DL-Lite with role inclusions

R. Kontchakov and M. Zakharyaschev

School of Computer Science and Information Systems
Birkbeck College London, U.K.
`http://www.dcs.bbk.ac.uk/~{roman,michael}`

Abstract. We analyse *DL-Lite* logics with role inclusions and present a complete classification of the trade-off between their expressiveness and computational complexity. In particular, we show that in logics with role inclusions the data complexity of instance checking becomes P-hard in the presence of functionality constraints, and coNP-hard if arbitrary number restrictions are allowed, even with a very primitive form of concept inclusions. Moreover, the combined complexity of satisfiability in this case jumps to ExpTime. On the positive side, it turns out that the combined complexity for the logics without number restrictions depends only on the form of concept inclusions and can range from NLogSpace and P to NP; the data complexity for such logics stays in LogSpace.

1 Introduction

Description Logic, a discipline conceived in the early 1990s as a family of knowledge representation formalisms, which stemmed from semantic networks and frames, has now been recognised as a 'cornerstone of the Semantic Web' for providing a formal basis for the Web Ontology Language (OWL). *DL-Lite* is part of OWL 1.1 (which is currently a W3C Member Submission); it belongs to the group of its fragments that 'can handle at least some interesting inference service in *polynomial time* with respect to either the number of facts in the ontology or the size of the ontology as a whole.' Although in many practical cases state-of-the-art DL reasoners can cope quite well with reasoning tasks of much higher worst-case complexity, new challenges are arising that require really tractable reasoning. Typical examples are ontologies with huge terminologies or a huge number of facts (data). *DL-Lite* was specially tailored to provide efficient query answering on the data, which becomes increasingly important in the context of data integration [18], the Semantic Web [15], P2P data management [3, 10, 13] and ontology-based data access [5, 7]. For instance, in a standard data integration scenario, the information about objects and relationships between them (ABox assertions, in the DL parlance) is stored in relational databases, while a specially designed ontology (TBox) defines a new 'logical' view of the stored data so that the user can query the integrated resources in terms of this ontology. In this case, one may be interested in the *combined complexity* of reasoning, when both TBox and ABox are regarded to be the input (e.g., to check consistency), as well as the *data complexity*, i.e., the complexity of solving a problem (say, instance checking or query answering) when the TBox and the query are fixed and only the ABox may vary. *DL-Lite* boasts polynomial combined complexity and LogSpace data complexity. Moreover, every conjunctive query to a *DL-Lite* ontology can be rewritten into a first-order query to the underlying databases, which can be expressed, say, in SQL, and thus the existing relational database engines can be used to execute these queries.

The idea of *DL-Lite* has actually given rise not to a single language but rather a family of related formalisms [6–8, 11]. Some of them are expressive enough to capture Extended Entity-Relationship diagrams [1], others enjoy particularly simple procedures for rewriting queries into SQL [8]. Unfortunately, a mechanical union of two languages of the family can easily ruin their nice computation properties. This situation poses a general research problem of investigating the impact of various DL constructs on the computational complexity of reasoning in *DL-Lite* logics. The impact of Boolean operators in concept inclusions as well as arbitrary number restrictions on roles was comprehensively analysed in [2].

In this paper, we focus on *DL-Lite* languages with *role inclusion axioms* (which are indispensable in data modelling) and present a complete picture (summarised in Table 1) of the trade-off between their expressiveness and computational complexity. In particular, we show that one cannot keep the data complexity of instance checking in LogSpace and have functionality constraints (or any kind of number restrictions) together with role inclusions in the language. Even logics with very primitive concept inclusions become P-hard for data complexity and coNP-hard if arbitrary number restrictions are allowed. Moreover, the combined complexity of satisfiability in this case jumps to ExpTime (the same as for the full-fledged logic $\mathcal{SHIQ}$ [21]). On the other hand, for the Horn fragment with functionality constraints, instance checking is P-complete for data complexity. Although this problem is not first-order reducible, it can be reformulated in Datalog [16]. On the positive side, it turns out that the combined complexity for the logics with role inclusions but without number restrictions depends only on the form of concept inclusions and can range from NLogSpace and P to NP; the data complexity of instance checking or satisfiability for such logics stays in LogSpace.

2 The *DL-Lite* family and its neighbours

We begin by defining a description logic that can be regarded as the *supremum* of the original *DL-Lite* family [6–8, 11] in the lattice of description logics. This *supremum* will be called *DL-Lite*$^{\mathcal{R},\mathcal{N}}_{bool}$. The language of *DL-Lite*$^{\mathcal{R},\mathcal{N}}_{bool}$ contains *object names* $a_0, a_1, \ldots$, *atomic concept names* $A_0, A_1, \ldots$, and *atomic role names* $P_0, P_1, \ldots$; its complex *roles* R and *concepts* C are defined as follows:

$$B ::= \bot \mid A_i \mid \geq q\, R, \qquad R ::= P_i \mid P_i^-,$$
$$C ::= B \mid \neg C \mid C_1 \sqcap C_2,$$

where $q \geq 1$. The concepts of the form B are called *basic*. A *DL-Lite*$^{\mathcal{R},\mathcal{N}}_{bool}$ *TBox*, $\mathcal{T}$, is a finite set of *concept inclusion* and *role inclusion axioms* of the form

$$C_1 \sqsubseteq C_2 \qquad \text{and} \qquad R_1 \sqsubseteq R_2,$$

and an *ABox*, $\mathcal{A}$, is a finite set of assertions of the form $A_k(a_i)$ and $P_k(a_i, a_j)$. Taken together, $\mathcal{T}$ and $\mathcal{A}$ constitute the *DL-Lite*$^{\mathcal{R},\mathcal{N}}_{bool}$ *knowledge base* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$.

As usual in description logic, an *interpretation* is a structure of the form

$$\mathcal{I} \;=\; \big(\Delta, a_0^{\mathcal{I}}, a_1^{\mathcal{I}}, \dots, A_0^{\mathcal{I}}, A_1^{\mathcal{I}}, \dots, P_0^{\mathcal{I}}, P_1^{\mathcal{I}}, \dots\big), \tag{1}$$

where $\Delta \neq \emptyset$, $a_i^{\mathcal{I}} \in \Delta$, $A_i^{\mathcal{I}} \subseteq \Delta$, $P_i^{\mathcal{I}} \subseteq \Delta \times \Delta$, and $a_i^{\mathcal{I}} \neq a_j^{\mathcal{I}}$, for all $i \neq j$. The role and concept constructors are interpreted in $\mathcal{I}$ in the standard way:

$$\begin{aligned}
(P_i^-)^{\mathcal{I}} &= \{(y,x) \in \Delta \times \Delta \mid (x,y) \in P_i^{\mathcal{I}}\}, && \text{(inverse role)}\\
\bot^{\mathcal{I}} &= \emptyset, && \text{(the empty set)}\\
(\geq q\, R)^{\mathcal{I}} &= \big\{x \in \Delta \mid \sharp\{y \in \Delta \mid (x,y) \in R^{\mathcal{I}}\} \geq q\big\}, && \text{(`at least } q\ R\text{-successors')}\\
(\neg C)^{\mathcal{I}} &= \Delta \setminus C^{\mathcal{I}}, && \text{(`not in } C\text{')}\\
(C_1 \sqcap C_2)^{\mathcal{I}} &= C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}, && \text{(`both in } C_1 \text{ and } C_2\text{')}
\end{aligned}$$

where $\sharp X$ denotes the cardinality of X. We also use the standard abbreviations: $C_1 \sqcup C_2 := \neg(\neg C_1 \sqcap \neg C_2)$, $\top := \neg\bot$, $\exists R := (\geq 1\, R)$ and $\leq q\, R := \neg(\geq q+1\, R)$.

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

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

A knowledge base (KB) $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is said to be *satisfiable* if there is an interpretation satisfying all the members of $\mathcal{T}$ and $\mathcal{A}$; such an interpretation is called a *model of* $\mathcal{K}$.

We will consider restrictions of *DL-Lite*$^{\mathcal{R},\mathcal{N}}_{bool}$ along three axes: Boolean operators ($_{bool}$) on concepts, number restrictions ($\mathcal{N}$) and role inclusions ($\mathcal{R}$). A *DL-Lite*$^{\mathcal{R},\mathcal{N}}_{bool}$ TBox $\mathcal{T}$ is a *Krom* TBox if its concept inclusions are of the form

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

where the B_i are basic concepts. $\mathcal{T}$ is called a *Horn* TBox if its concept inclusions are of the form

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

We use $\bigsqcap_k B_k \sqsubseteq \bigsqcap_i B_i'$ as an abbreviation for the set of inclusions $\bigsqcap_k B_k \sqsubseteq B_i'$. Finally, we call $\mathcal{T}$ a *core* TBox if its concept inclusions are of the form

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

As $B_1 \sqsubseteq \neg B_2$ is equivalent to $B_1 \sqcap B_2 \sqsubseteq \bot$, core TBoxes can be regarded as sitting precisely in the intersection of Krom and Horn TBoxes.

The fragments of *DL-Lite*$^{\mathcal{R},\mathcal{N}}_{bool}$ with Krom, Horn and core TBoxes will be denoted by *DL-Lite*$^{\mathcal{R},\mathcal{N}}_{krom}$, *DL-Lite*$^{\mathcal{R},\mathcal{N}}_{horn}$ and *DL-Lite*$^{\mathcal{R},\mathcal{N}}_{core}$, respectively.

