DL-Lite with Attributes, Datatypes and Sub-Roles (Full Version)

A. Artale and V. Ryzhikov
KRDB Research Centre
Free University of Bozen-Bolzano, Italy
{*lastname*}@inf.unibz.it

R. Kontchakov
Dept. of Comp. Science and Inf. Sys.
Birkbeck, University of London, UK
roman@dcs.bbk.ac.uk

June 18, 2012

Abstract

We extend the tractable *DL-Lite* languages by (*i*) relaxing the restriction on the allowed interaction between cardinality constraints and role inclusions; (*ii*) extending the languages with attributes and datatypes. On the one hand, we push the limits of the use of cardinality constraints over role hierarchies and also show effects of the ABox on allowed cardinality constraints. On the other hand, attributes—a notion borrowed from data models—associate concrete values from datatypes to abstract objects and in this way complement *DL-Lite* roles, which describe relationships between abstract objects. We present complexity results for two most important reasoning problems in *DL-Lite*: combined complexity of knowledge base satisfiability and data complexity of positive existential query answering.

1 Introduction

The *DL-Lite family* of description logics has recently been proposed and investigated by (Calvanese *et al.*, 2005, 2006, 2007) and later extended by (Artale *et al.*, 2007b; Poggi *et al.*, 2008; Artale *et al.*, 2009). The relevance of the *DL-Lite* family is witnessed by the fact that it forms the basis of OWL 2 QL, one of the three profiles of the Web Ontology Language OWL 2 (`www.w3.org/TR/owl2-profiles`). According to 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.

This paper extends the *DL-Lite* languages of (Artale *et al.*, 2009) by (*i*) relaxing the restriction on the interaction between cardinality constraints (or number restrictions, $\mathcal{N}$) and role inclusions (or hierarchies, $\mathcal{H}$); (*ii*) *attributes* ($\mathcal{A}$), i.e., the possibility to associate concrete values from datatypes to abstract objects. These extensions will be formalized in a new family of languages, $\textit{DL-Lite}_{\alpha}^{\mathcal{HNA}}$, with $\alpha \in \{\textit{core}, \textit{krom}, \textit{horn}, \textit{bool}\}$. Original and tight complexity results for both knowledge base satisfiability and query answering will be presented in this paper.

Role inclusions were introduced in *DL-Lite* by (Calvanese *et al.*, 2006). The possibility to combine them with cardinality constraints on roles has been studied by (Artale *et al.*, 2009): the obtained results show the dramatic impact of role inclusions, when combined with cardinality (or even functionality) constraints, on the computational complexity of reasoning. In particular, query answering becomes coNP-complete in data complexity even for the simplest, core, languages and PTIME-complete for the core and Horn languages with functionality constraints only; moreover, KB satisfiability, which is NLOGSPACE-complete in combined complexity for the simplest, core, language when role inclusions and cardinality constraints are used separately, becomes EXPTIME-complete when they both are present and interact. The *DL-Lite* logic $\textit{DL-Lite}_{\mathcal{A}}$, introduced by (Poggi *et al.*, 2008), retains role inclusions and functionality constraints but limits the interaction between them in order to regain nice computational properties. A similar restriction is also used by (Artale *et al.*, 2009) for limiting this kind of interaction and thus enjoying the computational properties of the *DL-Lite* fragments with only role inclusions or only cardinality constraints.

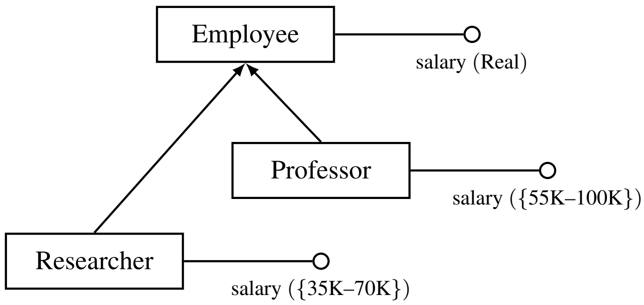

Figure 1: Salary example

The restriction—called **(inter)**—essentially forbids the use of cardinality constraints on roles that are specialized. In this paper we push the limits by relaxing this restriction and allowing specialization of roles even when cardinalities are specified on them. We present two new restrictions, **(inter$_\mathcal{T}$)** and **(inter$_{KB}$)**, whose difference lies in the fact that the latter takes account of the number of R-successors in the ABox while the former does not. We show that **(inter$_{KB}$)** does not lead to an increase in the complexity of knowledge base satisfiability, whereas adopting **(inter$_\mathcal{T}$)** brings the computational complexity up to EXPTIME.

The notion of *attributes*, borrowed from conceptual modelling formalisms, introduces a distinction between (abstract) objects and concrete values (integers, reals, strings, etc.) and, consequently, between concepts (sets of objects) and datatypes (sets of values), and between roles (relating objects to objects) and attributes (relating objects to values). The language *DL-Lite*$_\mathcal{A}$ (Poggi *et al.*, 2008) was introduced with the aim of capturing the notion of attributes in *DL-Lite* in the setting of *ontology-based data access* (OBDA). The datatypes of *DL-Lite*$_\mathcal{A}$ are pairwise disjoint sets of values and a similar choice is made by various DLs encoding conceptual models (Calvanese *et al.*, 1999; Berardi *et al.*, 2005; Artale *et al.*, 2007a). Furthermore, datatypes of *DL-Lite*$_\mathcal{A}$ are used for typing attributes *globally*: e.g., the concept inclusion $\exists salary^- \sqsubseteq Real$ can be used to constrain the range of attribute *salary* to the type *Real*. However, this means that even if associated with different concepts, attributes sharing the same name must have the same range restriction.

We consider a more expressive language for attributes and datatypes in *DL-Lite*. We present two main extensions of the original *DL-Lite*$_\mathcal{A}$: (*i*) datatypes are not necessarily disjoint; instead, Horn clauses define relations between them (in particular, disjointness and subtype relationships); (*ii*) range restrictions for attributes are *local* (rather than global), i.e., concept inclusion axioms of the form $C \sqsubseteq \forall U.T$ can be used to specify that all values of the attribute U of *instances of concept C* belong to datatype T. In this way, we capture a wider range of datatypes (e.g., intervals over the reals) and allow re-use of the very same attribute associated to different concepts, but with different range restrictions. As an example, consider the Entity-Relationship diagram in Fig. 1, which says, in particular, that

- employees' salary is of type *Real*, i.e., $Employee \sqsubseteq \forall salary.Real$;
- researchers' salary is in the range *35K–70K*, which is an interval type, a subset of *Real*, i.e., $Researcher \sqsubseteq \forall salary.\{35K\text{–}70K\}$;
- and professors' salary in the range *55K–100K*, i.e., $Professor \sqsubseteq \forall salary.\{55K\text{–}100K\}$;
- with researchers and professors being employees, i.e., $Researcher \sqsubseteq Employee$ and $Professor \sqsubseteq Employee$.

Local attributes are strictly more expressive than global ones: e.g., the concept inclusion $\top \sqsubseteq \forall salary.Real$ is equivalent to $\exists salary^- \sqsubseteq Real$ mentioned above and implies that *every* value of *salary* is a *Real*, independently of the type of the employee. Using local attributes we can infer concept disjointness from datatype disjointness for the *same* (existentially qualified) attribute. For example, assume that in the scenario of Fig. 1 we add the concept of *ForeignEmployee* as having at-least one *salary* that must be a *String* (to take account of the currency). Then *Employee* and *ForeignEmployee* become disjoint concepts—i.e., *Employee* $\sqcap$ *ForeignEmployee* $\sqsubseteq \bot$ will be implied—because of disjointness of the respective datatypes and restrictions on the salary attribute. We also allow more general datatype inclusions, which, for instance, can express that the intersection of a number of datatypes is empty.

Our work lies between the *DL-Lite*$_{\mathcal{A}}$ proposal and the extensions of DLs with concrete domains (see (Lutz, 2003) for an overview). According to the concrete domain terminology, we consider a path-free extension with unary predicates—predicates coincide with datatypes with a fixed interpretation, as in *DL-Lite*$_{\mathcal{A}}$. Differently from the concrete domain approach, we do not require attributes to be functional; instead, we can specify generic number restrictions over them, similarly to extensions of $\mathcal{EL}$ with datatypes (Baader *et al.*, 2005; Despoina *et al.*, 2011) and the notion of datatype properties in OWL 2 (Pan and Horrocks, 2011; Cuenca Grau *et al.*, 2008). Our approach works as far as datatypes are *safe*, i.e., unbounded and no covering constraints hold between them: query answering becomes coNP-hard in presence of datatypes of specific cardinalities (Franconi *et al.*, 2011; Savković, 2011) or in the presence of a datatype, whose extension is a subset of (is covered by) the union of two other datatypes (cf. Theorem 5).

We provide tight complexity results showing that addition of *local* and *safe* range restrictions on attributes to the Bool, Horn and core languages does not change the complexity of knowledge base satisfiability. On the other hand, surprisingly, for the Krom language complexity increases from NLogSpace to NP. These results reflect the intuition that universal restrictions on attributes—as studied in this paper—cannot introduce cyclic dependencies between concepts (datatypes); on the other hand, unrestricted use of universal restrictions ($\forall R.C$) together with sub-roles, by which qualified existential restrictions ($\exists R.C$) can be encoded, results in ExpTime-completeness (Calvanese *et al.*, 2007).

We complete our complexity results by showing that *positive existential query* answering (and so, conjunctive query answering) over core and Horn knowledge bases with attributes, local range restrictions and safe datatypes is still FO-rewritable and so, is in AC^0 in data complexity.

The paper is organized as follows. Section 2 presents *DL-Lite* and its fragments. In Section 3 we investigate the complexity of deciding knowledge base satisfiability when relaxing the restriction on the interaction between cardinality constraints and role inclusions. In Section 4 we consider the languages with attributes and datatypes and study combined complexity of knowledge base satisfiability and data complexity of answering positive existential queries. Section 5 concludes this paper.

2 The Description Logic *DL-Lite*$_{bool}^{\mathcal{HNA}}$

The language of *DL-Lite*$_{bool}^{\mathcal{HNA}}$ contains *object names* $a_0, a_1, \dots$, *value names* $v_0, v_1, \dots$, *concept names* $A_0, A_1, \dots$, *role names* $P_0, P_1, \dots$, *attribute names* $U_0, U_1, \dots$, and datatype names $T_0, T_1, \dots$. Complex *roles* R, *datatypes* T and *concepts* C are defined as follows:

$$\begin{array}{rcl} R & := & P_i \mid P_i^-, \\ T & := & \bot_{\mathcal{D}} \mid T_i, \\ B & := & \top \mid \bot \mid A_i \mid \geq q\, R \mid \geq q\, U_i, \\ C & := & B \mid \neg C \mid C_1 \sqcap C_2, \end{array}$$

where q is a positive integer. Concepts of the form B are called *basic concepts*. A *DL-Lite*$_{bool}^{\mathcal{HNA}}$ *TBox*, $\mathcal{T}$, is a finite set of *concept*, *role* and *attribute inclusions* of the form:

$$C_1 \sqsubseteq C_2 \text{ and } C \sqsubseteq \forall U.T, \qquad R_1 \sqsubseteq R_2, \qquad U_1 \sqsubseteq U_2,$$

and an *ABox*, $\mathcal{A}$, is a finite set of assertions of the form:

$$A_k(a_i), \quad \neg A_k(a_i), \quad P_k(a_i, a_j), \quad \neg P_k(a_i, a_j), \quad U_k(a_i, v_j).$$

We standardly abbreviate $\geq 1\, R$ and $\geq 1\, U$ by $\exists R$ and $\exists U$, respectively. Taken together, a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$ constitute the *knowledge base* (KB) $\mathcal{K} = (\mathcal{T}, \mathcal{A})$.

Semantics. As usual in description logic, an *interpretation*, $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, consists of a nonempty *domain* $\Delta^{\mathcal{I}}$ and an interpretation function $\cdot^{\mathcal{I}}$. The interpretation domain $\Delta^{\mathcal{I}}$ is the union of two nonempty disjoint sets: the *domain of objects* $\Delta_O^{\mathcal{I}}$ and the *domain of values* $\Delta_V^{\mathcal{I}}$. We assume that all interpretations agree on the semantics of each datatype T_i and of each value v_j. In particular, $\bot_{\mathcal{D}}^{\mathcal{I}} = \emptyset$ and $T_i^{\mathcal{I}} = \mathit{val}(T_i) \subseteq \Delta_V^{\mathcal{I}}$ is the set of values of the datatype T_i (which does not depend on the particular interpretation), and each v_j is interpreted by one specific value, denoted $\mathit{val}(v_j)$, i.e., $v_j^{\mathcal{I}} = \mathit{val}(v_j) \in \Delta_V^{\mathcal{I}}$ (which, again, does not depend on $\mathcal{I}$). Note that the datatypes do not have to be mutually disjoint—instead, we assume that datatype constraints are defined by Horn clauses (expressing, in particular, disjointness and subtype relationships)—we will clarify the assumptions in Section 4.

