Foundations of Instance Level Updates in Expressive Description Logics

Hongkai Liu[a], Carsten Lutz[b], Maja Miličić[c], Frank Wolter[d]

[a] *Institut für Theoretische Informatik*
TU Dresden, Germany
[b] *Fachbereich Informatik*
Universität Bremen, Germany
[c] *School of Computer Science*
The University of Manchester, UK
[d] *Department of Computer Science*
University of Liverpool, UK

Abstract

In description logic (DL), ABoxes are used for describing the state of affairs in an application domain. We consider the problem of updating ABoxes when the state changes, assuming that update information is described at an atomic level, i.e., in terms of possibly negated ABox assertions that involve only atomic concepts and roles. We analyze such basic ABox updates in several standard DLs, in particular addressing questions of expressibility and succinctness: can updated ABoxes always be expressed in the DL in which the original ABox was formulated and, if so, what is the size of the updated ABox? It turns out that DLs have to include nominals and the '@' constructor of hybrid logic for updated ABoxes to be expressible, and that this still holds when updated ABoxes are approximated. Moreover, the size of updated ABoxes is exponential in the role depth of the original ABox and the size of the update. We also show that this situation improves when updated ABoxes are allowed to contain additional auxiliary symbols. Then, DLs only need to include nominals for updated ABoxes to exist, and the size of updated ABoxes is polynomial in the size of both the original ABox and the update.

Keywords: Description Logics, ABoxes, Updates

1. Introduction

Description Logics (DLs) are a traditional family of knowledge representation formalisms which, in recent years, have played an important role as a logical underpinning of ontology languages such as the W3C recommendation OWL [1].

Email addresses: `liuhkai@tcs.inf.tu-dresden.de` (Hongkai Liu), `clu@uni-bremen.de` (Carsten Lutz), `maja.milicic@manchester.ac.uk` (Maja Miličić), `frank@csc.liv.ac.uk` (Frank Wolter)

In DLs, a knowledge base (KB) typically consists of two parts: a *TBox* to store intensional knowledge, i.e., a general formalization of the relevant concepts and relationships of the application domain; and an *ABox* to store extensional knowledge, i.e., instance level assertions that describe the current state of affairs in the application. Just like database systems, DL knowledge bases are not static entities, but have to be modified when the application domain evolves. This raises the fundamental *update problem*, which consists of rewriting the knowledge base to incorporate new information from the application without unnecessarily losing any existing knowledge. In the case of a DL knowledge base, at least three different incarnations of the update problem can be distinguished:

- *TBox updates*, triggered by changes of the intensional knowledge of the application domain;
- *ABox updates*, which have to be carried out when the intensional knowledge remains stable, but the state of affairs in the application changes;
- *KB updates*, which do not distinguish between the two levels of knowledge and allow simultaneous modification of the TBox and the ABox.

In typical applications, instance level knowledge stored in the ABox tends to change frequently whereas intensional knowledge in the TBox often remains stable for longer periods of time. Moreover, automatic TBox modifications are rarely desired because the TBox is typically the result of a careful and time-consuming manual modeling process, and thus its syntactic structure should not be changed in a radical way.

These observations lead us to study *ABox updates* as a fundamental and basic form of updates in a DL context. A central property of DL ABoxes is that they store *incomplete* knowledge, reflected by an open world semantics and the use of compound logical expressions that can involve disjunction and existential quantification. It follows that, technically, updating DL ABoxes is equivalent to updating logical theories, a problem with a long tradition in both the database and AI communities [4, 5, 6, 7, 8].

In the database and AI literature, for a long time no proper distinction was made between *updates* as studied in this paper and the related notion of a *revision*. While the purpose of update is to bring the knowlegde base up to date when the world described by it changes, revision aims at incorporating new knowledge that was obtained about a static world. Katsuno and Mendelzon [6] discuss this distinction in detail, show that update and revision are fundamentally different operations, and give 8 postulates that any rational update operator should satisfy. The protoypical update semantics that complies with these postulates is Winslett's well-known PMA semantics [4] whose general idea can, in our context, be stated as follows. The models of the original knowledge base $\mathcal{K}$ are viewed as those states of the world that are considered possible; when $\mathcal{K}$ is to be updated with new information $\mathcal{U}$, then the models of the resulting updated knowledge base $\mathcal{K}'$ should satisfy $\mathcal{U}$, but also be 'as close as possible' to the models of $\mathcal{K}$ (the *principle of minimal change*). In the case of updating propositional theories and logical databases as considered in [6, 4], the difficulty of defining what 'as close as possible' means mainly derives from the following two features: (i) the newly added information may be non-deterministic, e.g. when it involves disjunction; and (ii) the updated theory must satisfy additional domain constraints stated in the form of a logical background theory. As discussed in more detail in Section 6 of this paper, the combination of these features with the first-order quantification present in description logics leads to serious semantic difficulties and also to computational problems. For this reason, we concentrate on a simple, yet fundamental form of update where (i) the newly added information $\mathcal{U}$ consists of a set of ground literals, i.e., sets of ABox assertions $A(a)$ or $r(a,b)$ and their negations, where A is a concept name and r a role name (thus both are atomic); and (ii) no background theory is present, i.e., the knowledge base $\mathcal{K}$ comprises *only* an ABox, but no TBox. In this case, there seems to be only one sensible formalization of 'as close as possible': the models of $\mathcal{K}'$ are obtained from the models of $\mathcal{K}$ by (deterministically!) applying the changes dictated by the ground literals in $\mathcal{U}$. This semantics, which we adopt in the current paper, can thus be viewed as an incarnation of Winslett's semantic that avoids the potentially controversial cases.