Let $\alpha \in \{\mathit{core}, \mathit{krom}, \mathit{horn}, \mathit{bool}\}$. Denote by *DL-Lite*$^{\mathcal{R},\mathcal{F}}_{\alpha}$ the fragment of *DL-Lite*$^{\mathcal{R},\mathcal{N}}_{\alpha}$ in which number restrictions can occur only in *functionality constraints* of the form $\geq 2\, R \sqsubseteq \bot$ (saying that R is functional: if $(x,y), (x,z) \in R^{\mathcal{I}}$

then $y = z$). The fragment of $\mathit{DL\text{-}Lite}_\alpha^{\mathcal{R},\mathcal{N}}$ without number restrictions at all, i.e., without concepts of the form $\geq q\, R$, is denoted by $\mathit{DL\text{-}Lite}_\alpha^{\mathcal{R}}$. The fragments obtained by omitting the role inclusions—that is, $\mathit{DL\text{-}Lite}_\alpha^{\mathcal{N}}$ (with arbitrary number restrictions), $\mathit{DL\text{-}Lite}_\alpha^{\mathcal{F}}$ (with functionality constraints only), and $\mathit{DL\text{-}Lite}_\alpha$ (without any number restrictions)—have been analysed in [2]. Note that our notation is somewhat different from the original one; cf. [6, 7, 11, 8, 2].

We concentrate on three standard reasoning tasks for our logics $\mathcal{L}$:

- *satisfiability*: given an $\mathcal{L}$-KB $\mathcal{K}$, decide whether $\mathcal{K}$ is satisfiable;
- *instance checking*: given an object name a, a basic concept B and an $\mathcal{L}$-KB $\mathcal{K}$, decide whether $a^{\mathcal{I}} \in B^{\mathcal{I}}$ whenever $\mathcal{I} \models \mathcal{K}$;
- *query answering*: given a positive existential query $q(\boldsymbol{x})$, an $\mathcal{L}$-KB $\mathcal{K}$ and a tuple $\boldsymbol{a}$ of object names from its ABox, decide whether $\mathcal{K} \models q(\boldsymbol{a})$.

As is well known, many other reasoning tasks for description logics are LogSpace reducible to the satisfiability problem; for details see [2]. In particular, this is true of instance checking: an object a is an instance of concept B in every model of $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ iff the KB $(\mathcal{T} \cup \{A_{\neg B} \sqsubseteq \neg B\},\ \mathcal{A} \cup \{A_{\neg B}(a)\})$ is not satisfiable, where $A_{\neg B}$ is a fresh concept name.

Our aim is to investigate (i) the *combined complexity* of the satisfiability problem for the logics of our family, where the whole KB $\mathcal{K}$ is regarded as an input, and (ii) the *data complexity* (or *ABox complexity*) of the instance checking and query answering problems, where the given TBox is assumed to be fixed, while the input ABox can vary. The obtained results are summarised in Table 1.

language name	concept inc.	number restric.	complexity: combined satisfiability	complexity: data: inst. checking	complexity: data: query answering
$\mathit{DL\text{-}Lite}_{core}^{\mathcal{R}}$	core	−	NLogSpace [≤Th.1]	in LogSpace	in LogSpace
$\mathit{DL\text{-}Lite}_{core}^{\mathcal{R},\mathcal{F}}$	core	f	ExpTime [≥Th.3]	P [≥Th.6]	P
$\mathit{DL\text{-}Lite}_{core}^{\mathcal{R},\mathcal{N}}$	core	+	ExpTime	coNP [≥Th.5]	coNP
$\mathit{DL\text{-}Lite}_{krom}^{\mathcal{R}}$	Krom	−	NLogSpace [≤Th.1]	in LogSpace	coNP ≥[7]
$\mathit{DL\text{-}Lite}_{krom}^{\mathcal{R},\mathcal{F}}$	Krom	f	ExpTime	coNP [≥Th.4]	coNP
$\mathit{DL\text{-}Lite}_{krom}^{\mathcal{R},\mathcal{N}}$	Krom	+	ExpTime	coNP	coNP
$\mathit{DL\text{-}Lite}_{horn}^{\mathcal{R}}$	Horn	−	P [≤Th.1]	in LogSpace	in LogSpace [8]
$\mathit{DL\text{-}Lite}_{horn}^{\mathcal{R},\mathcal{F}}$	Horn	f	ExpTime	P ≥[8]	P ≤[12]
$\mathit{DL\text{-}Lite}_{horn}^{\mathcal{R},\mathcal{N}}$	Horn	+	ExpTime	coNP	coNP
$\mathit{DL\text{-}Lite}_{bool}^{\mathcal{R}}$	Bool	−	NP [≤Th.1]	in LogSpace [Th.2]	coNP
$\mathit{DL\text{-}Lite}_{bool}^{\mathcal{R},\mathcal{F}}$	Bool	f	ExpTime	coNP	coNP
$\mathit{DL\text{-}Lite}_{bool}^{\mathcal{R},\mathcal{N}}$	Bool	+	ExpTime ≤[14]	coNP	coNP ≤[14]

Table 1. Complexity of *DL-Lite* logics with role inclusions.

3 $\mathit{DL\text{-}Lite}_{bool}^{\mathcal{R}}$ and first-order logic with one variable

We begin by considering the logic $\mathit{DL\text{-}Lite}_{bool}^{\mathcal{R}}$ and its fragments. The key observation which clearly explains their computational behaviour is that the satisfiability problem for $\mathit{DL\text{-}Lite}_{bool}^{\mathcal{R}}$ knowledge bases is LogSpace reducible to the satisfiability problem for the *one-variable fragment* $\mathcal{QL}^1$ of first-order logic (without equality and function symbols) and that this reduction preserves the properties of core, Krom, or Horn formulas.

Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a $\mathit{DL\text{-}Lite}_{bool}^{\mathcal{R}}$ KB. Denote by $\mathit{role}(\mathcal{K})$ the set of role names occurring in $\mathcal{T}$ and $\mathcal{A}$, by $\mathit{role}^{\pm}(\mathcal{K})$ the set $\{P_k, P_k^- \mid P_k \in \mathit{role}(\mathcal{K})\}$, and by $\mathit{ob}(\mathcal{A})$ the set of object names in $\mathcal{A}$.

With every $a_i \in \mathit{ob}(\mathcal{A})$ we associate the individual constant a_i of $\mathcal{QL}^1$ and with every concept name A_i the unary predicate $A_i(x)$ from the signature of $\mathcal{QL}^1$. For each pair of roles $P_k, P_k^- \in \mathit{role}^{\pm}(\mathcal{K})$, we introduce a pair of fresh unary predicates $EP_k(x)$ and $EP_k^-(x)$, which will represent the domain and range of P_k, respectively (in other words, $EP_k(x)$ and $EP_k^-(x)$ are the sets of points with *at least one* P_k-successor and *at least one* P_k-predecessor, respectively). Additionally, for each pair of roles $P_k, P_k^- \in \mathit{role}^{\pm}(\mathcal{K})$, we take a pair of fresh individual constants dp_k and dp_k^- of $\mathcal{QL}^1$, which will serve as 'representatives' of the points from the domains of P_k and P_k^- (provided that they are not empty). Furthermore, for each pair $a_i, a_j \in \mathit{ob}(\mathcal{A})$ and each $R \in \mathit{role}^{\pm}(\mathcal{K})$, we take a fresh *propositional variable* Ra_ia_j of $\mathcal{QL}^1$ to encode $R(a_i, a_j)$. By induction on the construction of a $\mathit{DL\text{-}Lite}_{bool}^{\mathcal{R}}$ concept C we define the $\mathcal{QL}^1$-formula C^*:

$$\bot^* = \bot, \qquad (A_i)^* = A_i(x), \qquad (\exists R)^* = ER(x),$$
$$(\neg C)^* = \neg C^*(x), \qquad (C_1 \sqcap C_2)^* = C_1^*(x) \land C_2^*(x).$$

A $\mathit{DL\text{-}Lite}_{bool}^{\mathcal{R}}$ TBox $\mathcal{T}$ corresponds then to the $\mathcal{QL}^1$-sentence

$$\mathcal{T}^* = \bigwedge_{C_1 \sqsubseteq C_2 \in \mathcal{T}} \forall x \big(C_1^*(x) \to C_2^*(x)\big) \quad \land$$
$$\bigwedge_{R_1 \sqsubseteq R_2 \in \mathcal{T}} \Big[\forall x \big(ER_1(x) \to ER_2(x)\big) \ \land\ \forall x \big(\mathit{inv}(ER_1)(x) \to \mathit{inv}(ER_2)(x)\big) \Big],$$

where $\mathit{inv}(ER) = EP_k^-$ if $R = P_k$ and $\mathit{inv}(ER) = EP_k$ if $R = P_k^-$. For every role $R \in \mathit{role}^{\pm}(\mathcal{K})$, we also need the following $\mathcal{QL}^1$-sentence:

$$\varepsilon(R) = \forall x \big(ER(x) \to \mathit{inv}(ER)(\mathit{inv}(dr))\big),$$

where $\mathit{inv}(dr) = dp_k^-$ if $R = P_k$ and $\mathit{inv}(dr) = dp_k$ if $R = P_k^-$. This sentence says that if the domain of R is not empty then its range is not empty either: it contains the representative $\mathit{inv}(dr)$.