The interpretation function $\cdot^{\mathcal{I}}$ assigns an element $a_i^{\mathcal{I}} \in \Delta_O^{\mathcal{I}}$ to each object name a_i, a subset $A_k^{\mathcal{I}} \subseteq \Delta_O^{\mathcal{I}}$ of the domain of objects to each concept name A_k, a binary relation $P_k^{\mathcal{I}} \subseteq \Delta_O^{\mathcal{I}} \times \Delta_O^{\mathcal{I}}$ over the domain of objects to each role name P_k, and a binary relation $U_k^{\mathcal{I}} \subseteq \Delta_O^{\mathcal{I}} \times \Delta_V^{\mathcal{I}}$ to each attribute name U_k. We adopt the *unique name assumption* (UNA): $a_i^{\mathcal{I}} \neq a_j^{\mathcal{I}}$, for all $i \neq j$. It is known (Artale *et al.*, 2009) that *not* adopting the UNA in *DL-Lite* languages with cardinality constraints leads to a significant increase in the complexity of reasoning: KB satisfiability goes from NLOGSPACE to PTIME-hard with functionality constraints and even to NP-hard with arbitrary cardinality constraints; query answering loses the AC^0 data complexity. The role and concept constructs are interpreted in $\mathcal{I}$ in the standard way:

$$\begin{aligned}
(P_k^-)^{\mathcal{I}} &= \{(w', w) \in \Delta_O^{\mathcal{I}} \times \Delta_O^{\mathcal{I}} \mid (w, w') \in P_k^{\mathcal{I}}\},\\
\top^{\mathcal{I}} &= \Delta_O^{\mathcal{I}}, \qquad \bot^{\mathcal{I}} = \emptyset,\\
(C_1 \sqcap C_2)^{\mathcal{I}} &= C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}, \quad (\neg C)^{\mathcal{I}} = \Delta_O^{\mathcal{I}} \setminus C^{\mathcal{I}},\\
(\geq q\, R)^{\mathcal{I}} &= \big\{w \in \Delta_O^{\mathcal{I}} \mid \sharp\{w' \mid (w, w') \in R^{\mathcal{I}}\} \geq q\big\},\\
(\geq q\, U)^{\mathcal{I}} &= \big\{w \in \Delta_O^{\mathcal{I}} \mid \sharp\{v \mid (w, v) \in U^{\mathcal{I}}\} \geq q\big\},\\
(\forall U.\, T)^{\mathcal{I}} &= \big\{w \in \Delta_O^{\mathcal{I}} \mid v \in T^{\mathcal{I}}, \text{ for all } v \text{ with } (w, v) \in U^{\mathcal{I}}\big\},
\end{aligned}$$

where $\sharp X$ is the cardinality of X. The *satisfaction relation* $\models$ is also standard:

$$\begin{aligned}
&\mathcal{I} \models C_1 \sqsubseteq C_2 \text{ iff } C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}, &&\mathcal{I} \models R_1 \sqsubseteq R_2 \text{ iff } R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}},\\
&\mathcal{I} \models U_1 \sqsubseteq U_2 \text{ iff } U_1^{\mathcal{I}} \subseteq U_2^{\mathcal{I}}, &&\\
&\mathcal{I} \models A_k(a_i) \text{ iff } a_i^{\mathcal{I}} \in A_k^{\mathcal{I}}, &&\mathcal{I} \models \neg A_k(a_i) \text{ iff } a_i^{\mathcal{I}} \notin A_k^{\mathcal{I}},\\
&\mathcal{I} \models P_k(a_i, a_j) \text{ iff } (a_i^{\mathcal{I}}, a_j^{\mathcal{I}}) \in P_k^{\mathcal{I}}, &&\mathcal{I} \models \neg P_k(a_i, a_j) \text{ iff } (a_i^{\mathcal{I}}, a_j^{\mathcal{I}}) \notin P_k^{\mathcal{I}},\\
&\mathcal{I} \models U_k(a_i, v_j) \text{ iff } (a_i^{\mathcal{I}}, v_j^{\mathcal{I}}) \in U_k^{\mathcal{I}}. &&
\end{aligned}$$

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

A *positive existential query* $\mathsf{q}(x_1, \dots, x_n)$ is a first-order formula $\varphi(x_1, \dots, x_n)$ constructed by means of conjunction, disjunction and existential quantification starting from atoms of the from $A_k(t_1)$, $T_k(t_1)$, $P_k(t_1, t_2)$ and $U_k(t_1, t_2)$, where A_k is a concept name, T_k a datatype name, P_k a role name, U_k an attribute name, and t_1, t_2 are *terms* taken from the list of variables $y_0, y_1, \dots$, object names $a_0, a_1, \dots$ and value names $v_0, v_1, \dots$; object and value names are called *constants*. We write $\mathsf{q}(\vec{x})$ for a query with free variables $\vec{x} = x_1, \dots, x_n$ and $\mathsf{q}(\vec{a})$ for the result of replacing every occurrence of x_i in $\varphi(\vec{x})$ with the ith component a_i of a vector of constants $\vec{a} = a_1, \dots, a_n$. A *conjunctive query* is a positive existential query that contains no disjunction.

For a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, we say that a tuple $\vec{a}$ of constants from $\mathcal{A}$ is a *certain answer* to $\mathsf{q}(\vec{x})$ with respect to $\mathcal{K}$, and write $\mathcal{K} \models \mathsf{q}(\vec{a})$, if $\mathcal{I} \models \mathsf{q}(\vec{a})$ whenever $\mathcal{I} \models \mathcal{K}$. The *query answering problem* is: given a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, a query $\mathsf{q}(\vec{x})$ and a tuple $\vec{a}$ of constants from $\mathcal{A}$, decide whether $\mathcal{K} \models \mathsf{q}(\vec{a})$.

Fragments of *DL-Lite*$_{bool}^{\mathcal{HNA}}$. We consider syntactical restrictions on the language of *DL-Lite*$_{bool}^{\mathcal{HNA}}$ along two axes: (*i*) the Boolean operators ($_{bool}$) on concepts and (*ii*) the attributes ($\mathcal{A}$). Similarly to classical logic, we adopt the following definitions. A TBox $\mathcal{T}$ will be called a *Krom TBox*—from the Krom fragment of first-order logic— if only negation is allowed in the construction of its complex concepts, i.e., if

$$C \quad ::= \quad B \quad \mid \quad \neg B \qquad \text{(Krom)}$$

language	combined complexity of KB satisfiability				data compl.
	(inter)*	(inter$_\mathcal{T}$)	(inter$_{KB}$)	no restrict.	QA
DL-Lite$_{core}^{\mathcal{HN}}$	NLogSpace*	$\geq$NP [Th.1]	NLogSpace [Th.3]	ExpTime*	in AC^{0*}
DL-Lite$_{horn}^{\mathcal{HN}}$	PTime*	ExpTime [Th.1]	PTime [Th.3]		in AC^{0*}
DL-Lite$_{krom}^{\mathcal{HN}}$	NLogSpace*	$\geq$NP [Th.1]	NLogSpace [Th.3]		coNP*
DL-Lite$_{bool}^{\mathcal{HN}}$	NP*	ExpTime [Th.1]	NP [Th.3]		coNP*
DL-Lite$_{core}^{\mathcal{HNA}}$	NLogSpace [Th.7]	$\geq$NP [Th.1]	NLogSpace [Th.7]	ExpTime	in AC^0 [Th.11]
DL-Lite$_{horn}^{\mathcal{HNA}}$	PTime [Th.7]	ExpTime [Th.1]	PTime [Th.7]		in AC^0 [Th.11]
DL-Lite$_{krom}^{\mathcal{HNA}}$	NP [Th.9]	$\geq$NP [Th.1]	NP [Th.9]		coNP
DL-Lite$_{bool}^{\mathcal{HNA}}$	NP [Th.7]	ExpTime [Th.1]	NP [Th.7]		coNP

Table 1: Complexity of *DL-Lite* logics; * = (Artale *et al.*, 2009).

(here and below the B are basic concepts). A TBox $\mathcal{T}$ will be called a *Horn TBox* if its complex concepts are constructed by using only intersection:

$$C \quad ::= \quad B_1 \sqcap \cdots \sqcap B_k. \qquad \text{(Horn)}$$

Finally, we call $\mathcal{T}$ a *core TBox* if its concept inclusions are of the form:

$$B_1 \sqsubseteq B_2, \quad B_1 \sqsubseteq \forall U.T, \quad B_1 \sqcap B_2 \sqsubseteq \bot. \qquad \text{(core)}$$

Note that all the above fragments allow for positive occurrences of $\forall U.T$ on the right-hand side of concept inclusions. 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. In this paper, in addition to the full language of *DL-Lite*$_{bool}^{\mathcal{HNA}}$, we study the following logics:

***DL-Lite*$_{krom}^{\mathcal{HNA}}$, *DL-Lite*$_{horn}^{\mathcal{HNA}}$, *DL-Lite*$_{core}^{\mathcal{HNA}}$** are the fragments of *DL-Lite*$_{bool}^{\mathcal{HNA}}$ with Krom, Horn, and core TBoxes, respectively;

***DL-Lite*$_{\alpha}^{\mathcal{HN}}$**, for $\alpha \in \{core, krom, horn, bool\}$, is the fragment of *DL-Lite*$_{\alpha}^{\mathcal{HNA}}$ without attributes and datatypes.

Table 1 summarizes the obtained complexity results (with numbers q coded in binary) for KB satisfiability (combined complexity) and positive existential query answering (data complexity).

3 Complexity of Reasoning in *DL-Lite*$_{\alpha}^{\mathcal{HN}}$

As shown by (Artale *et al.*, 2009), reasoning in *DL-Lite*$_{\alpha}^{\mathcal{HN}}$ is already rather costly (ExpTime-complete) due to the interaction between role inclusions and cardinality constraints. However, both of these constructs turn out to be useful for the purposes of conceptual modelling. By limiting their interplay one can get languages with better computational properties. In this section we formulate and study two syntactic restrictions that are weaker than the ones known in the literature (Poggi *et al.*, 2008; Artale *et al.*, 2009).

In the following, we denote by $role^{\pm}(\mathcal{K})$ the set of roles P_k in $\mathcal{K}$ with their inverses P_k^-. For a role R, let $inv(R) = P_k^-$ if $R = P_k$ and $inv(R) = P_k$ if $R = P_k^-$. Given a TBox $\mathcal{T}$ we denote by $\sqsubseteq_{\mathcal{T}}^*$ the reflexive and transitive closure of the relation $\{(R, R'), (inv(R), inv(R')) \mid R \sqsubseteq R' \in \mathcal{T}\}$. We say that R' is a *proper sub-role* of R in $\mathcal{T}$ if $R' \sqsubseteq_{\mathcal{T}}^* R$ and $R \not\sqsubseteq_{\mathcal{T}}^* R'$. A proper sub-role R' of R is a *direct sub-role* of R if there is no other proper sub-role R'' of R such that R' is a proper sub-role of R''; $dsub_{\mathcal{T}}(R)$ denotes the set of direct sub-roles of R in $\mathcal{T}$. An occurrence of a concept on the right-hand (left-hand) side of a concept inclusion is called *negative* if it is in the scope of an odd (even) number of negations $\neg$; otherwise it is called *positive*.

3.1 Counting Successors in Hierarchies

The languages *DL-Lite*$_{\alpha}^{(\mathcal{HN})}$ of (Artale *et al.*, 2009) are the result of imposing the following syntactic restriction on *DL-Lite*$_{\alpha}^{\mathcal{HN}}$ TBoxes $\mathcal{T}$:

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

To formulate our subtler restrictions, we need the following parameters, for a TBox $\mathcal{T}$ and a role $R \in \mathit{role}^{\pm}(\mathcal{T})$:

$$\begin{aligned}
\mathit{ub}(R, \mathcal{T}) &= \min\bigl(\{\infty\} \cup \{q - 1 \mid q \geq 2 \text{ and } \geq q\, R \text{ occurs negatively in } \mathcal{T}\}\bigr),\\
\mathit{lb}(R, \mathcal{T}) &= \max\bigl(\{0\} \cup \{q \mid \geq q\, R \text{ occurs positively in } \mathcal{T}\}\bigr),\\
\mathit{rank}(R, \mathcal{T}) &= \max\bigl(\mathit{lb}(R, \mathcal{T}), \sum_{R' \in dsub_{\mathcal{T}}(R)} \mathit{rank}(R', \mathcal{T})\bigr).
\end{aligned}$$

Consider first the languages obtained from *DL-Lite*$_{\alpha}^{\mathcal{HN}}$ by imposing the following restriction:

(inter$_{\mathcal{T}}$) if $R \in \mathit{role}^{\pm}(\mathcal{T})$ has a proper sub-role in $\mathcal{T}$ then

$$\mathit{ub}(R, \mathcal{T}) \geq \mathit{rank}(R, \mathcal{T}).$$

It turns out, however, that these languages are too expressive to keep the same complexity of the satisfiability problem as their basic counterparts:

THEOREM 1. *Under* **(inter$_{\mathcal{T}}$)***, KB satisfiability is* NP*-hard for DL-Lite*$_{core}^{\mathcal{HN}}$ *and DL-Lite*$_{krom}^{\mathcal{HN}}$ *and* EXPTIME*-complete for DL-Lite*$_{horn}^{\mathcal{HN}}$ *and DL-Lite*$_{bool}^{\mathcal{HN}}$.

Proof. To prove NP-hardness, we show that graph 3-colorability can be reduced to *DL-Lite*$_{core}^{\mathcal{HN}}$ KB satisfiability. Let $G = (V, E)$ be a graph with vertices V and edges E and $\{r, g, b\}$ be three colors. Consider the following KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ with a role name S and its sub-roles R_i, for each vertex $v_i \in V$, and object names o, r, g, b and v_i, for each vertex $v_i \in V$:

$$\begin{aligned}
\mathcal{T} =& \{\geq (|V| + 4)\, S \sqsubseteq \bot\} \cup\\
& \{R_i \sqsubseteq S,\ \ B_1 \sqsubseteq \exists R_i,\ \ B_2 \sqcap \exists R_i^- \sqsubseteq \bot \mid v_i \in V\} \cup\\
& \{\exists R_i^- \sqcap \exists R_j^- \sqsubseteq \bot \mid (v_i, v_j) \in E\},\\
\mathcal{A} =& \{B_1(o), S(o, r), S(o, g), S(o, b)\} \cup \{S(o, v_i),\ \ B_2(v_i) \mid v_i \in V\}.
\end{aligned}$$

Clearly, $\mathcal{K}$ enjoys **(inter$_{\mathcal{T}}$)**. It can be shown that $\mathcal{K}$ is satisfiable iff G is 3-colorable. Indeed, for every vertex v_i, the individual v_i is an S-successor of o, which has another three S-successors: r, g and b. On the other hand, for each vertex v_i, o must have an R_i-successor (which is also an S-successor) but the total number of S-successors of o is bounded by $|V| + 3$. Since the v_j cannot be R_i-successors (for any pair i, j), all the R_i-successors of o must be among r, g and b, which by the range disjointness axiom for R_i and R_j (provided that $(v_i, v_j) \in E$) is possible iff the graph is 3-colorable.