As a starting point for the current paper, we observe that, in standard 'expressive' DLs such as those between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$, we can find an ABox $\mathcal{A}$ and update $\mathcal{U}$ of the restricted form described above such that the result of updating $\mathcal{A}$ with $\mathcal{U}$ *cannot be expressed* in the given DL. As a concrete example, take the following ABox $\mathcal{A}$, which is formulated in $\mathcal{ALC}$, the basic expressive DL with Boolean operators. It states that John is a parent with only happy children, that Peter is his child, and that Mary is a person:

$$\begin{array}{l}\mathsf{john{:}Person} \sqcap \exists \mathsf{has\text{-}child.Person} \sqcap \forall \mathsf{has\text{-}child.(Person} \sqcap \mathsf{Happy)} \\ \mathsf{has\text{-}child(john, peter)} \\ \mathsf{mary{:}Person}\end{array}$$

Suppose now that the situation changes by Mary becoming unhappy. The result of updating $\mathcal{A}$ with $\mathcal{U} = \{\mathsf{Mary} : \neg\mathsf{Happy}\}$ can be represented by the following ABox $\mathcal{A}'$, which is formulated in $\mathcal{ALCO}$, the extension of $\mathcal{ALC}$ with nominals (individual names inside concept descriptions):

$$\begin{array}{l}\mathsf{john{:}Person} \sqcap \exists \mathsf{has\text{-}child.Person} \sqcap \forall \mathsf{has\text{-}child.(Person} \sqcap (\mathsf{Happy} \sqcup \{\mathsf{mary}\})) \\ \mathsf{has\text{-}child(john, peter)} \\ \mathsf{mary{:}Person} \sqcap \neg\mathsf{Happy}\end{array}$$

To understand why $\mathcal{A}'$ is appropriate, note that $\mathcal{A}$ provides no information about whether or not Mary is a child of John. Because we cannot exclude that this is the case, John may now have an unhappy child, which is Mary. Thus, the new knowledge concerning Mary also resulted in an update of the knowledge concerning John. Using the nominal $\{\mathsf{mary}\}$ in the assertion for john is actually unavoidable as it can be shown that there is no $\mathcal{ALC}$-ABox that is equivalent to the $\mathcal{ALCO}$-ABox $\mathcal{A}'$. As a consequence, the update of the $\mathcal{ALC}$-ABox $\mathcal{A}$ with $\mathcal{U}$ cannot be expressed in $\mathcal{ALC}$.

We say that a description logic $\mathcal{L}$ does not *have updates* if there are an $\mathcal{L}$-ABox $\mathcal{A}$ and update $\mathcal{U}$ such that the result of updating $\mathcal{A}$ with $\mathcal{U}$ cannot be expressed in $\mathcal{L}$. The first main aim of this paper is to *understand how the problem of non-expressibility of updated ABoxes can be overcome.* In particular, we consider two options: (i) increasing the expressive power of DLs by adding additional constructors and (ii) relaxing the definition of updated ABoxes.

Regarding (i), we show that the addition of nominals (as in the example above) and the '@' constructor from hybrid logic suffices to ensure the existence of updated ABoxes in all DLs between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$. Intuitively, the '@' constructor enables 'jumps' between individuals by allowing the formation of concepts such as $@_a C$ which is satisfied at any point of an interpretation whenever the individual a satisfies the concept C. We also show that the '@'-constructor (but not nominals) can be replaced by Boolean ABoxes, i.e., ABoxes that admit Boolean operators to be applied to ABox assertions.

Regarding (ii), we consider the following definitions of updated ABoxes. An ABox $\mathcal{A}'$ is

- a *semantic update* of $\mathcal{A}$ with $\mathcal{U}$ if the models of $\mathcal{A}'$ are precisely those interpretations that can be obtained from models of $\mathcal{A}$ by making the assertions in $\mathcal{U}$ true (the standard definition);
- an *approximate update* of $\mathcal{A}$ with $\mathcal{U}$ regarding a DL $\mathcal{L}$ if $\mathcal{A}'$ entails exactly the same $\mathcal{L}$-ABox assertions as the semantic update of $\mathcal{A}$ with $\mathcal{U}$;
- a *projective update* of $\mathcal{A}$ with $\mathcal{U}$ if the models of $\mathcal{A}'$ are precisely the models of the semantic update, after projecting both to the symbols in $\mathcal{A}$ and $\mathcal{U}$;
- a *projective approximate update* of $\mathcal{A}$ with $\mathcal{U}$ if $\mathcal{A}'$ entails exactly the same $\mathcal{L}$-ABox assertions φ as the semantic update of $\mathcal{A}$ with $\mathcal{U}$ as long as φ uses only symbols from $\mathcal{A}$ and $\mathcal{U}$.[1]

Observe that projective updates allow the use of fresh, auxiliary symbols in $\mathcal{A}'$, and so do projective approximate updates. Also note that (projective and non-projective) approximate updates have an additional parameter, which is the DL $\mathcal{L}$ in which entailed ABox assertions are formulated. It is not hard to see that every semantic update is also an approximate update and a projective update, which in turn are also projective approximate updates. Moreover, semantic updates and approximate updates can be proved to be logically equivalent, if the former exists. Due to the new symbols, this is in general not true for projective updates. Similar forms of updates have been considered e.g. in [9, 11], see Section 7 for more details.