It should be clear how to translate a $\mathit{DL\text{-}Lite}_{bool}^{\mathcal{R}}$ ABox $\mathcal{A}$ into $\mathcal{QL}^1$:

$$\mathcal{A}^\dagger = \bigwedge_{A(a_i) \in \mathcal{A}} A(a_i) \quad \land \bigwedge_{P(a_i, a_j) \in \mathcal{A}} Pa_ia_j.$$

We also need formulas representing the relationship of the Ra_ia_j with the unary predicates for the role domain and range. For $R \in \mathit{role}^{\pm}(\mathcal{K})$, let

$$R^\dagger = \bigwedge_{a_i, a_j \in \mathit{ob}(\mathcal{A})} \big(Ra_ia_j \to ER(a_i)\big) \quad \land \bigwedge_{a_i, a_j \in \mathit{ob}(\mathcal{A})} \big(Ra_ia_j \to \mathit{inv}(R)a_ja_i\big),$$

where $inv(R)a_j a_i$ is the propositional variable $P_k^- a_j a_i$ if $R = P_k$ and $P_k a_j a_i$ if $R = P_k^-$. Finally, for the $DL\text{-}Lite_{bool}^{\mathcal{R}}$ knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, we set

$$\mathcal{K}^\dagger \; = \; \Big[\; \mathcal{T}^* \quad \wedge \bigwedge_{R \in role^\pm(\mathcal{K})} \varepsilon(R) \; \Big] \quad \wedge \quad \Big[\; \mathcal{A}^\dagger \quad \wedge \bigwedge_{R \in role^\pm(\mathcal{K})} R^\dagger \; \Big].$$

Lemma 1. *A $DL\text{-}Lite_{bool}^{\mathcal{R}}$ knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is satisfiable iff the $\mathcal{QL}^1$-sentence $\mathcal{K}^\dagger$ is satisfiable.*

Proof. Every model for $\mathcal{K}$ gives rise to a model for $\mathcal{K}^\dagger$ in an obvious way. The converse can be proved by 'unravelling' a first-order model for $\mathcal{K}^\dagger$ similarly to the unravelling construction in [2]. We only note here the main difference from that construction for $DL\text{-}Lite_{bool}^{\mathcal{N}}$: as $DL\text{-}Lite_{bool}^{\mathcal{R}}$ has no number restrictions, one can create as many R-successors to a point as required without violating the TBox axioms (it was not the case for $DL\text{-}Lite_{bool}^{\mathcal{N}}$; on the other hand the latter does not have role inclusions, which may force additional R-successors). ❑

As the reduction $\cdot^\dagger$ is computable in LogSpace and $\mathcal{K}^\dagger$ is a universal sentence, we can use the known complexity results for the relevant fragments of $\mathcal{QL}^1$ (see, e.g., [4]):

Theorem 1. *The satisfiability problem is* NLogSpace*-complete for $DL\text{-}Lite_{core}^{\mathcal{R}}$ and $DL\text{-}Lite_{krom}^{\mathcal{R}}$,* P*-complete for $DL\text{-}Lite_{horn}^{\mathcal{R}}$ and* NP*-complete for $DL\text{-}Lite_{bool}^{\mathcal{R}}$.*

Now we show that as far as *data complexity* is concerned, satisfiability of $DL\text{-}Lite_{bool}^{\mathcal{R}}$ KBs can be solved using only *logarithmic space* in the size of the ABox. In what follows, without loss of generality, we assume that all role names of a given KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ occur in its TBox and write $role^\pm(\mathcal{T})$ instead of $role^\pm(\mathcal{K})$. Let $\Sigma(\mathcal{T}) = \{ER(dr) \mid R \in role^\pm(\mathcal{T})\}$, and, for $\Sigma_0 \subseteq \Sigma(\mathcal{T})$, let

$$core_{\Sigma_0}(\mathcal{T}) = \bigwedge_{ER(dr) \in \Sigma_0} ER(dr) \quad \wedge \bigwedge_{R \in role^\pm(\mathcal{T})} \Big(\mathcal{T}^*[dr] \; \wedge \bigwedge_{R' \in role^\pm(\mathcal{T})} \varepsilon(R')[dr] \Big),$$

$$proj_{\Sigma_0}(\mathcal{K}, a) = \bigwedge_{inv(ER)(inv(dr)) \in \Sigma(\mathcal{T}) \setminus \Sigma_0} \neg ER(a) \quad \wedge \quad \mathcal{T}^*[a] \quad \wedge \quad \mathcal{A}^\flat(a),$$

where $\mathcal{T}^*[c]$ and $\varepsilon(R')[c]$ are instantiations of the universal quantifier in the respective formulas with the constant c, and $\mathcal{A}^\flat(a)$ is defined as below:

$$\mathcal{A}^\flat(a) \;\; = \bigwedge_{A(a) \in \mathcal{A}} A(a) \quad \wedge \bigwedge_{P_k \in role(\mathcal{K})} \Big[\bigwedge_{P_k(a, a_1) \in \mathcal{A}} EP_k(a) \wedge \bigwedge_{P_k(a_1, a) \in \mathcal{A}} EP_k^-(a) \Big].$$

Lemma 2. *$\mathcal{K}$ is satisfiable iff there is a subset Σ_0 of $\Sigma(\mathcal{T})$ such that* (i) *$core_{\Sigma_0}(\mathcal{T})$ is satisfiable and* (ii) *$proj_{\Sigma_0}(\mathcal{K}, a)$ is satisfiable for every $a \in ob(\mathcal{A})$.*

Proof. If $\mathcal{I} \models \mathcal{K}$, then we take $\Sigma_0 = \{ER(dr) \mid R \in role^\pm(\mathcal{T}), \; (\exists R)^\mathcal{I} \neq \emptyset\}$ and the first-order model $\mathfrak{M}$ induced by $\mathcal{I}$. It should be clear that we have $\mathfrak{M} \models core_{\Sigma_0}(\mathcal{T})$ and $\mathfrak{M} \models proj_{\Sigma_0}(\mathcal{K}, a)$, for all $a \in ob(\mathcal{A})$.

Conversely, let $\mathfrak{M}_{\Sigma_0}$ be an Herbrand model of $core_{\Sigma_0}(\mathcal{T})$ and $\mathfrak{M}_a$ an Herbrand model of $proj_{\Sigma_0}(\mathcal{K}, a)$, for $a \in ob(\mathcal{A})$. By definition, the domain of $\mathfrak{M}_{\Sigma_0}$ consists of $|role^{\pm}(\mathcal{T})|$ elements and the domains of the $\mathfrak{M}_a$ are singletons. Clearly, $\mathfrak{M}_{\Sigma_0} \models \mathcal{T}^*$ and $\mathfrak{M}_{\Sigma_0} \models \varepsilon(R)$, for every $R \in role^{\pm}(\mathcal{T})$, and $\mathfrak{M}_a \models \mathcal{T}^*$, for every $a \in ob(\mathcal{A})$. We construct a model $\mathfrak{M}$ by taking the disjoint union of $\mathfrak{M}_{\Sigma_0}$ with all of the $\mathfrak{M}_a$, where we set $P_k a_i a_j^{\mathfrak{M}}$ to be true iff $P_{k'}(a_i, a_j) \in \mathcal{A}$ or $P_{k'}^-(a_j, a_i) \in \mathcal{A}$ for a sub-role $P_{k'}$ of P_k. Let us show that $\mathfrak{M} \models \mathcal{K}^\dagger$. We have $\mathfrak{M} \models \mathcal{T}^*$ because $\mathcal{T}^*$ is universal, does not contain constants and is true in every component model. Consider now $\varepsilon(R) = \forall x\, \psi(x)$, where $\psi(x) = (ER(x) \to inv(ER)(inv(dr)))$. We show that, for every d in the domain of $\mathfrak{M}$, we have $\mathfrak{M} \models \psi[d]$. If d is of the form ${dr'}^{\mathfrak{M}}$, for some $R' \in role^{\pm}(\mathcal{T})$, then clearly $\mathfrak{M} \models \psi[d]$, since $\mathfrak{M}_{\Sigma_0} \models \varepsilon(R)$. If d is of the form $a^{\mathfrak{M}}$, for $a \in ob(\mathcal{A})$, then it trivially holds if $\mathfrak{M}_a \not\models ER(a)$. Otherwise, $\mathfrak{M}_a \models ER(a)$, and so $inv(ER)(inv(dr)) \notin \Sigma(\mathcal{T}) \setminus \Sigma_0$. Therefore, $\mathfrak{M} \models inv(ER)(inv(dr))$ and $\mathfrak{M} \models \psi[d]$. And we clearly have $\mathfrak{M} \models R^\dagger \wedge \mathcal{A}^\dagger$. ❑

This lemma states that satisfiability of a *DL-Lite*$_{bool}^{\mathcal{R}}$ KB can be checked locally: one guesses which roles are empty and which are not (i.e., the set Σ_0) and then checks whether each object in the ABox (independently of the others) satisfies the TBox and the role emptiness constraints Σ_0. This observation suggests a high degree of parallelism in the satisfiability check:

Theorem 2. *The data complexity of the satisfiability and instance checking problems for DL-Lite*$_{bool}^{\mathcal{R}}$ *knowledge bases is in* LogSpace.