EXPTIME-hardness can be proved by reduction of the complement of the state reachability problem for alternating Turing machines (ATMs). We only give an idea of the proof here. Suppose we are given an ATM that, on every input, requires only a polynomial number of cells on the tape. Without loss of generality we may assume that each state has exactly two successor states on each input symbol. Let n be the length of the input and ℓ the number of cells required. Then we need the following 3 sets of roles, for $0 \leq k < 3$,

- S_{kai}, for each symbol $a \in \Sigma$ and position $1 \leq i \leq \ell$, so that $\exists S_{kai}^-$ says 'the symbol a is written at the position i';
- H_{kqi}, for each state $q \in Q$ and head position $1 \leq i \leq \ell$, so that $\exists H_{kqi}^-$ says 'the current state is q and the head is over the position i';

(the three sets are required for the disjointness constraints below). Since each state has two successors, we also need two sub-roles (left and right) of each S_{kai}:

$$LS_{kai} \sqsubseteq S_{kai}, \quad RS_{kai} \sqsubseteq S_{kai}$$

and sub-roles LH_{kqi} and RH_{kqi} for each H_{kqi}. With the help of these pairs of roles we can encode transitions of the form $\delta(a, q) = \{(a_1, q_1, d_1), (a_2, q_2, d_2)\}$ in a natural way:

$$\exists S^-_{kai} \sqcap \exists H^-_{kqi} \sqsubseteq \exists LH_{\lfloor k+1 \rfloor q_1 (i+d_1)} \sqcap \exists LS_{\lfloor k+1 \rfloor a_1 i} \sqcap \\ \exists RH_{\lfloor k+1 \rfloor q_2 (i+d_2)} \sqcap \exists RS_{\lfloor k+1 \rfloor a_2 i},$$

where $\lfloor k \rfloor$ denotes the value of k modulo 3. We also need to say that cells that are not under the current position of the head do not change their symbols: for all $j \neq i$,

$$\exists S^-_{kaj} \sqcap \exists H^-_{kqi} \sqsubseteq \exists LS_{\lfloor k+1 \rfloor aj} \sqcap \exists RS_{\lfloor k+1 \rfloor aj}.$$

But now the main difficulty is to enforce that all the $LS_{\lfloor k+1 \rfloor aj}$- and $LH_{\lfloor k+1 \rfloor q_1 i_1}$-successors coincide (and similarly, their right counterparts). We could introduce a new functional super-role for all of them but then the restriction **(inter$_{\mathcal{T}}$)** would be violated. Instead, we will employ a role T_k and its two subroles L_k and R_k, for each $0 \leq k < 3$, and super-roles $\widehat{LS}_{kai}$, $\widehat{RS}_{kai}$, $\widehat{LH}_{kqi}$ and $\widehat{RH}_{kqi}$. Each of these super-roles contains its title role, L_k and $T^-_{\lfloor k-1 \rfloor}$ as its sub-roles and has not more than 2 successors, e.g.:

$$LS_{kai} \sqsubseteq \widehat{LS}_{kai}, \quad L_k \sqsubseteq \widehat{LS}_{kai}, \quad T^-_{\lfloor k-1 \rfloor} \sqsubseteq \widehat{LS}_{kai}, \quad \geq 3\, \widehat{LS}_{kai} \sqsubseteq \bot.$$

With the help of disjointness constraints of the form $\exists T_{\lfloor k-1 \rfloor} \sqcap \exists T^-_k \sqsubseteq \bot$ and $\exists T_{\lfloor k-1 \rfloor} \sqcap \exists S^-_{kai} \sqsubseteq \bot$ and an ABox, modelling the initial configuration and containing $H_{0q_01}(z, a)$, $S_{0a_11}(z, a), \dots, S_{0a_\ell \ell}(z, a)$ and $T_0(z, a)$, we can ensure that in all models of this TBox each point (but z) has a single $T_{\lfloor k-1 \rfloor}$-predecessor and a single L_k-successor, which is a T_k-successor, and, by the cardinality constraints above, is also the $LS_{\lfloor k+1 \rfloor aj}$- and $LH_{\lfloor k+1 \rfloor q_1 i_1}$-successor for the respective combination of subscripts. It is easily seen that the TBox enjoys **(inter$_{\mathcal{T}}$)** and encodes the tree of computations of the ATM. In a similar way one can encode the condition that a certain state is never reached. □

Both the NP- and EXPTIME-hardness proofs use the fact that the restriction **(inter$_{\mathcal{T}}$)** does not impose any bounds on the number of R-successors in the ABox. And the EXPTIME-hardness proof also reveals that if we are to maintain the low complexity of reasoning, we have to take into account not only the number of R-successors in the ABox, but also the number of R^--predecessors (i.e., R-successors) that come to the unnamed individuals outside the ABox. In the next section, this intuition will drive our next attempt to weaken the restrictions on the interaction of role inclusions and cardinality constraints.

3.2 Taking the ABox into Account

In this section, we formulate our second restriction, **(inter$_{KB}$)**, and show that the complexity of KB satisfiability remains low under it. We need the following additional parameters, for an ABox $\mathcal{A}$, a TBox $\mathcal{T}$ and $R \in \mathit{role}^{\pm}(\mathcal{T})$:

$$\mathit{rank}(R, \mathcal{A}) = \max\bigl(\{0\} \cup \{n \mid R_i(a, a_i) \in \mathcal{A}, R_i \sqsubseteq^*_{\mathcal{T}} R, \text{ for distinct } a_1, \dots, a_n\}\bigr),$$

$$\mathit{pred}(R, \mathcal{T}) = \begin{cases} 1, & \text{if } \mathit{lb}(R', \mathcal{T}) \geq 1, \text{ for some } R' \sqsubseteq^*_{\mathcal{T}} R^-, \\ 0, & \text{otherwise.} \end{cases}$$

Then our second restriction on role inclusions and cardinality constraints is as follows:

(inter$_{KB}$) if $R \in \mathit{role}^{\pm}(\mathcal{T})$ has a proper sub-role in $\mathcal{T}$ then

$$\mathit{ub}(R, \mathcal{T}) \geq \mathit{rank}(R, \mathcal{T}) + \max\bigl(\mathit{pred}(R, \mathcal{T}), \mathit{rank}(R, \mathcal{A})\bigr).$$

Both **(inter$_{\mathcal{T}}$)** and **(inter$_{KB}$)** are weaker than **(inter)** and, for example, allow for the specialization of functional roles: $\mathcal{T} = \{\geq 2\, R \sqsubseteq \bot,\ R_1 \sqsubseteq R_2,\ R_2 \sqsubseteq R\}$ and $\mathcal{A} = \{R(a, b), R_1(a_1, b_1), R_2(a_2, b_2)\}$ do not satisfy **(inter)**, but do satisfy both **(inter$_{\mathcal{T}}$)** and **(inter$_{KB}$)**. The above restrictions will also be applied to sub-attributes in the languages *DL-Lite*$^{\mathcal{HNA}}_{\alpha}$.

To show that **(inter$_{KB}$)** matches the complexity of KB satisfiability of the basic languages, we adapt the proof presented in (Artale *et al.*, 2009), where a *DL-Lite*$^{\mathcal{HN}}_{bool}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is encoded into a first-order sentence $\mathcal{K}^{\ddagger_e}$ with one variable. Every $a_i \in ob(\mathcal{A})$ is associated with the individual constant a_i, and every concept name A_i with the unary predicate $A_i(x)$. For each concept $\geq q\, R$ in $\mathcal{K}$ we introduce a fresh unary predicate $E_q R(x)$. We also introduce the set

$$dr(\mathcal{K}) = \{dp_k, dp_k^- \mid P_k \text{ is a role name in } \mathcal{K}\}$$

of individual constants, as representatives of the objects in the domain (dp_k) and the range (dp_k^-) of each role P_k, respectively. The encoding C^* of a concept C is defined inductively:

$$\begin{aligned} \bot^* &= \bot, & (A_i)^* &= A_i(x), \\ \top^* &= \top, & (\neg C)^* &= \neg C^*(x), \\ (\geq q\, R)^* &= E_q R(x), & (C_1 \sqcap C_2)^* &= C_1^*(x) \land C_2^*(x). \end{aligned}$$

The following sentence encodes the knowledge base $\mathcal{K}$:

$$\mathcal{K}^{\ddagger_e} = \forall x \Big[\mathcal{T}^*(x) \land \mathcal{T}^{\mathcal{R}}(x) \land \bigwedge_{R \in role^{\pm}(\mathcal{K})} \big(\epsilon_R(x) \land \delta_R(x)\big) \Big] \land \mathcal{A}^{\ddagger_e},$$

where

$$\begin{aligned} \mathcal{T}^*(x) &= \bigwedge_{C_1 \sqsubseteq C_2 \in \mathcal{T}} \big(C_1^*(x) \to C_2^*(x)\big), \\ \delta_R(x) &= \bigwedge_{q, q' \in Q^R_{\mathcal{T}},\ q' > q} \big(E_{q'} R(x) \to E_q R(x)\big), \\ \mathcal{T}^{\mathcal{R}}(x) &= \bigwedge_{R \sqsubseteq^*_{\mathcal{T}} R'} \bigwedge_{q \in Q^R_{\mathcal{T}}} \big(E_q R(x) \to E_q R'(x)\big), \end{aligned}$$

and $Q^R_{\mathcal{T}}$ contains 1, all q such that $\geq q\, R$ occurs in $\mathcal{T}$ and all $Q^{R'}_{\mathcal{T}}$, for $R' \sqsubseteq^*_{\mathcal{T}} R$. Sentence $\mathcal{A}^{\ddagger_e}$ encodes the ABox $\mathcal{A}$:

$$\mathcal{A}^{\ddagger_e} = \bigwedge_{A_k(a_i) \in \mathcal{A}} A_k(a_i) \;\land \bigwedge_{\neg A_k(a_i) \in \mathcal{A}} \neg A_k(a_i) \;\land \bigwedge_{R(a_i, a_j) \in \mathcal{A}} E_{q^{e}_{R,a_i}} R(a_i) \;\land \bigwedge_{\substack{\neg P_k(a_i, a_j) \in \mathcal{A} \\ R(a_i, a_j) \in \mathcal{A},\ R \sqsubseteq^*_{\mathcal{T}} P_k}} \bot,$$

where $q^{e}_{R,a}$ is the maximum number in $Q^R_{\mathcal{T}}$ such that there are $q^{e}_{R,a}$ many distinct a_i with[1] $R_i(a, a_i) \in \mathcal{A}$ and $R_i \sqsubseteq^*_{\mathcal{T}} R$. For each $R \in role^{\pm}(\mathcal{K})$, we also need a formula expressing the fact that the range of R is not empty whenever its domain is nonempty:

$$\epsilon_R(x) \quad = \quad E_1 R(x) \to inv(E_1 R(dr)),$$

with $inv(E_1 R(dr))$ denoting $E_1 P_k^-(dp_k^-)$ if $R = P_k$ and $E_1 P_k(dp_k)$ if $R = P_k^-$.

Lemma 2. *A DL-Lite*$^{\mathcal{HN}}_{bool}$ *KB $\mathcal{K}$ under* **(inter$_{KB}$)** *is satisfiable iff the one-variable sentence $\mathcal{K}^{\ddagger_e}$ is satisfiable.*

Proof. The only challenging direction is ($\Leftarrow$). To prove it, we adapt the proofs of Theorem 5.2 and Lemma 5.14 of (Artale *et al.*, 2009). The idea of the proof is to construct a *DL-Lite*$^{\mathcal{HN}}_{bool}$ model $\mathcal{I}$ of $\mathcal{K}$ from the minimal Herbrand model $\mathfrak{M}$ of $\mathcal{K}^{\ddagger_e}$ with domain $D = ob(\mathcal{A}) \cup dr(\mathcal{K})$. The interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is defined inductively: $\Delta^{\mathcal{I}} = \bigcup_{m=0}^{\infty} W_m$, such that W_0 is the set $ob(\mathcal{A})$, and each set W_{m+1}, $m \geq 0$, is constructed by adding to W_m fresh *copies* of elements of $dr(\mathcal{K})$. We write $cp(w)$ for the element $d \in D$

[1] We slightly abuse notation and write $R(a_i, a_j) \in \mathcal{A}$ instead of $P_k(a_i, a_j) \in \mathcal{A}$ if $R = P_k$ and $P_k(a_j, a_i) \in \mathcal{A}$ if $R = P_k^-$.

of which w is a copy, with $cp(a) = a$ for $a \in ob(\mathcal{A}) = W_0$. We define $a_i^{\mathcal{I}} = a_i^{\mathfrak{M}} = a_i$, for all individuals $a_i \in ob(\mathcal{A})$, and, for all concept names A_k,

$$A_k^{\mathcal{I}} = \{w \in \Delta^{\mathcal{I}} \mid \mathfrak{M} \models A_k^*[cp(w)]\},$$

The interpretation of each role P_k, is defined inductively as $P_k^{\mathcal{I}} = \bigcup_{m=0}^{\infty} P_k^m$, where $P_k^m \subseteq W_m \times W_m$, along with the construction of $\Delta^{\mathcal{I}}$. The initial interpretation of P_k is

$$P_k^0 = \{(a_i^{\mathfrak{M}}, a_j^{\mathfrak{M}}) \in W_0 \times W_0 \mid R(a_i, a_j) \in \mathcal{A} \text{ and } R \sqsubseteq_{\mathcal{T}}^* P_k\}.$$

The *required R-rank* $r(R, d)$ of $d \in D$ is defined as:

$$r(R, d) = \max\left(\{0\} \cup \{q \in Q_{\mathcal{T}}^{R+} \mid \mathfrak{M} \models E_q R[d]\}\right),$$

where $Q_{\mathcal{T}}^{R+}$ contains all q such that $\geq q\, R$ occurs positively in $\mathcal{T}$. Note that:

$$r(R, d) \leq lb(R, \mathcal{T}). \tag{1}$$

The *actual R-rank* $r_m(R, w)$ of a point $w \in \Delta^{\mathcal{I}}$ at step m is defined as follows:

$$r_m(R, w) = \sharp\{w' \mid (w, w') \in P_k^m \cup P_j^{m+1}, P_j \in dsub_{\mathcal{T}}(P_k)\},$$

if $R = P_k$; replace (w, w') by (w', w) if $R = P_k^-$. Assume that W_m and P_k^m, $m \geq 0$, have been already defined. Let $W_{m+1} \setminus W_m = \emptyset$. If the actual rank of some points is smaller than the required rank, then, we cure these defects by adding R-successors for them. For each role name P_k, we consider two sets of defects in P_k^m:

$$\Lambda_k^m = \{w \in W_m \setminus W_{m-1} \mid r_m(P_k, w) < r(P_k, cp(w))\},$$
$$\Lambda_k^{m-} = \{w \in W_m \setminus W_{m-1} \mid r_m(P_k^-, w) < r(P_k^-, cp(w))\}.$$

In each equivalence class $[R] = \{S \mid S \sqsubseteq_{\mathcal{T}}^* R,\ R \sqsubseteq_{\mathcal{T}}^* S\}$ we select a single role, a *representative*. Let $G = (Rep_{\mathcal{T}}, E)$ be a directed graph such that $Rep_{\mathcal{T}}$ is the set of representatives and $(R, R') \in E$ iff R is a proper sub-role of R'. We use an ascending total order induced on G when choosing an element $[P_k]$ in $Rep_{\mathcal{T}}$, and extend in that way W_m and P_k^m to W_{m+1} and P_k^{m+1}, respectively.