Unfortunately, it turns out that the more relaxed definitions of updated ABoxes only rarely help to overcome the problem of DLs not having updates.

[1] Actually, the 'real' definitions of projective updates and projective approximate updates are slightly stronger and use different (but more complicated to describe) sets of symbols. We refer to Definition 4 for full details.

More precisely, all DLs considered in this paper have approximate updates if, and only if, they have semantic updates. Projective updates are slightly more well-behaved: using a simple trick, one can see that a DL $\mathcal{L}$ has projective updates if the extension of $\mathcal{L}$ with the '@' constructor has semantic updates. Thus, we can ensure the existence of projective updates in the DLs between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$ by adding only nominals (but not the '@' constructor). Further relaxing projective updates to projective approximate updates turns out to not improve this situation.

The second main question studied in this paper concerns *the size of updated ABoxes*. The first relevant observation is that our construction of semantic updates in DLs that comprise nominals and the '@' constructor incurs an exponential blowup in the size of the update $\mathcal{U}$ and in the role depth of the original ABox $\mathcal{A}$, i.e., the nesting depth of existential and universal restrictions in $\mathcal{A}$. Although both measures are typically small in real applications, this raises the questions (i) whether an exponential blowup of semantic updates can be avoided by a more careful construction and (ii) whether other forms of update help to avoid an exponential blowup. Concerning (i), we show that an exponential blowup cannot be avoided unless NP $\cap$ co-NP is contained in the non-uniform version of the complexity class NC_1, which is considered very unlikely in complexity theory. Note that similar results such as those obtained by Cadoli et al. in [9] are not applicable due to the restricted form of updates considered in this paper (however, our results strengthen some of the results obtained by Cadoli et al.) Giving a positive answer to (ii), we then show that switching from semantic updates to projective ones dramatically improves the situation: in DLs that comprise nominals, not only the existence of projective updates is guaranteed, but it is even possible to construct projective updates whose size is polynomial both in the size of the original ABox and the update. Thus, projective updates are particularly well-suited for use in practical applications. We note that similar observations have already been made for the case of propositional logic in [9], where it is shown that for some update semantics, the projective version of updates is more succinct than the non-projective one.

Finally, we extend our update constructions to conditional updates, which can express statements such as '$A(a)$ is true after the update if $C(b)$ was true before'. We then apply the extended results to reasoning about actions in a DL context, as recently proposed and studied in [12, 13, 14]. In particular, we show that our construction of updated ABoxes can be used to implement a progression approach to the central *projection problem* [7], reproving a number of tight upper complexity bounds for this problem that have originally been obtained using the method of regression.

This paper is organized as follows. In Section 2, we provide a brief introduction to description logics, define the various kinds of ABox updates studied in this paper, and present some basic results regarding these updates. In particular, we interrelate the various definitions of updated ABoxes and prove that the '@' constructor is intimately related to projective updates, and to Boolean ABoxes.

The non-existence of updated ABoxes in standard expressive DLs of the $\mathcal{ALC}$ family is established in Section 3. We prove that DLs between $\mathcal{ALC}$ and $\mathcal{ALCQI}^{@}$ do not have approximate projective updates, and that DLs between $\mathcal{ALCO}$ and $\mathcal{ALCQIO}$ do not have approximate updates. This section is actually the only one where we explicitly consider (projective or non-projective) approximate updates since the positive results in subsequent sections all hold for stronger, non-approximating definitions of updated ABoxes.

In Section 4, we show that adding nominals and the '@' constructor to the DLs $\mathcal{ALC}$, $\mathcal{ALCI}$, $\mathcal{ALCQ}$, and $\mathcal{ALCQI}$ suffices to have semantic updates. We also establish the single-exponential size of updated ABoxes announced before, both for the case of single updates and iterated updates. We prove that an exponential blowup of updated ABoxes cannot be avoided, subject to the complexity-theoretic assumption that $\mathrm{NP} \cap \text{co-NP} \not\subseteq \mathrm{NC}_1$. Finally, we consider an extension of $\mathcal{ALCQIO}^{@}$ with certain role constructors which allows a simple construction of updated ABoxes that are exponential only in the size of the update, but not in the size of the original ABox.

We then focus on projective updates in Section 5, showing that they are enjoyed by all DLs between $\mathcal{ALCO}$ and $\mathcal{ALCQIO}^{@}$, including those that do not comprise the '@' constructor. We also show that projective updates can be constructed in polynomial time such that the resulting updated ABox is of polynomial size. With a small trick, these time and space bounds also apply to the case of iterated updates.

Section 6 is devoted to conditional updates and their application to reasoning about actions using DLs. Finally, Section 7 wraps up the paper, analyzing some possible extensions of our results (e.g. with TBoxes) and discussing related work.

This paper is a significantly extended and revised version of [15].

2. Preliminaries and Basic Definitions

We provide a brief introduction to description logics, define the various kinds of ABox updates studied in this paper, and present some basic observations regarding these updates. For the sake of readability, proofs for the results in this section are deferred to Appendix A.