Proof. Since the instance checking problem is reducible to (un)satisfiability, we consider only the latter. The following algorithm checks whether $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is satisfiable: for every $\Sigma_0 \subseteq \Sigma(\mathcal{T})$, compute $core_{\Sigma_0}(\mathcal{T})$, check whether it is satisfiable and then, for every object name $a \in ob(\mathcal{A})$, compute $proj_{\Sigma_0}(\mathcal{K}, a)$ and check whether it is satisfiable. This deterministic algorithm requires space bounded by a logarithmic function in $|\mathcal{A}|$. Indeed, it takes $|role^{\pm}(\mathcal{T})|$ cells of the worktape to enumerate all subsets Σ_0 of $\Sigma(\mathcal{T})$—this does not depend on $|\mathcal{A}|$. In order to enumerate all objects $a \in ob(\mathcal{A})$ one needs $\log |\mathcal{A}|$ worktape cells. As $core_{\Sigma_0}(\mathcal{T})$ does not depend on $\mathcal{A}$, the time to compute it does not depend on $|\mathcal{A}|$ either. For each $a \in ob(\mathcal{A})$, the size of $proj_{\Sigma_0}(\mathcal{K}, a)$ does not depend on $|\mathcal{A}|$ as $\mathcal{A}^\flat(a)$ contains at most one occurrence of every basic concept of $\mathcal{T}$; this formula can be computed in LogSpace w.r.t. $|\mathcal{A}|$: to compute $\mathcal{A}^\flat$, $\log |\mathcal{A}|$ cells are required to enumerate all objects a_1. Finally, both formulas are in essence propositional Boolean formulas and their satisfiability can be checked *deterministically* (though in time exponential and in space linear in the length of the formula, which in our case does not depend on $|\mathcal{A}|$). ❑

In fact, this algorithm shows that satisfiability and instance checking for *DL-Lite*$_{bool}^{\mathcal{R}}$ KBs belong to the parallel complexity class AC_0 (see, e.g., [19]).

Let us see now what happens if we extend *DL-Lite*$_{bool}^{\mathcal{R}}$ and its fragments with number restrictions.

4 Satisfiability: *DL-Lite*$_{core}^{\mathcal{R},\mathcal{F}}$ is ExpTime-hard

As follows from [14, Theorem 10 and Lemma 11], satisfiability of *DL-Lite*$_{bool}^{\mathcal{R},\mathcal{N}}$ knowledge bases can be decided in ExpTime. Our aim is to show that this upper bound cannot be improved even for the seemingly rather weak language *DL-Lite*$_{core}^{\mathcal{R},\mathcal{F}}$. We need the following observation showing that in certain cases in the core and Krom languages we can actually use intersections in the left-hand side of concept inclusions, which is not strictly speaking allowed by the syntax.

Suppose that a knowledge base $\mathcal{K}$ contains a concept inclusion of the form $A_1 \sqcap A_2 \sqsubseteq C$. Define a new KB $\mathcal{K}'$ by replacing this axiom in $\mathcal{K}$ with the following set of new axioms, where $R_1, R_2, R_3, R_{12}, R_{23}$ are fresh role names:

$$A_1 \sqsubseteq \exists R_1 \qquad A_2 \sqsubseteq \exists R_2, \tag{2}$$

$$R_1 \sqsubseteq R_{12}, \qquad R_2 \sqsubseteq R_{12}, \tag{3}$$

$$\geq 2\, R_{12} \sqsubseteq \bot, \tag{4}$$

$$\exists R_1^- \sqsubseteq \exists R_3^-, \tag{5}$$

$$\exists R_3 \sqsubseteq C, \tag{6}$$

$$R_3 \sqsubseteq R_{23}, \qquad R_2 \sqsubseteq R_{23}, \tag{7}$$

$$\geq 2\, R_{23}^- \sqsubseteq \bot. \tag{8}$$

Lemma 3. (i) *If $\mathcal{I} \models \mathcal{K}'$ then $\mathcal{I} \models \mathcal{K}$, for every interpretation $\mathcal{I}$.*

(ii) *If $\mathcal{I} \models \mathcal{K}$ and $C^{\mathcal{I}} \neq \emptyset$ then there is an extension $\mathcal{I}'$ of $\mathcal{I}$ such that it agrees with $\mathcal{I}$ on every symbol of $\mathcal{K}$ and $\mathcal{I}' \models \mathcal{K}'$.*

Proof. (i) Let $\mathcal{I} \models \mathcal{K}'$ and $x \in A_1^{\mathcal{I}} \cap A_2^{\mathcal{I}}$. By (2), there is y with $(x, y) \in R_1^{\mathcal{I}}$, and so $y \in (\exists R_1^-)^{\mathcal{I}}$, and there is z with $(x, z) \in R_2^{\mathcal{I}}$. By (3), $(x, y), (x, z) \in R_{12}^{\mathcal{I}}$, whence $y = z$ by (4). By (5), $y \in (\exists R_3^-)^{\mathcal{I}}$ and then there is u with $(u, y) \in R_3^{\mathcal{I}}$ and $u \in (\exists R_3)^{\mathcal{I}}$. By (6), $u \in C^{\mathcal{I}}$ and, by (7), $(u, y) \in R_{23}^{\mathcal{I}}$, and we also have $(x, y) \in R_{23}^{\mathcal{I}}$. Finally, it follows from (8) that $u = x$; so $x \in C^{\mathcal{I}}$. Thus, $\mathcal{I} \models \mathcal{K}$.

(ii) Take some point $c \in C^{\mathcal{I}}$ and define an extension $\mathcal{I}'$ of $\mathcal{I}$ to the new role names by setting $R_1^{\mathcal{I}'} = \{(x, x) \mid x \in A_1^{\mathcal{I}}\}$, $R_2^{\mathcal{I}'} = \{(x, x) \mid x \in A_2^{\mathcal{I}}\}$, $R_3^{\mathcal{I}'} = \{(x, x) \mid x \in (A_1 \sqcap A_2)^{\mathcal{I}}\} \cup \{(c, x) \mid x \in (A_1 \sqcap \neg A_2)^{\mathcal{I}}\}$, $R_{12}^{\mathcal{I}'} = R_1^{\mathcal{I}'} \cup R_2^{\mathcal{I}'}$ and $R_{23}^{\mathcal{I}'} = R_2^{\mathcal{I}'} \cup R_3^{\mathcal{I}'}$. It is readily seen that $\mathcal{I}' \models \mathcal{K}'$. ❑

We are now in a position to prove the following:

Theorem 3. *The satisfiability problem for DL-Lite*$_{core}^{\mathcal{R},\mathcal{F}}$ *KBs is* ExpTime*-hard.*

Proof. First we show how to encode polynomial-space-bounded *alternating Turing machines* (ATMs) by means of *DL-Lite*$_{horn}^{\mathcal{R},\mathcal{F}}$ KBs. As APSpace = ExpTime, where APSpace is the class of problems accepted by polynomial-space-bounded ATMs (see, e.g., [17]), this will establish ExpTime-hardness of satisfiability for *DL-Lite*$_{horn}^{\mathcal{R},\mathcal{F}}$. And then we will use Lemma 3 to get rid of the conjunctions in the left-hand side of the concept inclusions involved in this encoding of ATMs.

Without loss of generality, we can only consider ATMs $\mathcal{M}$ with *binary* computational trees. This means that, for every non-halting state q and every symbol a from the tape alphabet, $\mathcal{M}$ has precisely two instructions of the form

$(q, a) \rightsquigarrow^0_{\mathcal{M}} (q', a', d')$ and $(q, a) \rightsquigarrow^1_{\mathcal{M}} (q'', a'', d'')$, where $d', d'' \in \{\rightarrow, \leftarrow\}$ and $\rightarrow$ (respectively, $\leftarrow$) means 'move the head right (left) one cell.' We remind the reader that each non-halting state of $\mathcal{M}$ is either an *and-state* or an *or-state*.

Given such an ATM $\mathcal{M}$, a polynomial function $p(n)$ such that any run of $\mathcal{M}$ on any input of length n uses $\leq p(n)$ tape cells, and an input word $\boldsymbol{a} = a_1, \dots, a_n$, we construct a *DL-Lite*$^{\mathcal{R},\mathcal{F}}_{horn}$ knowledge base $\mathcal{K}_{\mathcal{M},\boldsymbol{a}}$ with the following properties: (i) the size of $\mathcal{K}_{\mathcal{M},\boldsymbol{a}}$ is polynomial in the size of $\mathcal{M}$, $\boldsymbol{a}$, and (ii) $\mathcal{M}$ accepts $\boldsymbol{a}$ iff $\mathcal{K}_{\mathcal{M},\boldsymbol{a}}$ is not satisfiable. Denote by Q the set of states and by Σ the tape alphabet of $\mathcal{M}$. To encode the instructions of $\mathcal{M}$, we need the following roles:

- S_q, S^0_q, S^1_q, for each $q \in Q$: informally, $x \in \exists S^-_q$ means that x represents a configuration of $\mathcal{M}$ with the state q, and $x \in \exists S^k_q$ that the next state, according to the transition $\rightsquigarrow^k_{\mathcal{M}}$, is q, where $k = 0, 1$;
- H_i, H^0_i, H^1_i, for each $i \leq p(n)$: informally, $x \in \exists H^-_i$ means that x represents a configuration of $\mathcal{M}$ where the head scans the ith cell, and $x \in \exists H^k_i$ that, according to the transition $\rightsquigarrow^k_{\mathcal{M}}$, $k = 0, 1$, in the next configuration the head scans the ith cell;
- $C_{ia}, C^0_{ia}, C^1_{ia}$, for all $i \leq p(n)$ and $a \in \Sigma$: informally, $x \in \exists C^-_{ia}$ means that x represents a configuration of $\mathcal{M}$ where the ith cell contains a, and $x \in \exists C^k_{ia}$ that, according to $\rightsquigarrow^k_{\mathcal{M}}$, in the next configuration the ith cell contains a.