(Λ_k^m) Let $w \in \Lambda_k^m, q = r(P_k, d) - r_m(P_k, w)$, $d = cp(w)$. There is $q' \geq q > 0$ with $\mathfrak{M} \models E_{q'} P_k[d]$ and so, $\mathfrak{M} \models E_1 P_k[d]$ and $\mathfrak{M} \models E_1 P_k^-[dp_k^-]$. We take q *fresh* copies $w_1', \dots, w_q'$ of dp_k^-, add them to W_{m+1} and for each $1 \leq i \leq q$, set $cp(w_i') = dp_k^-$, add the pairs (w, w_i') to each P_j^{m+1} with $P_k \sqsubseteq_{\mathcal{T}}^* P_j$ and the pairs (w_i', w) to each P_j^{m+1} with $P_k^- \sqsubseteq_{\mathcal{T}}^* P_j$;

(Λ_k^{m-}) This rule is the mirror image of (Λ_k^m): P_k and dp_k^- are replaced with P_k^- and dp_k, respectively.

We now show that $\mathcal{I} \models \varphi$ for each $\varphi \in \mathcal{T} \cup \mathcal{A}$. From the construction of $R^{\mathcal{I}}$, it immediately follows that the interpretation of roles respects role inclusions, i.e., $R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}}$ whenever $R_1 \sqsubseteq R_2 \in \mathcal{T}$. For $\varphi = A_k(a_i)$ and $\varphi = \neg A_k(a_i)$, the claim follows from the definition of $A_k^{\mathcal{I}}$. For $\varphi = P_k(a_i, a_j)$ and $\varphi = \neg P_k(a_i, a_j)$, we have $(a_i, a_j) \in P_k^{\mathcal{I}}$ iff $(a_i, a_j) \in P_k^0$ iff $R(a_i, a_j) \in \mathcal{A}$ and $R \sqsubseteq_{\mathcal{T}}^* P_k$. The challenging part is, however, to show that $\mathcal{I} \models C_1 \sqsubseteq C_2$ whenever $\mathfrak{M} \models \forall x\, (C_1^*(x) \to C_2^*(x))$, for each $\varphi = C_1 \sqsubseteq C_2$. We need to prove that, for all $w \in \Delta^{\mathcal{I}}$ and all $\geq q\, R$ in $\mathcal{T}$,

(a$_1$) $\mathfrak{M} \models E_q R[cp(w)]$ implies $w \in (\geq q\, R)^{\mathcal{I}}$, for all $\geq q\, R$ that occur positively in $\mathcal{T}$;

(a$_2$) $w \in (\geq q\, R)^{\mathcal{I}}$ implies $\mathfrak{M} \models E_q R[cp(w)]$, for all $\geq q\, R$ that occur negatively in $\mathcal{T}$.

In order to do that, we demonstrate the following property of the unravelling construction, for all $w \in W_m$:

$$r_m(R, w) \quad \leq \sum_{R_i \in dsub_{\mathcal{T}}(R)} rank(R_i, \mathcal{T}) + \begin{cases} rank(R, \mathcal{A}), & \text{if } m = 0, \\ pred(R, \mathcal{T}), & \text{if } m > 0. \end{cases} \tag{2}$$

First, note that we have, for all $w \in W_m$:

$$r_m(R,w) \;=\; s_w^R \;+\; \sharp\{w' \in W_m \mid (w,w') \in R^m\} \;\leq\; s_w^R + \begin{cases} \mathit{rank}(R,\mathcal{A}), & \text{if } m = 0,\\ \mathit{pred}(R,\mathcal{T}), & \text{if } m > 0,\end{cases}$$

where

$$s_w^R \;=\; \sharp\{w' \in W_{m+1} \setminus W_m \mid (w,w') \in R_i^{m+1}, R_i \in \mathit{dsub}_{\mathcal{T}}(R)\}.$$

Indeed, the case $m = 0$ is immediate from the definition of the P_k^0; if $m > 0$ then the second component of the sum does not exceed 1 because every such w is introduced to cure a defect of another $w' \in W_{m-1}$ and can be 1 only if an R_1-defect of w' was cured, for $R_1 \sqsubseteq_{\mathcal{T}}^* R^-$ and $\mathit{lb}(R_1,\mathcal{T}) \geq 1$. Now, by induction on the topological order in $G = (\mathit{Rep}_{\mathcal{T}}, E)$, we show that $s_w^R \leq \sum_{R_i \in \mathit{dsub}_{\mathcal{T}}(R)} \mathit{rank}(R_i,\mathcal{T})$. For the basis of induction, $\mathit{dsub}_{\mathcal{T}}(R) = \emptyset$ and so, by definition, $s_w^R = 0$ and the inequality trivially holds. For the inductive step, let $R_1,\dots,R_k$ be the direct sub-roles of R. If w has an R_i-successor w' that does not belong to any of its sub-roles, i.e., $(w,w') \in R_i^{m+1} \setminus \bigcup_{R_{ij} \in \mathit{dsub}_{\mathcal{T}}(R_i)} R_{ij}^{m+1}$, then R_i had a defect on w, which was cured, and therefore, $s_w^{R_i} \leq r(R_i, \mathit{cp}(w))$. Then, by (1) and the definition of *rank*, $r(R_i,\mathit{cp}(w)) \leq \mathit{lb}(R_i,\mathcal{T}) \leq \mathit{rank}(R_i,\mathcal{T})$, whence $s_w^{R_i} \leq \mathit{rank}(R_i,\mathcal{T})$. Otherwise, all R_i-successors of w come from its direct sub-roles, in which case $s_w^{R_i} = \sum_{R_{ij} \in \mathit{dsub}_{\mathcal{T}}(R_i)} s_w^{R_{ij}}$, whence, by the induction hypothesis, $s_w^{R_i} \leq \sum_{R_{ij} \in \mathit{dsub}_{\mathcal{T}}(R_i)} \mathit{rank}(R_{ij},\mathcal{T})$ and, by the definition of *rank*, $s_w^{R_i} \leq \mathit{rank}(R_i,\mathcal{T})$. In either case, $s_w^R = \sum_{R_i \in \mathit{dsub}_{\mathcal{T}}(R)} s_w^{R_i} \leq \sum_{R_i \in \mathit{dsub}_{\mathcal{T}}(R)} \mathit{rank}(R_i,\mathcal{T})$ and so, (2) holds.

We then proceed by showing **(a$_1$)** and **(a$_2$)** as follows:

(a$_1$) If $\geq q\,R$ occurs positively in $\mathcal{T}$ and $\mathfrak{M} \models E_q R[\mathit{cp}(w)]$ then, by the definition of the required rank, $q \leq r(R,\mathit{cp}(w))$ and so, the construction ensures that $w \in (\geq q\,R)^{\mathcal{I}}$.

(a$_2$) We consider the following three subcases:

- Let $\mathit{dsub}_{\mathcal{T}}(R) = \emptyset$. Suppose $w \in (\geq q\,R)^{\mathcal{I}}$. If $w \in W_0$ and there are $w_1,\dots,w_{q'} \in W_0$ with $q' \geq q$ and $(w,w_1),\dots,(w,w_{q'}) \in R^{\mathcal{I}}$ then, by $\mathcal{A}^{\ddagger_e}$, $\mathfrak{M} \models E_{q'} R[\mathit{cp}(w)]$ whence, by $\delta_R(x)$, $\mathfrak{M} \models E_q R[\mathit{cp}(w)]$. Otherwise, some $w' \in \Delta^{\mathcal{I}} \setminus W_0$ with $(w,w') \in R^{\mathcal{I}}$ was introduced to cure an R-defect of w and so $q \leq r(R,\mathit{cp}(w))$. Let $q' = r(R,\mathit{cp}(w))$. Then $\mathfrak{M} \models E_{q'} R[\mathit{cp}(w)]$ and, by $\delta_R(x)$, we obtain $\mathfrak{M} \models E_q R[\mathit{cp}(w)]$.
- Let $\mathit{dsub}_{\mathcal{T}}(R) \neq \emptyset$ and $\mathit{ub}(R,\mathcal{T}) = \infty$. Since $\geq q\,R$ occurs negatively in $\mathcal{T}$ then, by definition, $q = 1$. Suppose $w \in (\exists R)^{\mathcal{I}}$. If $w \in W_0$ and there is w' with $w' \in W_0$ and $(w,w') \in R^{\mathcal{I}}$ then, by $\mathcal{A}^{\ddagger_e}$ and $\delta_R(x)$, $\mathfrak{M} \models E_1 R[\mathit{cp}(w)]$. Otherwise, some $w' \in \Delta^{\mathcal{I}} \setminus W_0$ was introduced to cure an R_1-defect of w for some $R_1 \sqsubseteq_{\mathcal{T}}^* R$. It follows then that $r(R_1,\mathit{cp}(w)) \geq 1$ and so, $\mathfrak{M} \models E_1 R_1[\mathit{cp}(w)]$ whence, by $\mathcal{T}^{\mathcal{R}}(x)$, $\mathfrak{M} \models E_1 R[\mathit{cp}(w)]$.
- Let $\mathit{dsub}_{\mathcal{T}}(R) \neq \emptyset$ and $\mathit{ub}(R,\mathcal{T}) \neq \infty$. We show $(\geq q\,R)^{\mathcal{I}} = \emptyset$. Assume, to the contrary, there is $w \in (\geq q\,R)^{\mathcal{I}}$. Since $\geq q\,R$ occurs negatively in $\mathcal{T}$ and $\mathit{ub}(R,\mathcal{T}) \neq \infty$, $q > \mathit{ub}(R,\mathcal{T})$. By **(inter$_{KB}$)** and the definition of the required rank, $\mathit{ub}(R,\mathcal{T}) \geq \mathit{lb}(R,\mathcal{T}) \geq r(R,\mathit{cp}(w))$, whence $q > r(R,\mathit{cp}(w))$. On the other hand, $w \in W_m$, for some $m \geq 0$, and, by **(inter$_{KB}$)** and (2), $\mathit{ub}(R,\mathcal{T}) \geq r_m(R,w)$, whence $q > r_m(R,w)$. Then, since $w \in (\geq q\,R)^{\mathcal{I}}$, an R-defect was cured on w, and so, as the procedure (if applied) does not create more than $r(R,\mathit{cp}(w))$-many R-successors, we have $q \leq r(R,\mathit{cp}(w))$, contrary to $q > r(R,\mathit{cp}(w))$.

Finally, we can prove that, for all $C_1 \sqsubseteq C_2 \in \mathcal{T}$,

$$\mathfrak{M} \models \forall x\,\big(C_1^*(x) \to C_2^*(x)\big) \qquad \text{implies} \qquad \mathcal{I} \models C_1 \sqsubseteq C_2.$$

It should be clear that each $C_1 \sqsubseteq C_2$ is equivalent to a set of concept inclusions in the following normal form

$$\top \sqsubseteq D_1 \sqcup \dots \sqcup D_k,$$

where each D_i is either $\bot$, A, $\neg A$, $\geq q\,R$ or $\neg(\geq q\,R)$. It is to be noted that $\geq q\,R$ occurs positively in such concept inclusion if it occurs positively in $C_1 \sqsubseteq C_2$ and negatively if negatively in $C_1 \sqsubseteq C_2$. So, suffice it to prove that, for each concept inclusion,

$$\mathfrak{M} \models \forall x\, \big(D_1^*(x) \vee \cdots \vee D_k^*(x)\big) \qquad \text{implies} \qquad \mathcal{I} \models \top \sqsubseteq D_1 \sqcup \cdots \sqcup D_k.$$

Let $w \in \Delta^{\mathcal{I}}$. Then, we have $\mathfrak{M} \models D_i^*[cp(w)]$, for some $1 \leq i \leq k$. Obviously, D_i is not $\bot$. If D_i is A or $\neg A$ then we clearly have $w \in D_i^{\mathcal{I}}$. If D_i is $\geq q\,R$ then $\geq q\,R$ occurs positively in $\mathcal{T}$ and, by $(\mathbf{a}_1)$, $w \in (\geq q\,R)^{\mathcal{I}}$. If D_i is $\neg(\geq q\,R)$ then $\geq q\,R$ occurs negatively in $\mathcal{T}$ and, by $(\mathbf{a}_2)$, $w \notin (\geq q\,R)^{\mathcal{I}}$. In any case $w \in D_i^{\mathcal{I}}$ and so, $\mathcal{I} \models \top \sqsubseteq D_1 \sqcup \cdots \sqcup D_k$. ☐

THEOREM 3. *Under* $(\mathbf{inter}_{KB})$*, KB satisfiability is* NP*-complete in DL-Lite*$_{bool}^{\mathcal{HN}}$*,* PTIME*-complete in DL-Lite*$_{horn}^{\mathcal{HN}}$ *and* NLOGSPACE*-complete inDL-Lite*$_{krom}^{\mathcal{HN}}$ *and DL-Lite*$_{core}^{\mathcal{HN}}$*.*

4 Extending with Attributes

In this section we define the notion of *safe* datatypes and show that such restrictions are required for preserving data complexity of query answering; restriction (*i*) has been independently introduced by (Savković, 2011; Savkovic and Calvanese, 2012).

DEFINITION 4. *A set of datatypes* $\mathcal{D} = \{T_1, \ldots, T_n\}$ *is called* safe *if* (i) *the difference between an arbitrary intersection of datatypes and an arbitrary union of datatypes is either empty or unbounded*; (ii) *all constraints between datatypes are in the form of Horn clauses* $T_{i_1} \cap \cdots \cap T_{i_k} \subseteq_{\mathcal{D}} T_{i_0}$.

A set of datatypes $\mathcal{D}$ *is called* weakly safe *if* (i′) *arbitrary intersections of datatypes are either empty or unbounded and* (ii) *holds.*

It follows, in particular, that if $\mathcal{D}$ is (weakly) safe we can assume that each non-empty datatype T_i is unbounded (note that query answering becomes CONP-hard in presence of datatypes of specific cardinalities (Franconi *et al.*, 2011)); and if $\mathcal{D}$ is safe then also arbitrary intersections of datatypes are either empty or unbounded. Thus, if $\mathcal{D}$ is safe then it is also weakly safe. Condition (*ii*) ensures that datatype constraints in $\mathcal{D}$ have the form of Horn clauses, $T_1 \cap \cdots \cap T_k \subseteq_{\mathcal{D}} T$, and thus computable in PTIME; we further restrict datatype constraints to $T_1 \subseteq_{\mathcal{D}} T_2$ and $T_1 \cap \cdots \cap T_k \subseteq_{\mathcal{D}} \bot_{\mathcal{D}}$ when dealing with the *core* language. Indeed, allowing covering constraints between datatypes leads to CONP-hardness of conjunctive query answering:

THEOREM 5. *Conjunctive query answering in DL-Lite*$_{core}^{\mathcal{HNA}}$ *with covering constraints on datatypes is* CONP*-hard, even without role and attribute inclusions and number restrictions* (*and so, under* $(\mathbf{inter}_{KB})$).