2.1. Description Logics

We introduce the expressive description logic $\mathcal{ALCQIO}^{@}$ and its fragments studied in this paper. Our presentation will be brief and the reader is referred to [1] for more details. In DLs, *concepts* are inductively defined with the help of a set of *constructors*, starting with a countably infinite set $\mathsf{N_C}$ of *concept names*, a countably infinite set $\mathsf{N_R}$ of *role names*, and (possibly) a countably infinite set $\mathsf{N_I}$ of *individual names*. $\mathcal{ALCQIO}^{@}$-concepts are formed using the constructors shown in Figure 1. There, the inverse constructor is the only role constructor (used to construct compound roles), whereas the remaining seven constructors are concept constructors (used to construct compound concepts). A *role* is either a role name r or the inverse r^- of a role name r. In Figure 1 and

Name	Syntax	Semantics
inverse role	r^-	$(r^\mathcal{I})^{-1}$
nominal	$\{a\}$	$\{a^\mathcal{I}\}$
negation	$\neg C$	$\Delta^\mathcal{I} \setminus C^\mathcal{I}$
conjunction	$C \sqcap D$	$C^\mathcal{I} \cap D^\mathcal{I}$
disjunction	$C \sqcup D$	$C^\mathcal{I} \cup D^\mathcal{I}$
at-least restriction	$(\geqslant n\ r\ C)$	$\{x \in \Delta^\mathcal{I} \mid \#\{y \in C^\mathcal{I} \mid (x,y) \in r^\mathcal{I}\} \geq n\}$
at-most restriction	$(\leqslant n\ r\ C)$	$\{x \in \Delta^\mathcal{I} \mid \#\{y \in C^\mathcal{I} \mid (x,y) \in r^\mathcal{I}\} \leq n\}$
@ constructor	$@_a C$	$\Delta^\mathcal{I}$ if $a^\mathcal{I} \in C^\mathcal{I}$, and $\emptyset$ otherwise

Figure 1: Syntax and semantics of the description logic $\mathcal{ALCQIO}$.

throughout the paper, we use $\#S$ to denote the cardinality of a set S, a and b to denote individual names, r and s to denote roles, A, B to denote concept names, and C, D to denote (possibly compound) concepts. As usual, we use $\top$ as abbreviation for an arbitrary (but fixed) propositional tautology, $\bot$ for $\neg\top$, $\rightarrow$ and $\leftrightarrow$ for the usual Boolean abbreviations, $\exists r.C$ (*existential restriction*) for $(\geqslant 1\ r\ C)$, and $\forall r.C$ (*universal restriction*) for $(\leqslant 0\ r\ \neg C)$.

The fragment of $\mathcal{ALCQIO}^@$ that allows only for negation, conjunction, disjunction, and universal and existential restrictions is called $\mathcal{ALC}$. The availability of additional constructors is indicated by concatenation of a corresponding letter: $\mathcal{Q}$ stands for number restrictions; $\mathcal{I}$ stands for inverse roles, $\mathcal{O}$ for nominals and superscript '@' for the @ constructor. This explains the name $\mathcal{ALCQIO}^@$ for our DL, and also allows us to refer to fragments in a simple way. In particular, when we speak of *all DLs between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$* , we mean the logics $\mathcal{L}$ that can be obtained from $\mathcal{ALC}$ by all possible combinations of $\mathcal{Q}$, $\mathcal{I}$, and $\mathcal{O}$. In particular, $\mathcal{ALC}$ and $\mathcal{ALCQIO}$ themselves are regarded as DLs between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$. In the same way, we speak about all DLs between $\mathcal{ALC}^@$ and $\mathcal{ALCQIO}^@$ and, in general, about all DLs between any two descriptions logics. We note that, while the '@' constructor from hybrid logic [16] is somewhat unusual in a DL context, it will play an important role in the computation of updates later on.

The semantics of $\mathcal{ALCQIO}^@$-concepts is defined in terms of an *interpretation* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$. The *domain* $\Delta^\mathcal{I}$ is a non-empty set of individuals and the *interpretation function* $\cdot^\mathcal{I}$ maps

- each concept name $A \in \mathsf{N_C}$ to a subset $A^\mathcal{I}$ of $\Delta^\mathcal{I}$,
- each role name $r \in \mathsf{N_R}$ to a binary relation $r^\mathcal{I}$ on $\Delta^\mathcal{I}$, and
- each individual name $a \in \mathsf{N_I}$ to an element $a^\mathcal{I} \in \Delta^\mathcal{I}$ such that $a^\mathcal{I} \neq b^\mathcal{I}$ whenever a and b are distinct (the unique name assumption, UNA).

The extension of $\cdot^\mathcal{I}$ to inverse roles and arbitrary concepts is defined inductively as shown in the third column of Figure 1. Two concepts C and D are *equivalent*, written $C \equiv D$, iff $C^\mathcal{I} = D^\mathcal{I}$ for all interpretations $\mathcal{I}$.

In DLs, extensional knowledge is stored in an ABox. An $\mathcal{ALCQIO}^{@}$-*ABox* is a finite set of *concept assertions* $C(a)$, *role assertions* $r(a, b)$, and negated role assertions $\neg r(a, b)$, where r is a role *name*. For readability, we sometimes write concept assertions as $a{:}C$. As an abbreviation, we write $r^-(a, b) \in \mathcal{A}$ if $r(b, a)$ is contained in the ABox $\mathcal{A}$; similarly, $\neg r^-(a, b) \in \mathcal{A}$ abbreviates $\neg r(b, a) \in \mathcal{A}$. Observe that there is no need for explicitly introducing negated concept assertions due to the availability of negation as a concept constructor in $\mathcal{ALCQIO}^{@}$ and its fragments. An ABox $\mathcal{A}$ is *simple* if $C(a) \in \mathcal{A}$ implies that C is a *concept literal*, i.e., a concept name or a negated concept name. We use $\mathsf{Ind}(\mathcal{A})$ to denote the set of all individual names a used in the ABox $\mathcal{A}$ (i.e., all a such that there exists $C(a) \in \mathcal{A}$ or there exists $C(b) \in \mathcal{A}$ such that $@_a$ or $\{a\}$ occurs in C) and $\mathsf{role}(\mathcal{A})$ to denote the set of role names used in $\mathcal{A}$.

