Proc. of the 22nd Nat. Conf. on Artificial Intelligence (AAAI 2007)*, pages 361–366, 2007.
2. A. Artale, D. Calvanese, R. Kontchakov, and M. Zakharyaschev. The *DL-Lite* family and relations. Technical Report BBKCS-09-03, School of Computer Science and Information Systems, Birbeck College, London, 2009. Available at `http://www.dcs.bbk.ac.uk/research/techreps/2009/bbkcs-09-03.pdf`.
3. E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem*. Perspectives in Mathematical Logic. Springer, 1997.
4. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, and R. Rosati. Mastro-i: Efficient integration of relational data through DL ontologies. In *Proc. of the 2007 Description Logic Workshop (DL 2007)*, volume 250 of *CEUR Electronic Workshop Proceedings,* `http://ceur-ws.org/`, pages 227–234, 2007.
5. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, R. Rosati, and M. Ruzzi. Data integration through $\mathit{DL\text{-}Lite}_{\mathcal{A}}$ ontologies. In *Revised Selected Papers of the 3rd Int. Workshop on Semantics in Data and Knowledge Bases (SDKB 2008)*, volume 4925 of *Lecture Notes in Computer Science*, pages 26–47. Springer, 2008.
6. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. *DL-Lite*: Tractable description logics for ontologies. In *Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI 2005)*, pages 602–607, 2005.
7. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. In *Proc. of the 10th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2006)*, pages 260–270, 2006.
8. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *J. of Automated Reasoning*, 39(3):385–429, 2007.
9. T. Eiter, G. Gottlob, M. Ortiz, and M. Šimkus. Query answering in the description logic Horn-$\mathcal{SHIQ}$. In *Proc. of the 11th Eur. Conference on Logics in Artificial Intelligence (JELIA 2008)*, pages 166–179, 2008.
10. M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.
11. B. Glimm, I. Horrocks, C. Lutz, and U. Sattler. Conjunctive query answering for the description logic $\mathcal{SHIQ}$. In *Proc. of the 20th Int. Joint Conf. on Artificial Intelligence (IJCAI 2007)*, pages 399–404, 2007.
12. U. Hustadt, B. Motik, and U. Sattler. Data complexity of reasoning in very expressive description logics. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005)*, pages 466–471, 2005.
13. R. Kontchakov and M. Zakharyaschev. DL-Lite and role inclusions. In *Proc. of the 3rd Asian Semantic Web Conf. (ASWC 2008)*, volume 5367 of *Lecture Notes in Computer Science*, pages 16–30. Springer, 2008.
14. M. Ortiz, D. Calvanese, and T. Eiter. Characterizing data complexity for conjunctive query answering in expressive description logics. In *Proc. of the 21st Nat. Conf. on Artificial Intelligence (AAAI 2006)*, pages 275–280, 2006.
15. M. Ortiz, D. Calvanese, and T. Eiter. Data complexity of query answering in expressive description logics via tableaux. *J. of Automated Reasoning*, 41(1):61–98, 2008.

16. A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Linking data to ontologies. *J. on Data Semantics*, X:133–173, 2008.
17. O. Reingold. Undirected connectivity in log-space. *J. of the ACM*, 55(4), 2008.
18. A. Schaerf. On the complexity of the instance checking problem in concept languages with existential quantification. *J. of Intelligent Information Systems*, 2:265–278, 1993.