Proof. We prove the result by reduction of the complement of 2+2CNF (similar to instance checking in $\mathcal{ALE}$ (Schaerf, 1993)). Suppose we are given a CNF ψ in which every clause contains two positive and two negative literals (including the constants *true*, *false*). Let T be a datatype covered by non-empty disjoint T_0 and T_1. Let $\mathcal{T}$ contain the following concept inclusions for an attribute U and concepts B and C: $B \sqsubseteq \exists U$, $B \sqsubseteq \forall U.T$, $C \sqsubseteq \forall U.T_0$, and consider the following conjunctive query

$$\begin{aligned} \mathsf{q} = \exists y, \vec{t}, \vec{u}\, \big(P_1(y,t_1) \wedge P_2(y,t_2) \wedge N_1(y,t_3) \wedge N_2(y,t_4) \\ \wedge U(t_1,u_1) \wedge U(t_2,u_2) \wedge U(t_3,u_3) \wedge U(t_4,u_4) \\ \wedge T_0(u_1) \wedge T_0(u_2) \wedge T_1(u_3) \wedge T_1(u_4)\big) \end{aligned}$$

with roles P_1, P_2, N_1 and N_2. We construct an ABox $\mathcal{A}_\psi$ with individuals *true* and *false* for the propositional constants, an individual x_i, for each propositional variable x_i in ψ, and an individual c_i, for each clause of ψ. Let $\mathcal{A}_\psi$ contain assertion $B(x_i)$, for each propositional variable x_i in ψ, assertions $C(\mathit{false}), U(\mathit{true}, v_1)$, for a value v_1 of datatype T_1, and the following assertions, for each clause $x_{j_{i1}} \vee x_{j_{i2}} \vee \neg x_{j_{i3}} \vee \neg x_{j_{i4}}$ of ψ:

$$P_1(c_i, x_{j_{i1}}), P_2(c_i, x_{j_{i2}}), N_1(c_i, x_{j_{i3}}), N_2(c_i, x_{j_{i4}})$$

(here the x_j may include propositional constants). It is readily checked that $(\mathcal{T}, \mathcal{A}_\psi) \not\models \mathsf{q}$ iff ψ is satisfiable. Indeed, if ψ is satisfiable we construct $\mathcal{I}$ by 'extending' $\mathcal{A}_\psi$ by $U(x_i, v_0)$ if x_i is false in the satisfying assignment and by $U(x_i, v_1)$ otherwise, where v_0 is in T_0 and v_1 in T_1 (recall that these datatypes are non-empty and disjoint). Conversely, if $(\mathcal{T}, \mathcal{A}_\psi) \not\models \mathsf{q}$ then there is a model $\mathcal{I}$ of $(\mathcal{T}, \mathcal{A}_\psi)$ in which q is false. Then the satisfying assignment can be defined as follows: a propositional variable x_i is true if one of the attribute U values of x_i belongs to datatype T_1—it does not matter whether other values belong to T_0 or not, the negative answer to the query q guarantees that ψ is true under such an assignment. □

To illustrate condition (*i*), let us consider two datatypes: $T_1 = (> 0 \ \texttt{Int})$, positive integers, and $T_2 = (> 1 \ \texttt{Int})$, integers greater than 1. Let $\mathcal{T} = \{C \sqsubseteq \forall U.T_1,\ C \sqsubseteq \exists U,\ D \sqsubseteq (\geq 2\ U)\}$. Then the query $\mathsf{q}_1 = \exists y\,\big(U(a,y) \land T_2(y)\big)$ has a positive answer over $\mathcal{T}$ and $\mathcal{A}_1 = \{C(a), D(a)\}$ due to $T_1 \setminus T_2$ containing a single element and a having at least 2 U-attributes in T_1. On the other hand, both $\mathsf{q}_2 = \exists x, y\,\big(U(x,y) \land T_2(y)\big)$ and $\mathsf{q}_3 = \exists y\,\big(U(a_1,y) \land U(a_2,y)\big)$ have a negative answer over $\mathcal{T}$ and $\mathcal{A}_2 = \{C(a_1), C(a_2)\}$ albeit for two different reasons—the former is false in a model where both a_1, a_2 share integer 1 as their U-attribute, while the latter is false in a model with different U-attributes for a_1, a_2. These examples show that the construction of the canonical model when difference between datatypes is bounded—if it exists at all—should handle datatypes in an ad hoc way. The following theorem shows that without condition (*i*) we lose FO-rewritability of conjunctive queries in the presence of number restrictions.

THEOREM 6. *Conjunctive query answering in DL-Lite$_{core}^{\mathcal{HNA}}$ KBs with datatypes not respecting condition* (i) *of Definition 4 is* coNP*-hard, even without role and attribute inclusions* (*and so, under* **(inter**$_{KB}$**)**).

Proof. We modify the proof of Theorem 5. Assume that the difference between a datatype, T, and a union of two datatypes, T_0 and T_1, has a finite cardinality, say k. We replace the concept inclusion $B \sqsubseteq \exists U$ with $B \sqsubseteq\ \geq (k+1)\,U$, which forces a choice of at least one U attribute value to be in either T_0 or T_1. In the former case, as before, we assume that the propositional variable gets value *false*, while in the latter case it gets value *true*. □

Thus, the safe condition essentially disallows the use of *enumerations* and any datatype whose non-empty intersection or difference has a finite number of elements. From now on we consider only (weakly) safe datatypes.

4.1 Combined Complexity of KB Satisfiability

We start by showing that addition of attributes to the Bool, Horn and core languages does not change the complexity of KB satisfiability.

THEOREM 7. *Under restriction* **(inter**$_{KB}$**)**, *checking KB satisfiability with weakly safe datatypes is* NP*-complete in DL-Lite$_{bool}^{\mathcal{HNA}}$,* PTIME*-complete in DL-Lite$_{horn}^{\mathcal{HNA}}$ and* NLOGSPACE*-complete in DL-Lite$_{core}^{\mathcal{HNA}}$.*

Proof. We encode a *DL-Lite*$_\alpha^{\mathcal{HNA}}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ in a first-order sentence $\mathcal{K}^{\ddagger_{\mathsf{a}}}$ with one variable in a way similar to the translation of Lemma 2. Denote by *att*$(\mathcal{K})$ the set of all attribute names in $\mathcal{K}$ and by *val*$(\mathcal{A})$ the set of all value names in $\mathcal{A}$. Similarly to roles, we define the sets $Q_{\mathcal{T}}^U$ containing 1 and all q for occurrences of $\geq q\,U$ (including sub-attributes). The set of all datatype names in $\mathcal{K}$ is denoted *dt*$(\mathcal{K})$.

We take a unary predicate $E_qU(x)$, for each attribute name U and $q \in Q_{\mathcal{T}}^U$, denoting the set of objects with at least q values for the attribute U. We also take, for each attribute name U and each datatype name T, a unary predicate $UT(x)$, denoting the objects such that all their U attribute values belong to the datatype T (if they have attribute U values at all). Following this intuition, we extend $\cdot^*$ by the following statements:

$$(\geq q\,U)^* = E_qU(x), \qquad (\forall U.\bot_{\mathcal{D}})^* = \neg(\exists U)^* \quad \text{and} \qquad (\forall U.T_i)^* = UT_i(x).$$

The following sentence encodes the knowledge base $\mathcal{K}$:

$$\mathcal{K}^{\ddagger_{\mathsf{a}}} \ = \ \mathcal{K}^{\ddagger_{\mathsf{e}}} \ \land \ \forall x\left[\mathcal{T}^{\mathcal{U}}(x) \land \beta(x) \land \bigwedge_{U \in att(\mathcal{K})} \big(\delta_U(x) \land \theta_U(x)\big)\right] \ \land \ \mathcal{A}^{\ddagger_{\mathsf{a}}},$$

where $\mathcal{K}^{\ddagger_{\mathbf{e}}}$ is as in Section 3.2, $\mathcal{T}^{\mathcal{U}}(x)$ and $\delta_U(x)$ are similar to $\mathcal{T}^{\mathcal{R}}(x)$ and $\delta_R(x)$, but rephrased for attributes and their inclusions.

Attributes are involved in both existential and universal quantification. So, $\mathcal{T}^{\mathcal{U}}$ reflects the fact that if an object has an attribute U value (existential quantifier $\exists U$) then it also has a U' value, for each U' with $U \sqsubseteq^*_{\mathcal{T}} U'$; universal quantification propagates the datatypes in the opposite direction:

$$\beta(x) \;=\; \bigwedge_{U' \sqsubseteq U \in \mathcal{T}} \;\; \bigwedge_{T \in dt(\mathcal{K})} ((\forall U.T)^* \to (\forall U'.T)^*).$$

We also need a formula that captures the relationships between datatypes, as defined by the Horn clauses in $\mathcal{D}$, for all attributes U:

$$\theta_U(x) \;=\; \bigwedge_{T_1 \cap \dots \cap T_k \subseteq_{\mathcal{D}} T} \big((\forall U.T_1)^* \wedge \dots \wedge (\forall U.T_k)^* \to (\forall U.T)^*\big).$$

We note that the formula $\theta_U(x)$, in particular for disjoint datatypes, e.g., with $T_1 \cap T_2 \subseteq_{\mathcal{D}} \bot_{\mathcal{D}}$, demonstrates a subtle interaction between attribute range constraints, $\forall U.T$, and minimal cardinality constraints, $\exists U$ (see Theorem 9).

The attribute assertions in the ABox require the following formula:

$$\mathcal{A}^{\ddagger_{\mathbf{a}}} = \bigwedge_{U(a_i, v_j) \in \mathcal{A}} \Big[E_{q^{\mathbf{a}}_{U,a_i}} U(a_i) \;\;\wedge \bigwedge_{\substack{T \in dt(\mathcal{K}) \\ val(v_j) \notin val(T)}} \neg (\forall U.T)^*(a_i) \Big],$$

where $q^{\mathbf{a}}_{U,a_i}$ is defined similarly to $q^{\mathbf{e}}_{R,a_i}$.

Lemma 8. *$\mathcal{K}$ is satisfiable iff the $\mathcal{QL}^1$-sentence $\mathcal{K}^{\ddagger_{\mathbf{a}}}$ is satisfiable.*

Proof. ($\Leftarrow$) Let $\mathfrak{M} \models \mathcal{K}^{\ddagger_{\mathbf{a}}}$, we construct a model $\mathcal{I} = (\Delta^{\mathcal{I}}_O \cup \Delta^{\mathcal{I}}_V, \cdot^{\mathcal{I}})$ of $\mathcal{K}$ similarly to the way we proved Lemma 2 but this time datatypes will have to be taken into account. Let $\Delta^{\mathcal{I}}_O$ be defined inductively as before. Then, for each datatype T, let $T^{\mathcal{I}} = val(T)$ and let $\Delta^{\mathcal{I}}_V = \bigcup_{T \in dt(\mathcal{K})} val(T)$. We define $v_j^{\mathcal{I}} = val(v_j)$, for all value names $v_j \in val(\mathcal{A})$. For each attribute name U, to 'cure' its defects we begin with

$$U^0 = \{(a^{\mathfrak{M}}, val(v)) \mid U'(a, v) \in \mathcal{A}, U' \sqsubseteq^*_{\mathcal{T}} U\}.$$

For every attribute name U, we can define the *required* U-rank $r(U, d)$ of $d \in D$ and the *actual* U-rank $r_m(R, w)$ of a point $w \in W_m \subseteq \Delta^{\mathcal{I}}_O,\ m \geq 0$, as before, treating U as a role name. We can also consider the equivalence relation induced by the sub-attribute relation in $\mathcal{T}$, then we can choose representatives and a linear order on them respecting the sub-attribute relation of $\mathcal{T}$. We can start from the smaller attributes and 'cure' their defects. Let U_k be the smallest attribute name not considered so far. For each $w \in W_m$, let $q = r(U_k, cp(w)) - r_m(U_k, w)$. If the datatypes are safe and $q > 0$, take q fresh values $v_1, \dots, v_q \in \Delta^{\mathcal{I}}_V$ that belong *only* to datatypes T with $\mathfrak{M} \models U_k T[cp(w)]$. To show that it is possible, denote by $\mathcal{D}_w \subseteq \mathcal{D}$ the set of all datatypes T with $\mathfrak{M} \models U_k T[cp(w)]$. Consider the difference between the intersection $\bigcap \mathcal{D}_w$ of all $\mathcal{D}_w$ and the union $\bigcup(\mathcal{D} \setminus \mathcal{D}_w)$ of all datatypes in $\mathcal{D} \setminus \mathcal{D}_w$. We claim that this set always contains fresh datatype values $v_1, \dots, v_q$. By condition (*i*) of Definition 4, it suffices to show that the difference is not empty. For the sake of contradiction, assume that $(\bigcap \mathcal{D}_w) \setminus (\bigcup(\mathcal{D} \setminus \mathcal{D}_w)) = \emptyset$. Since $r(U_k, cp(w)) > 0$, we obtain $\bot_{\mathcal{D}} \notin \mathcal{D}_w$ and so, all the datatypes in $\mathcal{D}_w$ are non-empty. Then we have $(\bigcap \mathcal{D}_w) \subseteq (\bigcup(\mathcal{D} \setminus \mathcal{D}_w))$, which by condition (*ii*) of Definition 4, implies that there is $T_0 \in \mathcal{D} \setminus \mathcal{D}_w$ such that $\bigcap \mathcal{D}_w \subseteq_{\mathcal{D}} T_0$. Since the formula θ_U ensures that $\mathcal{D}_w$ respects all datatype inclusions $T_1 \cap \dots \cap T_k \subseteq_{\mathcal{D}} T_0$, the datatype T_0 must belong to $\mathcal{D}_w$, contrary to our definition. If the datatypes are weakly safe rather than safe, we still can take $v_1, \dots, v_n$ in the intersection of all the datatypes T with $\mathfrak{M} \models U_k T[cp(w)]$ (the values may, however, belong to additional datatypes, but this is irrelevant for the satisfiability of $\mathcal{K}$). So, for each $1 \leq j \leq q$, add the pair (w, v_j) to all attribute relations U^0 with $U_k \sqsubseteq^*_{\mathcal{T}} U$. Denote the relations resulting in applying the above procedure to all attributes by $U^{\mathcal{I}}$.