This intended meaning can be encoded using the following TBox axioms: for every instruction $(q, a) \rightsquigarrow^k_{\mathcal{M}} (q', a', \rightarrow)$ of $\mathcal{M}$ and every $i < p(n)$,

$$\exists S^-_q \sqcap \exists H^-_i \sqcap \exists C^-_{ia} \sqsubseteq \exists H^k_{i+1} \sqcap \exists S^k_{q'} \sqcap \exists C^k_{ia'}, \tag{9}$$

and for every instruction $(q, a) \rightsquigarrow^k_{\mathcal{M}} (q', a', \leftarrow)$ of $\mathcal{M}$ and every i, $1 < i \leq p(n)$,

$$\exists S^-_q \sqcap \exists H^-_i \sqcap \exists C^-_{ia} \sqsubseteq \exists H^k_{i-1} \sqcap \exists S^k_{q'} \sqcap \exists C^k_{ia'}, \tag{10}$$

To preserve the symbols on the tape that are not in the active cell, we use the following axioms, for $k = 0, 1$, $i, j \leq p(n)$ with $j \neq i$, and $a \in \Sigma$:

$$\exists H^-_j \sqcap \exists C^-_{ia} \sqsubseteq \exists C^k_{ia}. \tag{11}$$

To 'synchronise' our roles, we need two more (functional) roles T_0 and T_1 to represent the 0- and 1-successors of a configuration, and a number of role inclusions are added to the TBox: for all $k = 0, 1$, $i \leq p(n)$, $q \in Q$ and $a \in \Sigma$,

$$C^k_{ia} \sqsubseteq C_{ia}, \qquad H^k_i \sqsubseteq H_i, \qquad S^k_q \sqsubseteq S_q, \tag{12}$$

$$C^k_{ia} \sqsubseteq T_k, \qquad H^k_i \sqsubseteq T_k, \qquad S^k_q \sqsubseteq T_k, \tag{13}$$

$$\geq 2\, T_k \sqsubseteq \bot. \tag{14}$$

It remains to encode the acceptance conditions for $\mathcal{M}$ on $\boldsymbol{a}$. This can be done with the help of role names Y_0 and Y_1 and concept name A: for $q \in Q$, $k = 0, 1$,

$$\exists S^-_q \sqsubseteq A, \quad q \text{ an accepting state}, \tag{15}$$

$$Y_k \sqsubseteq T_k, \tag{16}$$

$$\geq 2\, T_k^- \sqsubseteq \bot, \tag{17}$$

$$\exists T_k^- \sqcap A \sqsubseteq \exists Y_k^-, \tag{18}$$

$$\exists S_q^- \sqcap \exists Y_k \sqsubseteq A, \quad q \text{ an or-state}, \tag{19}$$

$$\exists S_q^- \sqcap \exists Y_0 \sqcap \exists Y_1 \sqsubseteq A, \quad q \text{ an and-state}. \tag{20}$$

The TBox $\mathcal{T}$ of the *DL-Lite*$^{\mathcal{R},\mathcal{F}}_{horn}$ knowledge base $\mathcal{K}_{\mathcal{M},\boldsymbol{a}}$ we are constructing consists of axioms (9)–(20) together with the auxiliary axiom

$$A \sqcap B \sqsubseteq \bot, \tag{21}$$

where B is a fresh concept name. The ABox $\mathcal{A}$ of $\mathcal{K}_{\mathcal{M},\boldsymbol{a}}$ is comprised of the following assertions, for some object name s:

$$s\colon \exists S_{q_0}^-, \quad s\colon \exists H_1^-, \quad s\colon \exists C_{ia_i}^-, \text{ for } i \leq p(n), \qquad \text{and} \qquad s\colon B, \tag{22}$$

where q_0 is the initial state and a_i the ith symbol on the input tape, $i \leq p(n)$. Clearly, $\mathcal{K}_{\mathcal{M},\boldsymbol{a}} = (\mathcal{T}, \mathcal{A})$ is a *DL-Lite*$^{\mathcal{R},\mathcal{F}}_{horn}$ KB and its size is polynomial in the size of $\mathcal{M}$, $\boldsymbol{a}$. The proof of the following lemma is routine and left to the reader.

Lemma 4. *The ATM $\mathcal{M}$ accepts $\boldsymbol{a}$ iff the KB $\mathcal{K}_{\mathcal{M},\boldsymbol{a}}$ is not satisfiable.*

Before applying Lemma 3 in order to eliminate the conjunctions in the left-hand side of (9)–(11), (18)–(20), we check first that if $\mathcal{K}_{\mathcal{M},\boldsymbol{a}}$ is satisfiable then it is satisfiable in an interpretation $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{K}_{\mathcal{M},\boldsymbol{a}}$ and $C_2^{\mathcal{I}} \neq \emptyset$, for any C_2 occurring in an axiom of the form $C_0 \sqcap C_1 \sqsubseteq C_2$ in $\mathcal{K}$. Consider, for instance, axiom (9) and assume that $\mathcal{I} \models \mathcal{K}_{\mathcal{M},\boldsymbol{a}}$, but $(\exists S^k_{q'})^{\mathcal{I}} = \emptyset$. Then we can add to the domain of $\mathcal{I}$ two new points, say x and y, and set $(x,y) \in (S^k_{q'})^{\mathcal{I}}$, $(x,y) \in (S_{q'})^{\mathcal{I}}$, $(x,y) \in T_k^{\mathcal{I}}$. Furthermore, if q' is an accepting state, we also set $y \in A^{\mathcal{I}}$ and $(x,y) \in Y_k^{\mathcal{I}}$. One can readily check that the resulting interpretation is still a model for $\mathcal{K}_{\mathcal{M},\boldsymbol{a}}$. The remaining axioms are considered analogously.

Note that after an application of Lemma 3 we may have a conjunction in the left-hand side of $A_1 \sqsubseteq \exists R_1$. To eliminate it (using the same lemma), we observe that if $(\exists R_1)^{\mathcal{I}'} = \emptyset$, then we can always add two new points, say x and y, to the domain of $\mathcal{I}'$ and set $x \in C^{\mathcal{I}'}$, $(x,y) \in R_1^{\mathcal{I}'}$, $(x,y) \in R_{12}^{\mathcal{I}'}$, $(x,y) \in R_3^{\mathcal{I}'}$, and $(x,y) \in R_{23}^{\mathcal{I}'}$. It is readily checked that the resulting interpretation is still a model for the KB under consideration, and so we can apply Lemma 3 again. ❑

5 Instance checking with number restrictions

Theorem 4. *The instance checking problem (and query answering problem) for DL-Lite$^{\mathcal{R},\mathcal{F}}_{krom}$ is data complete for* coNP.

Proof. The coNP upper bound follows from [14, Theorem 12]. We prove the matching lower bound by reduction of the non-satisfiability problem for 2+2CNF, which is known to be coNP-complete [20]. Given a 2+2CNF

$$\varphi = \bigwedge_{k=1}^{n} (a_{k,1} \vee a_{k,2} \vee \neg a_{k,3} \vee \neg a_{k,4}),$$

where each $a_{k,j}$ is one of the propositional variables $a_1, \dots, a_m$, we construct a $DL\text{-}Lite^{\mathcal{R},\mathcal{F}}_{krom}$ KB $(\mathcal{T}, \mathcal{A}_\varphi)$ whose TBox $\mathcal{T}$ does not depend on φ and ABox $\mathcal{A}_\varphi$ is a linear encoding of φ. We will use the object names $f, c_1, \dots, c_n$ and $a_1, \dots, a_m$, role names $S, S_{\mathbf{f}}$ and $P_j, P_{j,\mathbf{t}}, P_{j,\mathbf{f}}$, for $1 \leq j \leq 4$, and concept names A, D.