An interpretation $\mathcal{I}$ *satisfies* a concept assertion $C(a)$ iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$, a role assertion $r(a, b)$ iff $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$, and a negated role assertion $\neg r(a, b)$ iff $(a^{\mathcal{I}}, b^{\mathcal{I}}) \notin r^{\mathcal{I}}$. We write $\mathcal{I} \models \varphi$ to denote satisfaction of an ABox assertion φ by an intepretation $\mathcal{I}$. This notation is lifted to sets of interpretations Γ in the obvious way, i.e., we write $\Gamma \models \varphi$ iff $\mathcal{I} \models \varphi$ for all $\mathcal{I} \in \Gamma$. An interpretation $\mathcal{I}$ is a *model* of an ABox $\mathcal{A}$, written $\mathcal{I} \models \mathcal{A}$, if $\mathcal{I} \models \varphi$ for all $\varphi \in \mathcal{A}$.

We use $M(\mathcal{A})$ to denote the set of all models of the ABox $\mathcal{A}$. An ABox is *consistent* iff $M(\mathcal{A}) \neq \emptyset$. Two ABoxes $\mathcal{A}$ and $\mathcal{A}'$ are *equivalent*, written $\mathcal{A} \equiv \mathcal{A}'$, iff $M(\mathcal{A}) = M(\mathcal{A}')$. An ABox assertion φ is a *consequence* of an ABox $\mathcal{A}$, written $\mathcal{A} \models \varphi$, if $M(\mathcal{A}) \subseteq M(\{\varphi\})$. This notion is lifted to ABoxes in the obvious way: $\mathcal{A}'$ is a *consequence* of $\mathcal{A}$, written $\mathcal{A} \models \mathcal{A}'$, if $M(\mathcal{A}) \subseteq M(\mathcal{A}')$.

2.2. Semantic Updates

We introduce the most natural form of ABox updates which we call 'semantic' because of their purely model-theoretic definition. Such updates have also been called 'logical' updates in the literature, see for example [9]. We start with considering the update of an interpretation rather than an ABox.

Definition 1 (Interpretation Update). *An* update $\mathcal{U}$ *is a consistent simple ABox. Let $\mathcal{U}$ be an update and $\mathcal{I}$ an interpretation. Define an interpretation $\mathcal{I}^{\mathcal{U}}$ by setting, for all individual names a, concept names A, and role names r:*

$$\begin{array}{rcl}
\Delta^{\mathcal{I}^{\mathcal{U}}} & = & \Delta^{\mathcal{I}} \\
a^{\mathcal{I}^{\mathcal{U}}} & = & a^{\mathcal{I}} \\
A^{\mathcal{I}^{\mathcal{U}}} & = & (A^{\mathcal{I}} \cup \{a^{\mathcal{I}} \mid A(a) \in \mathcal{U}\}) \setminus \{a^{\mathcal{I}} \mid \neg A(a) \in \mathcal{U}\} \\
r^{\mathcal{I}^{\mathcal{U}}} & = & (r^{\mathcal{I}} \cup \{(a^{\mathcal{I}}, b^{\mathcal{I}}) \mid r(a, b) \in \mathcal{U}\}) \setminus \{(a^{\mathcal{I}}, b^{\mathcal{I}}) \mid \neg r(a, b) \in \mathcal{U}\}
\end{array}$$

$\mathcal{I}^{\mathcal{U}}$ *is called the* result of updating $\mathcal{I}$ with $\mathcal{U}$ *.*

As the next step, updates are lifted to the level of ABoxes, which represent classes of models rather than single models as in Definition 1.

Definition 2 (Semantic Update). *Let $\mathcal{A}$ be an $\mathcal{ALCQIO}^{@}$-ABox and $\mathcal{U}$ an update. Define the class of* updated models *as*

$$M(\mathcal{A})^{\mathcal{U}} = \{\mathcal{I}^{\mathcal{U}} \mid \mathcal{I} \in M(\mathcal{A})\}.$$

An $\mathcal{ALCQIO}^@$*-ABox* $\mathcal{A}'$ *is a* semantic update of $\mathcal{A}$ with $\mathcal{U}$*, written* $\mathcal{A} \Longrightarrow_{\mathcal{U}} \mathcal{A}'$*, if*

$$M(\mathcal{A}') = M(\mathcal{A})^{\mathcal{U}}.$$

A description logic $\mathcal{L}$ has semantic updates *if for every* $\mathcal{L}$*-ABox* $\mathcal{A}$ *and update* $\mathcal{U}$*, there is an* $\mathcal{L}$*-ABox* $\mathcal{A}'$ *with* $\mathcal{A} \Longrightarrow_{\mathcal{U}} \mathcal{A}'$*.*

To illustrate Definition 2, consider the following ABox $\mathcal{E}$, which we use as a running $\mathcal{E}$xample:

$$\begin{array}{l} \mathsf{john} : \exists \mathsf{has\text{-}child}.\mathsf{Happy} \\ \mathsf{mary} : \mathsf{Happy} \sqcap \mathsf{Clever} \end{array}$$