Now, it can be shown that if $\mathfrak{M} \models \mathcal{K}^{\ddagger_{\mathbf{a}}}$ then $\mathcal{I} \models \varphi$ for every $\varphi \in \mathcal{K}$. Consider $C \sqsubseteq \forall U.T \in \mathcal{T}$. Take any $w \in C^{\mathcal{I}}$ and suppose $(w, v) \in U^{\mathcal{I}}$, for some $v \in \Delta^{\mathcal{I}}_V$. We have $\mathfrak{M} \models \forall x\, \big(C^*(x) \to (\forall U.T)^*(x)\big)$

and, by Lemma 2, $\mathfrak{M} \models C^*[cp(w)]$, whence $\mathfrak{M} \models (\forall U.T)^*[cp(w)]$. By construction of $U^{\mathcal{I}}$, two cases are possible. If $v \in val(\mathcal{A})$ then $w = cp(w) = a \in ob(\mathcal{A})$ and $U'(a,v) \in \mathcal{A}$, for $U' \sqsubseteq_{\mathcal{T}}^* U$, whence, by $\beta(x)$, we have $\mathfrak{M} \models (\forall U'.T)^*[a]$ and by the second conjunct of $\mathcal{A}^{\ddagger_{\mathbf{a}}}$, we have $v \in val(T) = T^{\mathcal{I}}$. Otherwise, $w \in W_m$ and v was introduced to cure a defect of w for some $U' \sqsubseteq_{\mathcal{T}}^* U$, in which case, by $\beta(x)$, $\mathfrak{M} \models (\forall U'.T)[cp(w)]$, and so, by construction of $U'^{\mathcal{I}}$, we obtain $v \in T^{\mathcal{I}}$.

For the other kinds of formulas the proof is similar to that one of Lemma 2.

($\Rightarrow$) Conversely, if $\mathcal{I}$ is a model of $\mathcal{K}$ with the domain $\Delta^{\mathcal{I}} = \Delta^{\mathcal{I}}_O \cup \Delta^{\mathcal{I}}_V$ we construct a model $\mathfrak{M} = (D, \cdot^{\mathfrak{M}})$ of $\mathcal{K}^{\ddagger_{\mathbf{a}}}$ with $D = \Delta^{\mathcal{I}}_O$. The only difference with the proof of Lemma 2 is how to define $UT^{\mathfrak{M}}$: we set

$$UT^{\mathfrak{M}} = \big\{ w \in \Delta^{\mathcal{I}}_O \mid v \in T^{\mathcal{I}}, \text{ for all } (w,v) \in U^{\mathcal{I}} \big\},$$

for every attribute U and every datatype name T. It follows that $\mathfrak{M} \models \mathcal{K}^{\ddagger_{\mathbf{a}}}$ ☐

Now, we are in a position to complete the proof of Theorem 7. It can be seen that, for a KB $\mathcal{K}$, the formula $\mathcal{K}^{\ddagger_{\mathbf{a}}}$ is of the form:

$$\forall x\, \psi(x) \quad \wedge \bigwedge_{R \in role^{\pm}(\mathcal{K})} \forall x \left((\exists inv(R))^*(x) \to (\exists R)^*(dr) \right) \quad \wedge \varphi,$$

where $\psi(x)$ is a quantifier-free formula with only unary predicates and without constants and φ is a conjunction of ground atoms. So, $\mathcal{K}^{\ddagger_{\mathbf{a}}}$ is satisfiable iff there is a subset Ξ of $role^{\pm}(\mathcal{K})$ such that the conjunction of all

$$\psi(a) \quad \wedge \bigwedge_{R \in \Xi} (\exists R)^*(dr) \quad \wedge \bigwedge_{R \in role^{\pm}(\mathcal{K}) \setminus \Xi} \neg(\exists R)^*(dr) \quad \wedge \quad \varphi,$$

for constants a from $ob(\mathcal{A}) \cup val(\mathcal{A}) \cup dr(\mathcal{K})$, is satisfiable. This suggests the following NP-algorithm for checking satisfiability of $\mathcal{K}^{\ddagger_{\mathbf{a}}}$: guess Ξ and then, for each constant a, guess a set of unary predicates Θ (a *type*) and check whether this choice is consistent with the above formula in the sense that if the predicates in Θ are assumed true and those not in Θ false, the formula is true. This consistency check, though, is not straightforward as the length of ψ is polynomial in the length of $\mathcal{K}$ but, due to the $\theta_U(x)$, can be exponential in the number of datatypes. So, all the conjuncts of $\psi(a)$ but $\theta_U(a)$ can be straightforwardly checked in time polynomial in $|\mathcal{K}|$, while to check the datatype constraints the following is enough: take each negative predicate $(\forall U.T)^*(a)$ and verify that T is *not* implied by the intersection of all positive predicates $(\forall U.T_i)^*(a)$ (if it is implied than the guess was incorrect; here we use the fact that datatype constraints are Horn formulas).

Next, if $\mathcal{K}$ is a KB with a Horn TBox, we do not need to guess Ξ and the types Θ: they can be constructed in a bottom-up fashion as the formula $\mathcal{K}^{\ddagger_{\mathbf{a}}}$ is a Horn formula, which is satisfiable iff it is satisfiable in a minimal model.

Finally, if $\mathcal{K}$ is a KB with a core TBox then the datatype inclusions can only be of the form $T_1 \subseteq_{\mathcal{D}} T_2$, and $T_1 \cap \cdots \cap T_k \subseteq_{\mathcal{D}} \bot_{\mathcal{D}}$. We observe that the clauses of ψ can only be of the form $\forall x\,(B_1(x) \to B_2(x))$ or $\forall x\,(B_1(x) \wedge \cdots \wedge B_k(x) \to \bot)$; the latter type of clauses come either from the θ_U or from disjointness axioms of the form $B_1 \sqsubseteq \neg B_2$. As the minimal model construction procedure can make use only of the former type of clauses, it can be done in NLogSpace because all the clauses are binary. The disjointness clauses $\forall x\,(B_1(x) \wedge \cdots \wedge B_k(x) \to \bot)$ can be checked in an on-the-fly manner, which gives us the NLogSpace upper bound. ☐

It is of interest to note that the complexity of KB satisfiability increases in the case of Krom TBoxes:

Theorem 9. *Satisfiability of DL-Lite$^{\mathcal{HNA}}_{krom}$ KBs is* NP*-hard with a single pair of disjoint datatypes, even without role and attribute inclusions and number restrictions* (*and so, under* **(inter$_{KB}$)**).

Proof. The proof is by reduction of 3SAT. It exploits the structure of the formula $\theta_U(x)$ in $\mathcal{K}^{\ddagger_{\mathbf{a}}}$: if datatypes T and T' are disjoint then the concept inclusion

$$\forall U.T \sqcap \forall U.T' \sqcap \exists U \sqsubseteq \bot,$$

although not in the syntax of *DL-Lite*$^{\mathcal{HNA}}_{krom}$, is a logical consequence of $\mathcal{T}$. Using such ternary intersections with the full negation of the Krom fragment one can encode 3SAT. Let $\varphi = \bigwedge_{i=1}^{m} C_i$ be a 3CNF, where the

C_i are ternary clauses over variables $p_1, \dots, p_n$. Now, suppose $p_{i_1} \lor \neg p_{i_2} \lor p_{i_3}$ is the ith clause of φ. It is equivalent to $\neg p_{i_1} \land p_{i_2} \land \neg p_{i_3} \to \bot$ and so, can be encoded as follows:

$$\neg A_{i_1} \sqsubseteq \forall U_i.T, \qquad A_{i_2} \sqsubseteq \forall U_i.T', \qquad \neg A_{i_3} \sqsubseteq \exists U_i,$$

where $A_1, \dots, A_n$ are concept names for variables $p_1, \dots, p_n$, and U_i is an attribute for the ith clause (note that Krom concept inclusions of the form $\neg B \sqsubseteq B'$ are required, which is not allowed in the core TBoxes). Let $\mathcal{T}$ consist of all such inclusions for clauses in φ. It can be seen that φ is satisfiable iff $\mathcal{T}$ is satisfiable. □

4.2 Data Complexity of Query Answering

In this section we study the data complexity of answering positive existential queries over a KB expressed in languages with attributes and datatypes. In the following, we slightly abuse notation and use H for an attribute name U, a role name P or inverse role P^-.

REMARK 10. It follows from the proofs of Theorems 7 and Lemma 2 that, for a *DL-Lite*$_{bool}^{\mathcal{HNA}}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ under restriction **(inter**$_{KB}$**)**, every model $\mathfrak{M}$ of $\mathcal{K}^{\ddagger_\mathsf{a}}$ induces a forest-shaped model $\mathcal{I}_\mathfrak{M}$ of $\mathcal{K}$ with the following properties:

(forest) The ABox constants $a \in ob(\mathcal{A}) \cup val(\mathcal{A})$ induce a partitioning of $\Delta^{\mathcal{I}_\mathfrak{M}}$ into disjoint labelled trees $\mathfrak{T}_a = (T_a, E_a, \ell_a)$ with nodes T_a, edges E_a, root $a^{\mathcal{I}_\mathfrak{M}}$, and a labelling function ℓ_a that assigns a role or an attribute name to each edge (indicating a minimal, w.r.t. $\sqsubseteq_\mathcal{T}^*$, role or attribute name that required a fresh successor due to an existential quantifier); the trees for $v \in val(\mathcal{A})$ are degenerate, i.e., contain a single node, v.

(copy) There is a function, $cp\colon \Delta^{\mathcal{I}_\mathfrak{M}} \to ob(\mathcal{A}) \cup val(\mathcal{A}) \cup dr(\mathcal{K})$ such that $cp(a^{\mathcal{I}_\mathfrak{M}}) = a$, if $a \in ob(\mathcal{A}) \cup val(\mathcal{A})$, and $cp(w) = dr$, if $(w', w) \in E_a$ and $\ell_a(w', w) = inv(R)$, for $w' \in T_a$.

(role) For every role (attribute) name H,

$$H^{\mathcal{I}_\mathfrak{M}} = \big\{(a_i, a_j) \mid H'(a_i, a_j) \in \mathcal{A},\ H' \sqsubseteq_\mathcal{T}^* H\big\} \cup \\ \big\{(w, w') \in E_a \mid \ell_a(w, w') = H',\ H' \sqsubseteq_\mathcal{T}^* H,\ a \in ob(\mathcal{A})\big\}.$$

THEOREM 11. *With safe datatypes, the positive existential query answering problem for DL-Lite*$_{horn}^{\mathcal{HNA}}$ *and DL-Lite*$_{core}^{\mathcal{HNA}}$*, under restriction* **(inter**$_{KB}$**)***, is in* AC^0 *for data complexity.*

Proof. Suppose that we are given a *consistent* *DL-Lite*$_{horn}^{\mathcal{HNA}}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and a positive existential query in prenex form $\mathsf{q}(\vec{x}) = \exists \vec{y}\, \varphi(\vec{x}, \vec{y})$ in the signature of $\mathcal{K}$. Since $\mathcal{K}^{\ddagger_\mathsf{a}}$ is a Horn sentence, it is enough to consider just one special model $\mathcal{I}_0$ of $\mathcal{K}$. Let $\mathfrak{M}_0$ be the *minimal Herbrand model* of (the universal Horn sentence) $\mathcal{K}^{\ddagger_\mathsf{a}}$. We remind the reader (for details consult, e.g., (Apt, 1990; Rautenberg, 2006)) that $\mathfrak{M}_0$ can be constructed by taking the intersection of all Herbrand models for $\mathcal{K}^{\ddagger_\mathsf{a}}$, that is, of all models based on the domain that consists of the constant symbols from $\mathcal{K}^{\ddagger_\mathsf{a}}$—i.e., $ob(\mathcal{A}) \cup val(\mathcal{A}) \cup dr(\mathcal{K})$.

Let $\mathcal{I}_0$ be the *canonical* model of $\mathcal{K}$, i.e., the model induced by $\mathfrak{M}_0$ along the construction presented in Theorem 7. Denote the domain of $\mathcal{I}_0$ by $\Delta^{\mathcal{I}_0}$. The following properties follow from the construction of the canonical model: for all basic concepts B and datatypes T,

$$a_i^{\mathcal{I}_0} \in B^{\mathcal{I}_0} \text{ iff } \mathcal{K} \models B(a_i), \quad \text{for } a_i \in ob(\mathcal{A}), \tag{3}$$

$$w \in B^{\mathcal{I}_0} \text{ iff } \mathcal{T} \models \exists R \sqsubseteq B, \quad \text{for } w \in \Delta_O^{\mathcal{I}_0} \text{ with } cp(w) = dr. \tag{4}$$

$$v_i^{\mathcal{I}_0} \in T^{\mathcal{I}_0} \text{ iff } val(v_i) \in val(T), \quad \text{for } v_i \in val(\mathcal{A}), \tag{5}$$

$$v \in T^{\mathcal{I}_0} \text{ iff there are } B_1, \dots, B_k \text{ such that } w \in B_1^{\mathcal{I}_0}, \dots, B_k^{\mathcal{I}_0} \text{ and } \mathcal{T} \models B_1 \sqcap \dots \sqcap B_k \sqsubseteq \forall U.T, \\ \text{for } (w, v) \in U^{\mathcal{I}_0} \text{ with } v^{\mathcal{I}_0} \notin val(\mathcal{A}). \tag{6}$$

Formula (3) describes conditions when a named object, a_i, belongs to a basic concept, B, in the canonical model $\mathcal{I}_0$—we say it describes the *type* of a_i. Similarly, (4) describes types of unnamed objects, which are copies of the dr, for roles R; it is worth pointing out that those types are determined by a single concept, $\exists R$. The same two properties were used in the proof of he proof of Theorem 7.1 (Artale *et al.*, 2009). The other two properties are specific to datatypes: (5) describes the type of a named datatype value and (6) the type of an unnamed datatype value. We note that (6) holds only for safe datatypes, and even weakly safe datatypes cannot guarantee that in the process of unravelling it is always possible, for every $w \in (\exists U)^{\mathcal{I}_0}$, to pick a fresh attribute U value of the 'minimal type', i.e., a datatype value that belongs *only* to datatypes T with $w \in (\forall U.T)^{\mathcal{I}_0}$. It can be shown that the canonical model $\mathcal{I}_0$ provides correct answers to all queries:

LEMMA 12. *$\mathcal{K} \models \mathsf{q}(\vec{a})$ iff $\mathcal{I}_0 \models \mathsf{q}(\vec{a})$, for all $\vec{a}$.*

Proof. Suppose $\mathcal{I}_0 \models \mathcal{K}$. As $\mathsf{q}(\vec{a})$ is a positive existential sentence, it is enough to construct a homomorphism $h\colon \mathcal{I}_0 \to \mathcal{I}$. By property **(forest)** of Remark 10, the domain $\Delta^{\mathcal{I}_0}$ of $\mathcal{I}_0$ is partitioned into disjoint trees $\mathfrak{T}_a$, for $a \in ob(\mathcal{A}) \cup val(\mathcal{A})$. Define the *depth* of a point $w \in \Delta^{\mathcal{I}_0}$ to be the length of the shortest path in the respective tree to its root. Denote by W_m the set of points of depth $\leq m$ (including also values $v \in \Delta_V^{\mathcal{I}_0}$) that were taken to satisfy existential quantifiers for objects in W_{m-1}; in particular, $W_0 = ob(\mathcal{A}) \cup val(\mathcal{A})$.