Define $\mathcal{A}_\varphi$ to be the set of the following assertions, for $1 \leq k \leq n$:

$$S(f, c_k), \quad P_1(c_k, a_{k,1}), \quad P_2(c_k, a_{k,2}), \quad P_3(c_k, a_{k,3}), \quad P_4(c_k, a_{k,4})$$

and let $\mathcal{T}$ consist of the axioms

$$\geq 2\, P_j \sqsubseteq \bot, \qquad \text{for } 1 \leq j \leq 4, \tag{23}$$

$$P_{j,\mathbf{f}} \sqsubseteq P_j, \qquad P_{j,\mathbf{t}} \sqsubseteq P_j, \qquad \text{for } 1 \leq j \leq 4, \tag{24}$$

$$\neg \exists P_{j,\mathbf{t}} \sqsubseteq \exists P_{j,\mathbf{f}}, \qquad \text{for } 1 \leq j \leq 4, \tag{25}$$

$$\exists P^-_{j,\mathbf{f}} \sqsubseteq \neg A, \qquad \exists P^-_{j,\mathbf{t}} \sqsubseteq A, \qquad \text{for } 1 \leq j \leq 4, \tag{26}$$

$$\exists P_{1,\mathbf{f}} \sqcap \exists P_{2,\mathbf{f}} \sqcap \exists P_{3,\mathbf{t}} \sqcap \exists P_{4,\mathbf{t}} \sqsubseteq \exists S^-_{\mathbf{f}}\,, \tag{27}$$

$$\geq 2\, S^- \sqsubseteq \bot, \tag{28}$$

$$S_{\mathbf{f}} \sqsubseteq S, \tag{29}$$

$$\exists S_{\mathbf{f}} \sqsubseteq D. \tag{30}$$

It should be clear that $(\mathcal{T}, \mathcal{A}_\varphi)$ is LogSpace computable (in $|\varphi|$). Note, however, that axiom (27) does not belong to $DL\text{-}Lite^{\mathcal{R},\mathcal{F}}_{krom}$ because of the conjunctions in its left-hand side. However, they can be eliminated with the help of Lemma 3. So let us prove that $(\mathcal{T}, \mathcal{A}_\varphi) \models D(f)$ iff φ is not satisfiable.

Suppose first that φ is satisfiable. Then there is an assignment $\mathfrak{a}$ of the truth-values $\mathbf{t}$ and $\mathbf{f}$ to propositional variables such that $\mathfrak{a}(a_{k,1}) = \mathbf{t}$ or $\mathfrak{a}(a_{k,2}) = \mathbf{t}$ or $\mathfrak{a}(a_{k,3}) = \mathbf{f}$ or $\mathfrak{a}(a_{k,4}) = \mathbf{f}$, for all $k \in \{1, \dots, n\}$. Consider the interpretation $\mathcal{I}$ with $\Delta^{\mathcal{I}} = \{x_1, \dots, x_m, y_1, \dots, y_n, z\}$ and

- $f^{\mathcal{I}} = z, \quad c_k^{\mathcal{I}} = y_k$, for $1 \leq k \leq n, \quad a_i^{\mathcal{I}} = x_i$, for $1 \leq i \leq m$,
- $A^{\mathcal{I}} = \{x_i \mid \mathfrak{a}(a_i) = \mathbf{t}\} \cup \{y_k \mid 1 \leq k \leq n\} \cup \{z\}$,
- $P^{\mathcal{I}}_{j,\mathbf{t}} = \{(y_k, a^{\mathcal{I}}_{k,j}) \mid 1 \leq k \leq n,\ \mathfrak{a}(a_{k,j}) = \mathbf{t}\} \cup \{(x_i, x_i) \mid \mathfrak{a}(a_i) = \mathbf{t}\} \cup \{(z, z)\}$,
  $P^{\mathcal{I}}_{j,\mathbf{f}} = \{(y_k, a^{\mathcal{I}}_{k,j}) \mid 1 \leq k \leq n,\ \mathfrak{a}(a_{k,j}) = \mathbf{f}\} \cup \{(x_i, x_i) \mid \mathfrak{a}(a_i) = \mathbf{f}\}$,
  $P^{\mathcal{I}}_j = P^{\mathcal{I}}_{j,\mathbf{t}} \cup P^{\mathcal{I}}_{j,\mathbf{f}}$, for $1 \leq j \leq 4$,
- $S^{\mathcal{I}} = \{(z, y_k) \mid 1 \leq k \leq n\}$,
  $S^{\mathcal{I}}_{\mathbf{f}} = \{(z, y_k) \mid 1 \leq k \leq n, \mathfrak{a}(a_{k,1} \vee a_{k,2} \vee \neg a_{k,3} \vee \neg a_{k,4}) = \mathbf{f}\} = \emptyset$,
- $D^{\mathcal{I}} = \{z \mid \mathfrak{a}(\varphi) = \mathbf{f}\} = \emptyset$.

It is not hard to check that $\mathcal{I} \models (\mathcal{T}, \mathcal{A}_\varphi)$, and clearly $\mathcal{I} \not\models D(f)$.

Assume now that φ is not satisfiable and $\mathcal{I} \models (\mathcal{T}, \mathcal{A}_\varphi)$. Define an assignment $\mathfrak{a}$ by taking $\mathfrak{a}(a_i) = \mathbf{t}$ iff $a_i^{\mathcal{I}} \in A^{\mathcal{I}}$. As φ is not satisfiable, there is k, $1 \leq k \leq n$, such that $\mathfrak{a}(a_{k,1}) = \mathfrak{a}(a_{k,2}) = \mathbf{f}$, $\mathfrak{a}(a_{k,3}) = \mathfrak{a}(a_{k,4}) = \mathbf{t}$. In view of (25), for each j, $1 \leq j \leq 4$, we have $c_k^{\mathcal{I}} \in (\exists P_{j,\mathbf{t}})^{\mathcal{I}} \cup (\exists P_{j,\mathbf{f}})^{\mathcal{I}}$, and by (24), $c_k^{\mathcal{I}} \in (\exists P_j)^{\mathcal{I}}$. Therefore, by (23) and (26), $c_k^{\mathcal{I}} \in (\exists P_{j,\mathbf{t}})^{\mathcal{I}}$ if $\mathfrak{a}(a_{k,j}) = \mathbf{t}$ and $c_k^{\mathcal{I}} \in (\exists P_{j,\mathbf{f}})^{\mathcal{I}}$ if $\mathfrak{a}(a_{k,j}) = \mathbf{f}$, and hence, by (27), $c_k^{\mathcal{I}} \in (\exists S^-_{\mathbf{f}})^{\mathcal{I}}$. Then by (28) and (29), we have $f^{\mathcal{I}} \in (\exists S_{\mathbf{f}})^{\mathcal{I}}$, from which, by (30), $f^{\mathcal{I}} \in D^{\mathcal{I}}$. It follows that $(\mathcal{T}, \mathcal{A}_\varphi) \models D(f)$. ❑

If the functionality constraints are relaxed just a bit to allow for axioms of the form $\geq 2\,R \sqsubseteq A$ then the same complexity result holds for the core fragment:

Theorem 5. *The instance checking problem* (*and query answering problem*) *for* $\textit{DL-Lite}^{\mathcal{R},\mathcal{N}}_{core}$ *is data complete for* coNP.

Proof. The coNP upper bound again follows from [14, Theorem 12], and the matching lower bound is proved by reduction of the non-satisfiability problem for 2+2CNF. The main difference from the previous proof is that $\textit{DL-Lite}^{\mathcal{R},\mathcal{N}}_{core}$, unlike $\textit{DL-Lite}^{\mathcal{R},\mathcal{F}}_{krom}$, cannot express 'covering conditions' like (25). It turns out, however, that we can use number restrictions to represent this kind of constraints. Given a 2+2CNF φ, we take the ABox $\mathcal{A}_\varphi$ constructed in the proof of Theorem 4 (and computable in LogSpace in $|\varphi|$). The (φ independent) $\textit{DL-Lite}^{\mathcal{R},\mathcal{N}}_{core}$ TBox $\mathcal{T}$, describing the meaning of any such representation of 2+2CNF ψ in terms of $\mathcal{A}_\psi$, is also defined in the same way as in that proof except that axiom (25) is now replaced by the following set of axioms:

$$T_{j,1} \sqsubseteq T_j, \qquad T_{j,2} \sqsubseteq T_j, \qquad T_{j,3} \sqsubseteq T_j, \tag{31}$$

$$\geq 2\,T_j^- \sqsubseteq \bot, \tag{32}$$

$$\exists P_j \sqsubseteq \exists T_{j,1}, \qquad \exists P_j \sqsubseteq \exists T_{j,2}, \tag{33}$$

$$\exists T_{j,1}^- \sqcap \exists T_{j,2}^- \sqsubseteq \exists T_{j,3}^-, \tag{34}$$

$$\geq 2\,T_j \sqsubseteq \exists P_{j,\mathbf{t}} \qquad \exists T_{j,3} \sqsubseteq \exists P_{j,\mathbf{f}}, \tag{35}$$

where $T_j, T_{j,1}, T_{j,2}, T_{j,3}$ are fresh role names, for $1 \leq j \leq 4$. It should be clear that $(\mathcal{T}, \mathcal{A}_\varphi)$ is LogSpace computable (in $|\varphi|$). The conjunctions in the left-hand side of (27) and (34) can be eliminated by using Lemma 3. So it remains to prove that $(\mathcal{T}, \mathcal{A}_\varphi) \models D(f)$ iff φ is not satisfiable.