The following ABox $\mathcal{E}'$ is a semantic update of $\mathcal{E}$ with $\mathcal{U} = \{\neg\mathsf{Happy}(\mathsf{mary})\}$:

$$\begin{array}{l} \mathsf{john} : \exists \mathsf{has\text{-}child}.(\mathsf{Happy} \sqcup \{\mathsf{mary}\}) \\ \mathsf{mary} : \neg\mathsf{Happy} \sqcap \mathsf{Clever} \end{array}$$

To understand the disjunction, note that there are two kinds of models of $\mathcal{E}$: those where John has a happy child that is not Mary, and those where Mary is the only happy child of John. In models of the former kind, John still satisfies $\exists\mathsf{has\text{-}child}.\mathsf{Happy}$ after the update (first disjunct); in models of the latter kind, Mary is still a child of John after the update (second disjunct). For the sake of completeness, we provide a proof of the following in Appendix A:

Observation 1. $\mathcal{E} \Longrightarrow_{\mathcal{U}} \mathcal{E}'$*.*

As captured by the following lemma, semantic updates are unique up to logical equivalence and do not depend on the syntactic presentation of the original ABox. The lemma is an immediate consequence of the definition of semantic updates.

Lemma 1. *Let* $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_1', \mathcal{A}_2'$ *be* $\mathcal{ALCQIO}^@$*-ABoxes. Then* $\mathcal{A}_1 \equiv \mathcal{A}_2$ *and* $\mathcal{A}_i \Longrightarrow_{\mathcal{U}} \mathcal{A}_i'$ *for* $i \in \{1,2\}$ *implies* $\mathcal{A}_1' \equiv \mathcal{A}_2'$*.*

We remark that making the UNA, as we do, has an impact on semantic updates. To show the difference between updates with UNA and without, consider the ABox $\mathcal{A} = \{A(a_1)\}$ and the update $\mathcal{U} = \{\neg A(a_2)\}$, where $a_1 \neq a_2$. Then $\mathcal{A} \cup \mathcal{U}$ is a semantic update of $\mathcal{A}$ with $\mathcal{U}$ under UNA, but the semantic update of $\mathcal{A}$ with $\mathcal{U}$ without UNA is

$$\mathcal{U} \cup \{a_1 : (\{a_2\} \sqcup A)\}.$$

Thus, dropping the UNA results in a case distinctions regarding the identity of the individual names a_1 and a_2. Apart from such case distinctions, dropping the UNA poses no major technical problems.

Semantic updates are, in a sense, the 'ideal' kind of update. However, it turns out that many standard DLs such as $\mathcal{ALC}$ do not have semantic updates. For example, it can be proved that for the above $\mathcal{ALC}$-ABox $\mathcal{A}$ and update $\mathcal{U}$, there is no semantic update in $\mathcal{ALC}$ (and thus we had to resort to the $\mathcal{ALCO}$-ABox $\mathcal{A}'$ for presenting the semantic update). This problem, which is studied in detail in Section 3, motivates the consideration of other, weaker forms of updates.

2.3. Approximate Updates

We obtain a weaker form of update by considering the logical consequences of ABoxes instead of their models. This approach to weakening updates has been introduced in a DL context in [11].

Definition 3 (Approximate updates). *Let $\mathcal{A}$ be an $\mathcal{ALCQIO}^{@}$-ABox, $\mathcal{U}$ an update, and $\mathcal{L}$ a description logic. An $\mathcal{ALCQIO}^{@}$-ABox $\mathcal{A}'$ is an* approximate update of $\mathcal{A}$ with $\mathcal{U}$ regarding $\mathcal{L}$, *written $\mathcal{A} \longrightarrow^{\mathcal{L}}_{\mathcal{U}} \mathcal{A}'$, if for all $\mathcal{L}$-ABox assertions φ, we have*

$$M(\mathcal{A})^{\mathcal{U}} \models \varphi \quad \Leftrightarrow \quad M(\mathcal{A}') \models \varphi.$$

A description logic $\mathcal{L}$ has approximate updates *if for every $\mathcal{L}$-ABox $\mathcal{A}$ and update $\mathcal{U}$, there is an $\mathcal{L}$-ABox $\mathcal{A}'$ with $\mathcal{A} \longrightarrow^{\mathcal{L}}_{\mathcal{U}} \mathcal{A}'$.*

As an example, reconsider the example $\mathcal{ALC}$-ABox $\mathcal{E}$ and update $\mathcal{U}$ from Section 2.2. The following $\mathcal{ALC}$-ABox $\mathcal{E}''$ is an approximate update of $\mathcal{E}$ with $\mathcal{U}$ regarding $\mathcal{ALC}$:

$$\begin{array}{l}\mathsf{john} : \exists \mathsf{has\text{-}child}.(\mathsf{Happy} \sqcup \mathsf{Clever}) \\ \mathsf{mary} : \neg \mathsf{Happy} \sqcap \mathsf{Clever}\end{array}$$

Indeed, we prove in Appendix A:

Observation 2. *$\mathcal{E} \longrightarrow^{\mathcal{ALC}}_{\mathcal{U}} \mathcal{E}''$.*

Recall that, in contrast, there is no $\mathcal{ALC}$-ABox that is a semantic update of $\mathcal{E}$ with $\mathcal{U}$.