We construct h as the union of maps h_m, $m \geq 0$, where each h_m is defined on W_m and has the following properties: $h_{m+1}(w) = h_m(w)$, for all $w \in W_m$, and

($\mathbf{a}_m$) for all $w \in W_m$, if $w \in B^{\mathcal{I}_0}$ then $h_m(w) \in B^{\mathcal{I}}$, for each basic concept B;

($\mathbf{b}_m$) for all $u, w \in W_m$, if $(u, w) \in R^{\mathcal{I}_0}$ then $(h_m(u), h_m(w)) \in R^{\mathcal{I}}$, for each $R \in role^{\pm}(\mathcal{K})$;

($\mathbf{t}_m$) for all $v \in W_m$, if $v \in T^{\mathcal{I}_0}$ then $h_m(v) \in T^{\mathcal{I}}$, for each datatype name T;

($\mathbf{v}_m$) for all $u, v \in W_m$, if $(u, v) \in U_k^{\mathcal{I}_0}$ then $(h_m(u), h_m(v)) \in U_k^{\mathcal{I}}$, for each $U_k \in att(\mathcal{K})$.

For the basis of induction, we set $h_0(a_i) = a_i^{\mathcal{I}}$, for $a_i \in ob(\mathcal{A})$, and $h_0(v_i) = v_i^{\mathcal{I}}$, for $v_i \in val(\mathcal{A})$. Property **($\mathbf{a}_0$)** follows then from (3), **($\mathbf{t}_0$)** from (5) and **($\mathbf{b}_0$)** and **($\mathbf{v}_0$)** from **(role)**.

For the induction step, suppose that h_m has already been defined for W_m, $m \geq 0$. Set $h_{m+1}(w) = h_m(w)$ for all $w \in W_m$. Consider an arbitrary $w \in W_{m+1} \setminus W_m$. By **(forest)**, there is a unique $u \in W_m$ such that $(u, w) \in E_a$, for some $\mathfrak{T}_a$.

- Let $\ell_a(u, w) = S \in role^{\pm}(\mathcal{K})$. Then, by **(copy)**, $cp(w) = inv(ds)$. By **(role)**, $u \in (\exists S)^{\mathcal{I}_0}$ and, by **($\mathbf{a}_m$)**, $h_m(u) \in (\exists S)^{\mathcal{I}}$, which means that there is $w_1 \in \Delta^{\mathcal{I}}$ with $(h_m(u), w_1) \in S^{\mathcal{I}}$. Set $h_{m+1}(w) = w_1$. As $cp(w) = inv(ds)$ and $(\exists inv(S))^{\mathcal{I}_0} \neq \emptyset$, it follows from (4) that if $w \in B^{\mathcal{I}_0}$ then $w' \in B^{\mathcal{I}}$ whenever we have $w' \in (\exists inv(S))^{\mathcal{I}}$. As $w_1 \in (\exists inv(S))^{\mathcal{I}}$, we obtain **($\mathbf{a}_{m+1}$)** for w. To show **($\mathbf{b}_{m+1}$)**, we notice that, by **(role)**, we have $(u, w) \in R^{\mathcal{I}_0}$ just when $S \sqsubseteq_{\mathcal{T}}^* R$. Thus, since $(u, w) \in S^{\mathcal{I}_0}$ and $(h_{m+1}(u), h_{m+1}(w)) \in S^{\mathcal{I}}$ and, as $S \sqsubseteq_{\mathcal{T}}^* R$, then, $(h_{m+1}(u), h_{m+1}(w)) \in R^{\mathcal{I}}$.
- Let $\ell_a(u, w) = U \in att(\mathcal{K})$, then $w = v \in W_{m+1} \cap \Delta_V^{\mathcal{I}_0}$. By **(role)**, $u \in (\exists U)^{\mathcal{I}_0}$ and, by **($\mathbf{a}_m$)**, $h_m(u) \in (\exists U)^{\mathcal{I}}$, which means that there is $v_1 \in \Delta_V^{\mathcal{I}}$ with $(h_m(u), v_1) \in U^{\mathcal{I}}$. Set $h_{m+1}(v) = v_1$. To show **($\mathbf{v}_{m+1}$)**, we notice that, by **(role)**, we have $(u, v) \in U_k^{\mathcal{I}_0}$ just when $U \sqsubseteq_{\mathcal{T}}^* U_k$; but then we have $(h_{m+1}(u), h_{m+1}(v)) \in U^{\mathcal{I}} \subseteq U_k^{\mathcal{I}}$. Next, we show **($\mathbf{t}_{m+1}$)**. By definition, $v \notin val(\mathcal{A})$. If $v \in T^{\mathcal{I}_0}$ then, by (6), there are basic concepts $B_1, \dots, B_k$ such that $\mathcal{T} \models B_1 \sqcap \dots \sqcap B_k \sqsubseteq \forall U.T$ and $u \in B_1^{\mathcal{I}_0}, \dots, B_k^{\mathcal{I}_0}$. By **($\mathbf{a}_{m+1}$)**, we have $h_{m+1}(u) \in B_1^{\mathcal{I}}, \dots, B_k^{\mathcal{I}}$, whence $h_{m+1}(u) \in (\forall U.T)^{\mathcal{I}}$ and so, as $(h_{m+1}(u), h_{m+1}(v)) \in U^{\mathcal{I}}$, we obtain $h_{m+1}(v) \in T^{\mathcal{I}}$.

This completes the proof of the lemma. ☐

Our next lemma shows that in this case to check whether $\mathcal{I}_0 \models \mathsf{q}(\vec{a})$ it suffices to consider only the points of depth $\leq m_0$ in $\Delta^{\mathcal{I}_0}$, for some m_0 that does not depend on $|\mathcal{A}|$:

LEMMA 13. *If $\mathcal{I}_0 \models \exists \vec{y}\, \varphi(\vec{a}, \vec{y})$ then there is an assignment $\mathfrak{a}_0$ in W_{m_0} such that $\mathcal{I}_0 \models^{\mathfrak{a}_0} \varphi(\vec{a}, \vec{y})$ and $\mathfrak{a}_0(y_i) \in W_{m_0}$, for all $y_i \in \vec{y}$, where $m_0 = |\vec{y}| + |role^{\pm}(\mathcal{T})| + 1$.*

Proof. The proof is similar to that one of Lemma 7.4 in (Artale *et al.*, 2009) observing that attributes cannot be nested and cannot have role successors either. ☐

To complete the proof of Theorem 11, we encode the problem '$\mathcal{I}_0 \models \mathsf{q}(\vec{a})$?' as a model checking problem for first-order formulas over the ABox $\mathcal{A}$ considered as a first-order model, denoted by $\mathfrak{A}_\mathcal{A}$, with domain $\mathit{ob}(\mathcal{A}) \cup \mathit{val}(\mathcal{A})$; we assume that this first-order model also contains all datatype extensions. Now we define a first-order formula $\varphi_{\mathcal{T},\mathsf{q}}(\vec{x})$ in the signature of $\mathcal{T}$ and q such that *(i)* $\varphi_{\mathcal{T},\mathsf{q}}(\vec{x})$ depends on $\mathcal{T}$ and q but not on $\mathcal{A}$, and *(ii)* $\mathfrak{A}_\mathcal{A} \models \varphi_{\mathcal{T},\mathsf{q}}(\vec{a})$ iff $\mathcal{I}_0 \models \mathsf{q}(\vec{a})$.

Denote by $\mathit{con}(\mathcal{K})$ the set of basic concepts in $\mathcal{K}$ together with all concepts of the form $\forall U.T$, for attribute names U and datatypes T from $\mathcal{T}$. We begin by defining formulas $\psi_B(x)$, for $B \in \mathit{con}(\mathcal{K})$, that describe the types of named objects (cf. (3)): for all $a_i \in \mathit{ob}(\mathcal{A})$,

$$\mathfrak{A}_\mathcal{A} \models \psi_B(a_i) \;\text{ iff }\; a_i^{\mathcal{I}_0} \in B^{\mathcal{I}_0}, \quad \text{if } B \text{ is a basic concept,} \tag{7}$$

$$\mathfrak{A}_\mathcal{A} \models \psi_{\forall U.T}(a_i) \;\text{ iff }\; a_i^{\mathcal{I}_0} \in B_1^{\mathcal{I}_0}, \dots, B_k^{\mathcal{I}_0} \text{ and } \mathcal{T} \models B_1 \sqcap \dots \sqcap B_k \sqsubseteq \forall U.T. \tag{8}$$

These formulas are defined as the 'fixed-points' of sequences $\psi_B^0(x), \psi_B^1(x), \dots$:

$$\psi_B^0(x) = \begin{cases} A(x), & \text{if } B = A,\\ \exists y_1 \dots \exists y_q \big(\bigwedge_{1 \le i < j \le q} (y_i \ne y_j) \land \bigwedge_{1 \le i \le q} H^\mathcal{T}(x, y_i) \big), & \text{if } B = \geq q\, H,\\ \bot, & \text{if } B = \forall U.T, \end{cases}$$

$$\psi_B^i(x) = \psi_B^0(x) \lor \bigvee_{B_1 \sqcap \dots \sqcap B_k \sqsubseteq B \in \mathsf{ext}(\mathcal{T})} \big(\psi_{B_1}^{i-1}(x) \land \dots \land \psi_{B_k}^{i-1}(x) \big),$$

where

$$H^\mathcal{T}(x,y) = \bigvee_{H' \sqsubseteq_\mathcal{T}^* H} H'(x,y),$$

and $\mathsf{ext}(\mathcal{T})$ denotes the extension of $\mathcal{T}$ with the following concept inclusions:

- $\geq q'\, H \sqsubseteq\, \geq q\, H$, for all $q, q' \in Q_\mathcal{T}^H$ with $q' > q$ and all $H \in \mathit{role}^\pm(\mathcal{T}) \cup \mathit{att}(\mathcal{T})$,
- $\geq q\, H \sqsubseteq\, \geq q\, H'$, for all $H \sqsubseteq_\mathcal{T}^* H'$ and $q \in Q_\mathcal{T}^H$,
- $\forall U.T \sqsubseteq \forall U'.T$, for all $U' \sqsubseteq_\mathcal{T}^* U$ and $T \in \mathit{dt}(\mathcal{K})$,
- $\forall U.T_1 \sqcap \dots \sqcap \forall U.T_k \sqsubseteq \forall U.T$, for all $T_1 \cap \dots \cap T_k \subseteq_\mathcal{D} T$.

It should be clear that there is N with $\psi_B^N(x) \equiv \psi_B^{N+1}(x)$, for all B at the same time, and that N does not exceed the cardinality of $\mathit{con}(\mathcal{T})$. We set $\psi_B(x) = \psi_B^N(x)$.

Next we define sentences $\theta_{B,dr}$, for $B \in \mathit{con}(\mathcal{K})$ and dr with $R \in \mathit{role}^\pm(\mathcal{K})$, that describe types of the unnamed points, i.e., copies of the dr (cf. (4)): for all w with $\mathit{cp}(w) = dr$,

$$\mathfrak{A}_\mathcal{A} \models \theta_{B,dr} \;\text{ iff }\; w \in B^{\mathcal{I}_0}, \;\text{ if } B \text{ is a basic concept,} \tag{9}$$

$$\mathfrak{A}_\mathcal{A} \models \theta_{\forall U.T,dr} \;\text{ iff }\; \mathcal{T} \models \exists R \sqsubseteq \forall U.T. \tag{10}$$

Note that the type of copies of dr is determined by a single concept, $\exists R$, and therefore, there is no need to consider conjunctions in (10); see also (8). We inductively define a sequence $\theta_{B,dr}^0, \theta_{B,dr}^1, \dots$ by taking $\theta_{B,dr}^0 = \top$, if $B = \exists R$, and $\theta_{B,dr}^0 = \bot$, otherwise, and

$$\theta_{B,dr}^i = \theta_{B,dr}^0 \lor \bigvee_{B_1 \sqcap \dots \sqcap B_k \sqsubseteq B \in \mathsf{ext}(\mathcal{T})} \big(\theta_{B_1,dr}^{i-1} \land \dots \land \theta_{B_k,dr}^{i-1} \big).$$

As with the ψ_B, we set $\theta_{B,dr} = \theta_{B,dr}^N$.

Now, suppose $\mathcal{I}_0 \models^{\mathfrak{a}_0} \varphi(\vec{a}, \vec{y})$ and $\mathfrak{a}_0(y_i) \in W_{m_0}$, for every $y_i \in \vec{y}$, where m_0 is as in Lemma 13. Recall that our aim is to compute the answer to this query in the first-order model $\mathfrak{A}_\mathcal{A}$ representing the ABox. This model, however, does not contain points in $W_{m_0} \setminus W_0$, and to represent them, we use the following 'trick.' By **(forest)**, every $w \in W_{m_0}$ is uniquely determined by a pair (a, σ), where a is the root of the tree $\mathfrak{T}_a$ containing w and σ is the sequence of labels $\ell_a(u, v)$ on the path from a to w. Not every such pair, however, corresponds to an element in W_{m_0}. In order to identify points of W_{m_0}, we consider the following directed graph $G_\mathcal{T} = (V_\mathcal{T}, E_\mathcal{T})$, where $V_\mathcal{T}$ is the set of equivalence classes $[H] = \{H' \mid H \sqsubseteq^*_\mathcal{T} H' \text{ and } H' \sqsubseteq^*_\mathcal{T} H\}$ and $E_\mathcal{T}$ is the set of all pairs $([R], [H])$ such that

(p) $\mathcal{T} \models \exists R^- \sqsubseteq \exists H$ and $R^- \not\sqsubseteq^*_\mathcal{T} H$,

and H has no proper sub-role/attribute satisfying **(p)**. Let $\Sigma_{\mathcal{T},m_0}$ be the set of all paths in the graph $G_\mathcal{T}$ of length $\leq m_0$: more precisely,

$$\Sigma_{\mathcal{T},m_0} = \{\varepsilon\} \cup V_\mathcal{T} \cup \{([H_1], \dots, [H_n]) \mid 2 \leq n \leq m_0 \text{ and } ([H_j], [H_{j+1}]) \in E_\mathcal{T}, \text{ for } 1 \leq j < n\}.$$

By the unravelling procedure, $\sigma \in \Sigma_{\mathcal{T},m_0}$, for all pairs (a, σ) representing elements of W_{m_0}. We note, however, that a pair (a, σ) with $\sigma = ([H], \dots) \in \Sigma_{\mathcal{T},m_0}$ corresponds to a $w \in W_{m_0}$ only if a has not enough H-witnesses in $\mathfrak{A}_\mathcal{A}$ (see the last conjunct of (11) below).