Suppose first that φ is satisfiable. Then there is an assignment $\mathfrak{a}$ of the truth-values $\mathbf{t}$ and $\mathbf{f}$ to propositional variables such that $\mathfrak{a}(a_{k,1}) = \mathbf{t}$ or $\mathfrak{a}(a_{k,2}) = \mathbf{t}$ or $\mathfrak{a}(a_{k,3}) = \mathbf{f}$ or $\mathfrak{a}(a_{k,4}) = \mathbf{f}$, for all k, $1 \leq k \leq n$. Consider the interpretation $\mathcal{I}$ with $\Delta^{\mathcal{I}} = \{x_1, \dots, x_m, z\} \cup \{y_k, u_{k,j,1}, u_{k,j,2} \mid 1 \leq j \leq 4, 1 \leq k \leq n\}$ and

- $f^{\mathcal{I}} = z$, $c_k^{\mathcal{I}} = y_k$, for $1 \leq k \leq n$, $a_i^{\mathcal{I}} = x_i$, for $1 \leq i \leq m$,
- $A^{\mathcal{I}} = \{x_i \mid 1 \leq i \leq m, \mathfrak{a}(a_i) = \mathbf{t}\}$,
- $P_{j,\mathbf{t}}^{\mathcal{I}} = \{(y_k, a_{k,j}^{\mathcal{I}}) \mid 1 \leq k \leq n,\ \mathfrak{a}(a_{k,j}) = \mathbf{t}\}$, for $1 \leq j \leq 4$,
- $P_{j,\mathbf{f}}^{\mathcal{I}} = \{(y_k, a_{k,j}^{\mathcal{I}}) \mid 1 \leq k \leq n,\ \mathfrak{a}(a_{k,j}) = \mathbf{f}\}$, for $1 \leq j \leq 4$,
- $P_j^{\mathcal{I}} = P_{j,\mathbf{t}}^{\mathcal{I}} \cup P_{j,\mathbf{f}}^{\mathcal{I}}$, for $1 \leq j \leq 4$,
- $T_{j,1}^{\mathcal{I}} = \{(y_k, u_{k,j,1}) \mid 1 \leq k \leq n\}$, for $1 \leq j \leq 4$,
- $T_{j,2}^{\mathcal{I}} = \{(y_k, u_{k,j,2}) \mid 1 \leq k \leq n,\ \mathfrak{a}(a_{k,j}) = \mathbf{t}\} \cup$ $\{(y_k, u_{k,j,1}) \mid 1 \leq k \leq n,\ \mathfrak{a}(a_{k,j}) = \mathbf{f}\}$, for $1 \leq j \leq 4$,
- $T_{j,3}^{\mathcal{I}} = \{(y_i, u_{k,j,1}) \mid 1 \leq k \leq n,\ \mathfrak{a}(a_{k,j}) = \mathbf{f}\}$, for $1 \leq j \leq 4$,
- $T_j^{\mathcal{I}} = T_{j,1}^{\mathcal{I}} \cup T_{j,2}^{\mathcal{I}}$, for $1 \leq j \leq 4$,
- $S_{\mathbf{f}}^{\mathcal{I}}$, $S^{\mathcal{I}}$ and $D^{\mathcal{I}}$ are defined in the same way as in the proof of Theorem 4.

It is not hard to check that $\mathcal{I} \models (\mathcal{T}, \mathcal{A}_\varphi)$, and clearly $\mathcal{I} \not\models D(f)$.

Assume now that φ is not satisfiable and $\mathcal{I} \models (\mathcal{T}, \mathcal{A}_\varphi)$. Define an assignment $\mathfrak{a}$ by taking $\mathfrak{a}(a_i) = \mathbf{t}$ iff $a_i^{\mathcal{I}} \in A^{\mathcal{I}}$. As φ is not satisfiable, there is k, $1 \leq k \leq n$, such that $\mathfrak{a}(a_{k,1}) = \mathfrak{a}(a_{k,2}) = \mathbf{f}$, $\mathfrak{a}(a_{k,3}) = \mathfrak{a}(a_{k,4}) = \mathbf{t}$.

For each j, $1 \leq j \leq 4$, we have $c_k^{\mathcal{I}} \in (\exists P_j)^{\mathcal{I}}$; by (33), $c_k^{\mathcal{I}} \in (\exists T_{j,1})^{\mathcal{I}}, (\exists T_{j,2})^{\mathcal{I}}$. So there are v_1, v_2 such that $(c_k^{\mathcal{I}}, v_1) \in T_{j,1}^{\mathcal{I}}$ and $(c_k^{\mathcal{I}}, v_2) \in T_{j,2}^{\mathcal{I}}$. If $v_1 \neq v_2$ then $c_k^{\mathcal{I}} \in (\geq 2\, T_j)^{\mathcal{I}}$ and, by (35), $c_k^{\mathcal{I}} \in (P_{j,\mathbf{t}})^{\mathcal{I}}$. Otherwise, if $v_1 = v_2 = v$, we have by (34), $v \in (\exists T_{j,3}^-)^{\mathcal{I}}$, and so by (31) and (32), $c_k^{\mathcal{I}} \in (\exists T_{j,3})^{\mathcal{I}}$, from which, by (35), $c_k^{\mathcal{I}} \in (P_{j,\mathbf{f}})^{\mathcal{I}}$. Therefore, $c_k^{\mathcal{I}} \in (\exists P_{j,\mathbf{t}})^{\mathcal{I}} \cup (\exists P_{j,\mathbf{f}})^{\mathcal{I}}$, and by (24), $c_k^{\mathcal{I}} \in (\exists P_j)^{\mathcal{I}}$. Thus, by (23) and (26), $c_k^{\mathcal{I}} \in (\exists P_{j,\mathbf{t}})^{\mathcal{I}}$ if $\mathfrak{a}(a_{k,j}) = \mathbf{t}$ and $c_k^{\mathcal{I}} \in (\exists P_{j,\mathbf{f}})^{\mathcal{I}}$ if $\mathfrak{a}(a_{k,j}) = \mathbf{f}$, and hence, by (27), $c_k^{\mathcal{I}} \in (\exists S_{\mathbf{f}}^-)^{\mathcal{I}}$. Then by (28) and (29), we have $f^{\mathcal{I}} \in (\exists S_{\mathbf{f}})^{\mathcal{I}}$, from which, by (30), $f^{\mathcal{I}} \in D^{\mathcal{I}}$. It follows that $(\mathcal{T}, \mathcal{A}_\varphi) \models D(f)$. ❑

However, the core fragment with only functionality constraints is data complete for P (the lower bound would follow from [8, Theorem 6, item 2] but unfortunately the proof in [8] is fallacious).

Theorem 6. *The instance checking problem (and query answering problem) for DL-Lite$_{core}^{\mathcal{R},\mathcal{F}}$ is data complete for* P.

Proof. The polynomial upper bound follows from [12]. We prove the matching lower bound by reduction of the entailment problem for Horn-CNF, which is known to be P-complete (see, e.g., [4, Exercise 2.2.4]). Given a Horn-CNF

$$\varphi = \bigwedge_{k=1}^{n} (\neg a_{k,1} \vee \neg a_{k,2} \vee a_{k,3}) \quad \wedge \quad \bigwedge_{l=1}^{p} a_{l,0},$$

where each $a_{k,j}$ and each $a_{l,0}$ is one of the propositional variables $a_1, \dots, a_m$, we construct a *DL-Lite*$_{core}^{\mathcal{R},\mathcal{F}}$ knowledge base $(\mathcal{T}, \mathcal{A}_\varphi)$ whose TBox $\mathcal{T}$ does not depend on φ and ABox $\mathcal{A}_\varphi$ is computed in LogSpace from φ. We will need the object names $c_1, \dots, c_n$ and $v_{k,j,i}$, for $1 \leq k \leq n$, $1 \leq j \leq 3$, $1 \leq i \leq m$ (for each variable, we take one object name for each possible occurrence of this variable in each non-unit clause), role names $S, S_{\mathbf{t}}$ and $P_j, P_{j,\mathbf{t}}$, for $1 \leq j \leq 3$, and a concept name A. Define $\mathcal{A}_\varphi$ to be the set containing the assertions:

$$S(v_{1,1,i}, v_{1,2,i}), S(v_{1,2,i}, v_{1,3,i}), S(v_{1,3,i}, v_{2,1,i}), S(v_{2,1,i}, v_{2,2,i}), S(v_{2,2,i}, v_{2,3,i}), \dots$$
$$\dots, S(v_{n,2,i}, v_{n,3,i}), S(v_{n,3,i}, v_{1,1,i}), \text{ for } 1 \leq i \leq m,$$
$$P_j(v_{k,j,i}, c_k) \quad \text{iff} \quad a_{k,j} = a_i, \qquad \text{for } 1 \leq i \leq m, \ \ 1 \leq k \leq n, \ \ 1 \leq j \leq 3,$$
$$A(v_{1,1,i}) \quad \text{iff} \quad a_{l,0} = a_i, \qquad \text{for } 1 \leq i \leq m, \ \ 1 \leq l \leq p$$

(all objects for each variable are organised in an S-cycle and $P_j(v_{k,j,i}, c_k) \in \mathcal{A}_\varphi$ iff the variable a_i occurs in the kth non-unit clause of φ in the jth position). And let $\mathcal{T}$ consist of the following concept and role inclusions:

$$S_{\mathbf{t}} \sqsubseteq S, \qquad \geq 2\, S \sqsubseteq \bot, \quad A \sqsubseteq \exists S_{\mathbf{t}}, \quad \exists S_{\mathbf{t}}^- \sqsubseteq A, \tag{36}$$
$$P_{j,\mathbf{t}} \sqsubseteq P_j, \qquad \geq 2\, P_j \sqsubseteq \bot, \quad A \sqsubseteq \exists P_{j,\mathbf{t}}, \quad \text{for } 1 \leq j \leq 2, \tag{37}$$
$$P_{3,\mathbf{t}} \sqsubseteq P_3, \tag{38}$$
$$\geq 2\, P_3^- \sqsubseteq \bot, \tag{39}$$

$$\exists P^-_{1,\mathbf{t}} \sqcap \exists P^-_{2,\mathbf{t}} \sqsubseteq \exists P^-_{3,\mathbf{t}}, \quad (40)$$

$$\exists P_{3,\mathbf{t}} \sqsubseteq A. \quad (41)$$

It should be clear that $(\mathcal{T}, \mathcal{A}_\varphi)$ is LogSpace computable (in $|\varphi|$). As in the previous proofs, here we have an axiom, namely (40), that does not belong to *DL-Lite*$^{\mathcal{R},\mathcal{F}}_{core}$ because of the conjunctions in its left-hand side. As before, this conjunction is eliminated with the help of Lemma 3. Our aim is to show that $(\mathcal{T}, \mathcal{A}_\varphi) \models A(v_{1,1,i_0})$ iff $\varphi \models a_{i_0}$.