We now relate approximate updates to semantic updates in a precise way. If a semantic update exists, then an ABox is an approximate update regarding a DL $\mathcal{L}$ iff it has the same $\mathcal{L}$-consequences as the semantic update. This is Point 1 of the following Lemma, and it is an immediate consequence of the definition of approximate updates. Point 2 follows from Point 1 and asserts that semantic updates are approximate updates regarding *any* DL $\mathcal{L}$.

Lemma 2. *Let $\mathcal{A}$ and $\mathcal{A}'$ be $\mathcal{ALCQIO}^{@}$-ABoxes, $\mathcal{U}$ an update, and $\mathcal{L}$ a description logic. Assume $\mathcal{A} \Longrightarrow_{\mathcal{U}} \mathcal{A}_s$. Then*

1. *$\mathcal{A} \longrightarrow^{\mathcal{L}}_{\mathcal{U}} \mathcal{A}'$ iff for all $\mathcal{L}$-ABox assertions φ, $\mathcal{A}' \models \varphi \Leftrightarrow \mathcal{A}_s \models \varphi$;*
2. *$\mathcal{A} \longrightarrow^{\mathcal{L}}_{\mathcal{U}} \mathcal{A}_s$.*

Note that, in contrast to the semantic case, the definition of approximate updates depends on the DL $\mathcal{L}$ used. Indeed, the choice of $\mathcal{L}$ can make a difference: the $\mathcal{ALCO}$-ABox $\mathcal{E}'$ from Section 2.2 is a semantic update of $\mathcal{E}$ with $\mathcal{U}$, thus by Lemma 2 also an approximate update of $\mathcal{E}$ with $\mathcal{U}$ regarding $\mathcal{ALCO}$. Since clearly $\mathcal{E}'$ and the above $\mathcal{E}''$ have different $\mathcal{ALCO}$-consequences, $\mathcal{E}''$ is not an approximate update of $\mathcal{E}$ with $\mathcal{U}$ *regarding* $\mathcal{ALCO}$. However, as observed above, $\mathcal{E}''$ is an approximate update of $\mathcal{E}$ with $\mathcal{U}$ *regarding* $\mathcal{ALC}$.

We can derive interesting additional properties of approximate updates regarding a DL $\mathcal{L}$ when we demand that the updated ABox is formulated in the same DL $\mathcal{L}$, as in the definition of $\mathcal{L}$ 'having approximate updates'. Then, approximate updates are unique up to logical equivalence and also equivalent to semantic updates, if the latter exist. This is captured by Points 1 and 2 of the following lemma.

Lemma 3. *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be $\mathcal{ALCQIO}^{@}$-ABoxes, $\mathcal{U}$ an update, $\mathcal{L}$ a description logic, and $\mathcal{A}_1'$, $\mathcal{A}_2'$ $\mathcal{L}$-ABoxes. Then*

1. *$\mathcal{A}_1 \equiv \mathcal{A}_2$ and $\mathcal{A}_i \longrightarrow_{\mathcal{U}}^{\mathcal{L}} \mathcal{A}_i'$ for $i \in \{1, 2\}$ imply $\mathcal{A}_1' \equiv \mathcal{A}_2'$;*
2. *$\mathcal{A}_1 \longrightarrow_{\mathcal{U}}^{\mathcal{L}} \mathcal{A}_1'$ and $\mathcal{A}_1 \Longrightarrow_{\mathcal{U}} \mathcal{A}_2'$ imply $\mathcal{A}_1' \equiv \mathcal{A}_2'$.*

Point 1 is an immediate consequence of the fact that for any two $\mathcal{L}$-ABoxes $\mathcal{A}$ and $\mathcal{A}'$: $\mathcal{A} \models \varphi$ iff $\mathcal{A}' \models \varphi$ for all $\mathcal{L}$-ABox assertions φ implies $\mathcal{A} \models \mathcal{A}'$ and $\mathcal{A}' \models \mathcal{A}$, which in turn implies $\mathcal{A} \equiv \mathcal{A}'$. Point 2 follows together with Point 2 of Lemma 2. Note that, by Point 1 of Lemma 3, approximate updates do not depend on the syntactic presentation of the original ABox.

We remark that approximate updates are less generally useful than semantic ones. In particular, one main use of DL ABoxes is for query answering, see for example [2, 17]. While semantic updates give correct answers to queries formulated in *any* query language, approximate updates do not. For example, it follows directly from the definition that approximate updates regarding a DL $\mathcal{L}$ give correct answers to *instance queries* $C(a)$ with C formulated in $\mathcal{L}$, but this is not true for *unions of conjunctive queries (UCQs)*: the individual name john is not included in the certain answer to the following UCQ $q(x)$ when posed to the approximate update $\mathcal{E}''$ of $\mathcal{E}$ with $\mathcal{U}$ regarding $\mathcal{ALC}$ given above:

$$q(x) = \mathsf{has\text{-}child}(x, \mathsf{mary}) \vee (\exists y.\mathsf{has\text{-}child}(x, y) \wedge \mathsf{Happy}(y)).$$

In contrast, john is included in the certain answer to $q(x)$ when posed to the semantic update $\mathcal{E}'$ of $\mathcal{E}$ with $\mathcal{U}$ given in Section 2.2.

2.4. Semantic Projective Updates and Approximate Projective Updates

Although weaker, approximate updates turn out to be almost as elusive as semantic ones and are not enjoyed by many standard DLs, see Section 3. For this reason and to overcome the exponential blowup that we will encounter in the construction of semantic updates (when they exist), we consider an additional way of relaxing updated ABoxes, namely to allow additional 'auxiliary' symbols (concept names, role names, and individual names) in the updated ABox. This can be done both for semantic updates and approximate updates, which gives rise to the four forms of update studied in this paper. Updates admitting auxiliary symbols have been studied in [9, 10] in a propositional logic context.