In the first-order rewriting $\varphi_{\mathcal{T},\mathfrak{q}}$ we are about to define we assume that the bound variables y_i range over W_0 and represent the first component of the pairs (a, σ) (these y_i should not be confused with the y_i in the original query $\mathfrak{q}$, which range over W_{m_0}), whereas the second component is encoded in the ith member σ_i of a vector $\vec{\sigma}$. Note that constants and free variables do not need a second component, σ, and, to unify the notation, for each term t we denote its σ-component by $t^{\vec{\sigma}}$, which is defined as follows: $t^{\vec{\sigma}} = \varepsilon$ if t is a constant or free variable and $t^{\vec{\sigma}} = \sigma_i$ if $t = y_i$.

Let k be the number of bound variables y_i and let $\Sigma^k_{\mathcal{T},m_0}$ be the set of k-tuples $\vec{\sigma} = (\sigma_1, \dots, \sigma_k)$ with $\sigma_i \in \Sigma_{\mathcal{T},m_0}$. Given an assignment $\mathfrak{a}_0$ in W_{m_0}, we denote by *split*$(\mathfrak{a}_0)$ the pair $(\mathfrak{a}, \vec{\sigma})$ made of an assignment $\mathfrak{a}$ in $\mathcal{A}$ and $\vec{\sigma} \in \Sigma^k_{\mathcal{T},m_0}$ such that $t^{\vec{\sigma}} = ([H_1], \dots, [H_n])$, for a sequence $H_1, \dots, H_n$ of ℓ_a-labels on the path from a to $\mathfrak{a}_0(t)$. We define now, for every $\vec{\sigma} \in \Sigma^k_{\mathcal{T},m_0}$, concept name A, role or attribute name H and datatype name T:

$$A^{\vec{\sigma}}(t) = \begin{cases} \psi_A(t), & \text{if } t^{\vec{\sigma}} = \varepsilon, \\ \theta_{A,\mathit{inv}(ds)}, & \text{if } t^{\vec{\sigma}} = \sigma'.[S], \end{cases}$$

$$H^{\vec{\sigma}}(t_1, t_2) = \begin{cases} H^\mathcal{T}(t_1, t_2), & \text{if } t_1^{\vec{\sigma}} = t_2^{\vec{\sigma}} = \varepsilon, \\ (t_1 = t_2), & \text{if } t_1^{\vec{\sigma}}.[S] = t_2^{\vec{\sigma}}, \text{ or } t_2^{\vec{\sigma}} = t_1^{\vec{\sigma}}.[S^-], \text{ for } S \sqsubseteq^*_\mathcal{T} H, \\ \bot, & \text{otherwise}, \end{cases}$$

$$T^{\vec{\sigma}}(t) = \begin{cases} T(t), & \text{if } t^{\vec{\sigma}} = \varepsilon, \\ \psi_{\forall U.T}(t), & \text{if } t^{\vec{\sigma}} = [U], \\ \theta_{\forall U.T, ds^-}, & \text{if } t^{\vec{\sigma}} = \sigma'.[S].[U]. \end{cases}$$

LEMMA 14. *For each assignment $\mathfrak{a}_0$ in W_{m_0} with split$(\mathfrak{a}_0) = (\mathfrak{a}, \sigma)$,*

$$\mathcal{I}_0 \models^{\mathfrak{a}_0} A(t) \text{ iff } \mathcal{A} \models^{\mathfrak{a}} A^{\vec{\sigma}}(t), \text{ for concept names } A,$$
$$\mathcal{I}_0 \models^{\mathfrak{a}_0} H(t_1, t_2) \text{ iff } \mathcal{A} \models^{\mathfrak{a}} H^{\vec{\sigma}}(t_1, t_2), \text{ for roles and attribute names } H,$$
$$\mathcal{I}_0 \models^{\mathfrak{a}_0} T(t) \text{ iff } \mathcal{A} \models^{\mathfrak{a}} T^{\vec{\sigma}}(t), \text{ for datatype names } T.$$

Proof. For atoms of the form $A(a)$, $A(x_i)$ and $A(y_i)$ with $\sigma_i = \varepsilon$ the claim follows from (7). For $A(y_i)$ with $\sigma_i = \sigma'.[S]$, by **(copy)**, we have $\mathit{cp}(\mathfrak{a}_0(y_i)) = \mathit{inv}(dr)$, for $R \in [S]$; the claim then follows from (9).

For atoms of the form $T(v)$, $T(x_i)$ and $T(y_i)$ with $\sigma_i = \varepsilon$ the claim follows from the fact that the interpretation of datatypes is fixed and independent from the particular interpretation $\mathcal{I}$ (recall that $\mathfrak{A}_\mathcal{A}$ includes all datatypes extensions). Consider $T(y_i)$ with $\sigma_i \neq \varepsilon$. Let $v = \mathfrak{a}_0(y_i)$. Then $v \notin W_0$ and, by **(role)**, there is $w \in \Delta^{\mathcal{I}_0}$ such that w and v are in the same tree $\mathfrak{T}_a$, for some $a \in \mathit{ob}(\mathcal{A})$, with $(w, v) \in E_a$ and $\ell_a(w, v) = U_i$, for some $U_i \sqsubseteq U$, i.e., $\sigma_i = \sigma'.[U]$, and $(w, v) \in U^{\mathcal{I}_0}$. By (6), $v \in T^{\mathcal{I}_0}$ iff $w \in B_1^{\mathcal{I}_0}, \dots, B_k^{\mathcal{I}_0}$ such that $\mathcal{T} \models B_1 \sqcap \dots \sqcap B_k \sqsubseteq \forall U.T$, which is equivalent to $\mathfrak{A}_\mathcal{A} \models^{\mathfrak{a}} T^{\vec{\sigma}}(y_i)$. Indeed, if $w = a \in \mathit{ob}(\mathcal{A})$, then, by (8), this is equivalent to $\mathfrak{A}_\mathcal{A} \models \psi_{\forall U.T}(a_i)$; otherwise, $\mathit{cp}(w) = dr^-$ and $\sigma = \sigma'.[S].[U]$ for some $R \in [S]$ and then, by (4) and (10), it is equivalent to $\mathfrak{A}_\mathcal{A} \models \theta_{\forall U.T,\mathit{inv}(ds)}$.

Finally, for atoms of the form $H(y_{i_1}, y_{i_2})$ with $\sigma_{i_1} = \sigma_{i_2} = \varepsilon$, the claim follows from **(role)**. Consider now the case of $H(y_{i_1}, y_{i_2})$ with $\sigma_{i_2} \neq \varepsilon$: we have $\mathfrak{a}_0(y_{i_2}) \notin W_0$ and thus, by **(role)**, $\mathcal{I}_0 \models^{\mathfrak{a}_0} H(y_{i_1}, y_{i_2})$ iff

- $\mathfrak{a}_0(y_{i_1}), \mathfrak{a}_0(y_{i_2})$ are in the same tree $\mathfrak{T}_a$, for $a \in ob(\mathcal{A})$, i.e., $\mathfrak{A}_{\mathcal{A}} \models^{\mathfrak{a}} (y_{i_1} = y_{i_2})$,
- and either $(\mathfrak{a}_0(y_{i_1}), \mathfrak{a}_0(y_{i_2})) \in E_a$ and then $\ell_a(\mathfrak{a}_0(y_{i_1}), \mathfrak{a}_0(y_{i_2})) = S$ for some $S \sqsubseteq^*_{\mathcal{T}} H$, or $(\mathfrak{a}_0(y_{i_2}), \mathfrak{a}_0(y_{i_1})) \in E_a$ and then $\ell_a(\mathfrak{a}_0(y_{i_2}), \mathfrak{a}_0(y_{i_1})) = S$ for some $inv(S) \sqsubseteq^*_{\mathcal{T}} H$.

Other cases are similar and left to the reader. □

Finally, we define the first-order rewriting of $\mathsf{q}(\vec{x}) = \exists \vec{y}\, \varphi(\vec{x}, \vec{y})$ with $\vec{y} = y_1, \dots, y_k$ and $\mathcal{T}$ by taking:

$$\varphi_{\mathcal{T},\mathsf{q}}(\vec{x}) \;=\; \exists \vec{y} \bigvee_{\vec{\sigma} \in \Sigma^k_{\mathcal{T},m_0}} \Big(\varphi^{\vec{\sigma}}(\vec{x}, \vec{y}) \;\wedge \bigwedge_{\substack{1 \leq i \leq |\vec{y}| \\ \sigma_i = ([H_i], \dots) \neq \varepsilon}} \big(\neg \psi^0_{\exists H_i}(y_i) \wedge \psi_{\exists H_i}(y_i)\big)\Big), \tag{11}$$

where $\varphi^{\vec{\sigma}}(\vec{x}, \vec{y})$ is the result of attaching the superscript $^{\vec{\sigma}}$ to each atom of φ; the last conjunct ensures that each pair (a, σ_i) corresponds an element of $w \in W_{m_0}$. By Lemma 14, for every assignment $\mathfrak{a}_0$ in W_{m_0}, we have $\mathcal{I}_0 \models^{\mathfrak{a}_0} \varphi(\vec{x}, \vec{y})$ iff $\mathfrak{A}_{\mathcal{A}} \models^{\mathfrak{a}} \varphi^{\vec{\sigma}}(\vec{x}, \vec{y})$ for $(\mathfrak{a}, \sigma) = split(\mathfrak{a}_0)$. For the converse direction observe that if the second conjunct of $\varphi_{\mathcal{T},\mathsf{q}}(\vec{x})$ is true in $\mathfrak{A}_{\mathcal{A}}$ under an assignment $\mathfrak{a}$ then there is an assignment $\mathfrak{a}_0$ in W_{m_0} with $split(\mathfrak{a}_0) = (\mathfrak{a}, \vec{\sigma})$. □

5 Conclusions

We studied two different extensions of the *DL-Lite* languages. First, we considered the interaction between cardinality constraints and role inclusions and their impact on the complexity of satisfiability. We presented two alternative restrictions both relaxing the one analyzed by (Artale *et al.*, 2009), where roles with sub-roles cannot have maximum cardinality constraints. Our results imply that if the complexity of the KB satisfiability problem is to remain low, the number of R-successors in the ABox has to be taken into account (e.g., **(inter$_{KB}$)**); otherwise, under the condition **(inter$_{\mathcal{T}}$)**, complexity of KB satisfiability becomes NP-hard, even for the core fragment, and EXPTIME-complete even for the Horn fragment.

Then we considered *local attributes* that allow the use of the same attribute associated to different concepts with different datatype range restrictions (with Horn clauses defining relationship between datatypes). Notably, this is the first time that *DL-Lite* is equipped with a form of the universal restriction $\forall U.T$. We showed that such an extension is harmless with the only exception of the Krom fragment, where the complexity rises from NLOGSPACE to NP. We studied also the problem of answering positive existential queries and showed that for the Horn and core languages the problem remains in AC^0(i.e., FO-rewritable).

As a future work, given the encouraging results obtained here, we aim at better clarifying the connection of this work with the literature on concrete domains and analyzing the influence of different concrete domains on the complexity of the logics.

References

K. Apt. Logic programming. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics*, pages 493–574. Elsevier and MIT Press, 1990.

A. Artale, D. Calvanese, R. Kontchakov, V. Ryzhikov, and M. Zakharyaschev. Reasoning over extended ER models. In *Proc. of the 26th Int. Conf. on Conceptual Modeling (ER 2007)*, volume 4801 of *Lecture Notes in Computer Science*, pages 277–292. Springer, 2007.

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.

A. Artale, D. Calvanese, R. Kontchakov, and M. Zakharyaschev. The *DL-Lite* family and relations. *Journal of Artificial Intelligence Research*, 36:1–69, 2009.

F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence, IJCAI-05*. Morgan-Kaufmann Publishers, 2005.

D. Berardi, D. Calvanese, and G. De Giacomo. Reasoning on UML class diagrams. *Artificial Intelligence*, 168(1–2):70–118, 2005.

D. Calvanese, M. Lenzerini, and D. Nardi. Unifying class-based representation formalisms. *Journal of Artificial Intelligence Research*, 11:199–240, 1999.

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)*, pages 602–607, 2005.

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.

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. *Journal of Automated Reasoning*, 39(3):385–429, 2007.

B. Cuenca Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, and U. Sattler. OWL 2: The next step for OWL. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4):309–322, 2008.

M. Despoina, Y. Kazakov, and I. Horrocks. Tractable extensions of the description logic EL with numerical datatypes. *Journal of Automated Reasoning*, 2011.

E. Franconi, Y. A. Ibáñez-García, and I. Seylan. Query answering with DBoxes is hard. *Electr. Notes Theor. Comput. Sci.*, 278:71–84, 2011.

C. Lutz. Description logics with concrete domains—a survey. In *Advances in Modal Logics Volume 4*. King's College Publications, 2003.

J. Pan and I. Horrocks. OWL-Eu: Adding customised datatypes into OWL. *Web Semantics: Science, Services and Agents on the World Wide Web*, 4(1), 2011.

A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Linking Data to Ontologies. *Journal on Data Semantics*, X:133–173, 2008.

W. Rautenberg. *A Concise Introduction to Mathematical Logic*. Springer, 2006.

O. Savkovic and D. Calvanese. Introducing datatypes in *DL-Lite*. In *Proc. of the 20th European Conf. on Artificial Intelligence (ECAI 2012)*, 2012.

O. Savković. *Managing Datatypes in Ontology-Based Data Access*. MSc dissertation, European Master in Computational Logic, Faculty of Computer Science, Free University of Bozen-Bolzano, October 2011.

A. Schaerf. On the complexity of the instance checking problem in concept languages with existential quantification. *Journal of Intelligent Information Systems*, 2:265–278, 1993.