Suppose $\varphi \not\models a_{i_0}$. Then there is an assignment $\mathfrak{a}$ with $\mathfrak{a}(\varphi) = \mathbf{t}$ and $\mathfrak{a}(a_{i_0}) = \mathbf{f}$. We construct a model $\mathcal{I}$ for $(\mathcal{T}, \mathcal{A}_\varphi)$ such that $\mathcal{I} \not\models A(v_{1,1,i_0})$. Define $\mathcal{I}$ by taking $\Delta^\mathcal{I} = \{x_{k,j,i}, z_{k,j,i} \mid 1 \le k \le n, 1 \le j \le 3, 1 \le i \le m\} \cup \{y_k \mid 1 \le k \le n\}$, $c^\mathcal{I}_k = y_k$, for $1 \le k \le n$, $v^\mathcal{I}_{k,j,i} = x_{k,j,i}$, for $1 \le k \le n$, $\le j \le 3$, $1 \le i \le m$. The extensions of the concept and role names are defined as in Fig. 1. It is routine to check that we indeed have $\mathcal{I} \models (\mathcal{T}, \mathcal{A}_\varphi)$ and $\mathcal{I} \not\models A(v_{1,1,i_0})$.

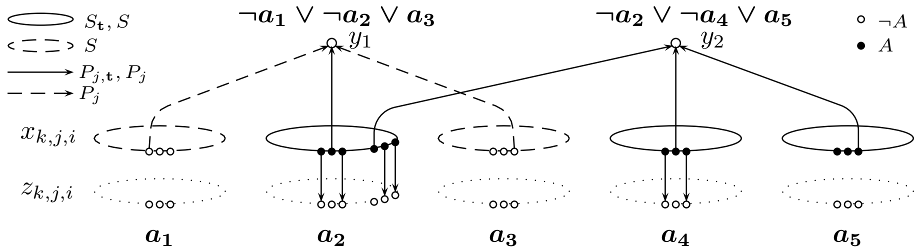

Fig. 1. The model $\mathcal{I}$ satisfying $(\mathcal{T}, \mathcal{A}_\varphi)$, for $\varphi = (\neg a_1 \vee \neg a_2 \vee a_3) \wedge (\neg a_2 \vee \neg a_4 \vee a_5)$.

Conversely, assume now that $\varphi \models a_{i_0}$. Consider some $\mathcal{I} \models (\mathcal{T}, \mathcal{A}_\varphi)$ and define $\mathfrak{a}$ to be the assignment such that $\mathfrak{a}(a_i) = \mathbf{t}$ iff $v^\mathcal{I}_{1,1,i} \in A^\mathcal{I}$, for $1 \le i \le m$. By (36), for each i, $1 \le i \le m$, we have either $v^\mathcal{I}_{k,j,i} \in A^\mathcal{I}$, for *all* k, j with $1 \le k \le n$, $1 \le j \le 3$, or $v^\mathcal{I}_{k,j,i} \notin A^\mathcal{I}$, for *all* k, j with $1 \le k \le n$, $1 \le j \le 3$.

Now, if we have $\mathfrak{a}(a_{k,1}) = \mathbf{t}$ and $\mathfrak{a}(a_{k,2}) = \mathbf{t}$, for $1 \le k \le n$ then, by (37), $c^\mathcal{I}_k \in (\exists P^-_{1,\mathbf{t}})^\mathcal{I}, (\exists P^-_{2,\mathbf{t}})^\mathcal{I}$. By (40), $c^\mathcal{I}_k \in (\exists P^-_{3,\mathbf{t}})^\mathcal{I}$ and hence, by (39) and (38), $v^\mathcal{I}_{k,3,i} \in (\exists P_{3,\mathbf{t}})^\mathcal{I}$, where $a_{k,3} = a_i$, which means, by (41), that $v^\mathcal{I}_{k,3,i} \in A^\mathcal{I}$, and so $v^\mathcal{I}_{1,1,i} \in A^\mathcal{I}$ and $\mathfrak{a}(a_i) = \mathbf{t}$. It follows that $\mathfrak{a}(\varphi) = \mathbf{t}$, and hence $\mathfrak{a}(a_{i_0}) = \mathbf{t}$, which, by definition, means that $v^\mathcal{I}_{1,1,i_0} \in A^\mathcal{I}$. As $\mathcal{I}$ was an arbitrary model of $(\mathcal{T}, \mathcal{A}_\varphi)$, we can conclude that $(\mathcal{T}, \mathcal{A}_\varphi) \models A(v_{1,1,i_0})$. ❑

6 Conclusion

The results obtained in this paper and [2] show the following: (1) One can add either number restrictions or role inclusions to the basic (*core*, *horn*, *krom* and *bool*) *DL-Lite* logics without changing their complexity. (2) However, taken together, these constructs spoil the nice computational properties of the basic *DL-Lite* logics. (3) If both of them are really needed for an application, one should try and restrict their interaction (e.g., by avoiding axioms of the form $R \sqsubseteq P$ with functional role P, as suggested in [9]). Exploring in depth this interaction, as well as the impact of other constructs (transitive roles, Booleans on roles, etc.) on the computational properties of *DL-Lite* logics is an interesting and practically important area for further research.

References

1. A. Artale, D. Calvanese, R. Kontchakov, V. Ryzhikov, and M. Zakharyaschev. Complexity of reasoning over Entity-Relationship models. In *Proc. of DL*, pages 163–170, 2007.
2. A. Artale, D. Calvanese, R. Kontchakov, and M. Zakharyaschev. DL-Lite in the light of first-order logic. In *Proc. of AAAI*, pages 361–366, 2007.
3. P. Bernstein, F. Giunchiglia, A. Kementsietsidis, J. Mylopoulos, L. Serafini, and I. Zaihrayeu. Data management for peer-to-peer computing: A vision. In *Proc. of WebDB*, 2002.
4. E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem.* Perspectives in Mathematical Logic. Springer, 1997.
5. A. Borgida, R. Brachman, D. McGuinness, and L. Alperin Resnick. CLASSIC: A structural data model for objects. In *Proc. of ACM SIGMOD*, pages 59–67, 1989.
6. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. DL-Lite: Tractable description logics for ontologies. In *Proc. of AAAI*, pages 602–607, 2005.
7. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tailoring OWL for data intensive ontologies. In *Proc. of OWLED*, 2005.
8. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. In *Proc. of KR*, pages 260–270, 2006.
9. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi and R. Rosati. Linking data to ontologies: the description logic *DL-Lite*$_A$. In *Proc. OWLED*, 2006.
10. D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Logical foundations of peer-to-peer data integration. In *Proc. of PODS*, pages 241–251, 2004.
11. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *J. of Automated Reasoning*, 39:385–429, 2007.
12. T. Eiter, G. Gottlob, M. Ortiz, and M. Šimkus. Query answering in the description logic Horn-$\mathcal{SHIQ}$. In *Proc. of JELIA*, 2008.
13. E. Franconi, G. Kuper, A. Lopatenko, and I. Zaihrayeu. Queries and updates in the coDB peer-to-peer database system. In *Proc. of VLDB*, pages 1277–1280, 2004.
14. B. Glimm, I. Horrocks, C. Lutz, and U. Sattler. Conjunctive query answering for the description logic $\mathcal{SHIQ}$. In *Proc. of IJCAI*, pages 399–404, 2007.
15. J. Heflin and J. Hendler. A portrait of the Semantic Web in action. *IEEE Intelligent Systems*, 16:54–59, 2001.
16. U. Hustadt, B. Motik, and U. Sattler. Reasoning in description logics by a reduction to disjunctive Datalog. *J. of Automated Reasoning*, 39:351–384, 2007.
17. D. Kozen. *Theory of Computation.* Springer, 2006.
18. M. Lenzerini. Data integration: A theoretical perspective. In *Proc. of PODS*, pages 233–246, 2002.
19. Ch. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.
20. A. Schaerf. On the complexity of the instance checking problem in concept languages with existential quantification. *J. Intell. Infor. Systems*, 2:265–278, 1993.
21. K. Schild. A correspondence theory for terminological logics: Preliminary report. In *Proc. of IJCAI*, pages 466–471. Morgan Kaufmann, 1991.