Elements of $\mathsf{N_C} \cup \mathsf{N_R} \cup \mathsf{N_I}$ are called *symbols*. A *signature* is a set of symbols. The signature $\mathsf{sig}(C)$ *of a concept* C is the set of symbols that occur in C. The signature $\mathsf{sig}(\mathcal{A})$ of an ABox $\mathcal{A}$ is defined likewise; in particular all individual names used in $\mathcal{A}$ are included in $\mathsf{sig}(\mathcal{A})$. For a signature S, we use $\overline{S}$ to denote $(\mathsf{N_C} \cup \mathsf{N_R} \cup \mathsf{N_I}) \setminus S$. The *reduct* $\mathcal{I}_{\restriction S}$ of an interpretation $\mathcal{I}$ to a signature S is the interpretation obtained from $\mathcal{I}$ by 'forgetting' the interpretation of all symbols that are not in S; i.e., $\mathcal{I}_{\restriction S}$ interprets only the symbols in S, but not other symbols. The notion of reducts is lifted to classes of models M by setting $M_{\restriction S} = \{\mathcal{I}_{\restriction S} \mid \mathcal{I} \in M\}$.

Definition 4 (Projective updates).
Let $\mathcal{A}$ and $\mathcal{A}'$ be $\mathcal{ALCQIO}^{@}$-ABoxes, $\mathcal{U}$ an update, and $\mathcal{L}$ a description logic. Then $\mathsf{Fr}(\mathcal{A}') = \mathsf{sig}(\mathcal{A}') \setminus (\mathsf{sig}(\mathcal{A}) \cup \mathsf{sig}(\mathcal{U}))$ is the set of fresh symbols *of $\mathcal{A}'$. We call $\mathcal{A}'$ a*

- semantic projective update of $\mathcal{A}$ with $\mathcal{U}$, *written $\mathcal{A} \Longrightarrow^{\mathsf{p}}_{\mathcal{U}} \mathcal{A}'$, if*

$$M(\mathcal{A}')_{\restriction \overline{\mathsf{Fr}(\mathcal{A}')}} = M(\mathcal{A})^{\mathcal{U}}_{\restriction \overline{\mathsf{Fr}(\mathcal{A}')}}.$$

- approximate projective update of $\mathcal{A}$ with $\mathcal{U}$ regarding $\mathcal{L}$, *written $\mathcal{A} \longrightarrow^{\mathsf{p},\mathcal{L}}_{\mathcal{U}} \mathcal{A}'$, if for all $\mathcal{L}$-assertions φ with $\mathsf{sig}(\varphi) \subseteq \overline{\mathsf{Fr}(\mathcal{A}')}$, we have*

$$M(\mathcal{A})^{\mathcal{U}} \models \varphi \quad \Leftrightarrow \quad M(\mathcal{A}') \models \varphi.$$

A description logic $\mathcal{L}$

- has semantic projective updates *iff for every $\mathcal{L}$-ABox $\mathcal{A}$ and every update $\mathcal{U}$, there is an $\mathcal{L}$-ABox $\mathcal{A}'$ such that $\mathcal{A} \Longrightarrow^{\mathsf{p}}_{\mathcal{U}} \mathcal{A}'$;*
- has approximate projective updates *iff for every $\mathcal{L}$-ABox $\mathcal{A}$ and every update $\mathcal{U}$, there is an $\mathcal{L}$-ABox $\mathcal{A}'$ such that $\mathcal{A} \longrightarrow^{\mathsf{p},\mathcal{L}}_{\mathcal{U}} \mathcal{A}'$.*

As an example, consider the following ABox $\mathcal{F}$

$$\mathsf{john} : \exists \mathsf{has\text{-}friend}.\mathsf{Smart}$$

and update $\mathcal{U} = \{\neg\mathsf{has\text{-}friend}(\mathsf{john}, \mathsf{mary})\}$. We will show in the proof of Theorem 2 that there is no $\mathcal{L}$-ABox that is a semantic update of $\mathcal{F}$ with $\mathcal{U}$ and neither an $\mathcal{L}$-ABox that is an approximate update of $\mathcal{F}$ with $\mathcal{U}$ regarding $\mathcal{L}$, for many DLs $\mathcal{L}$ including $\mathcal{ALC}$ and $\mathcal{ALCO}$. However, we find a semantic projective update $\mathcal{F}'$ formulated in $\mathcal{ALCO}$ that consists of $\mathcal{U}$ and

$$\mathsf{john} : (\exists \mathsf{has\text{-}friend}.\mathsf{Smart}) \sqcup \exists \mathsf{r}.(\{\mathsf{mary}\} \sqcap \mathsf{Smart}),$$

where r is a fresh role name.

Observation 3. *$\mathcal{F} \Longrightarrow^{\mathsf{p}}_{\mathcal{U}} \mathcal{F}'$.*

We leave a formal proof of Observation 3 to the reader and only give an intuition of why $\mathcal{F}'$ is a semantic projective update. Again, there are two kinds of models of $\mathcal{A}$: those where John has a smart friend that is not Mary, and those where Mary is the only smart friend of John. In models of the former kind, John still satisfies $\exists\mathsf{has\text{-}friend}.\mathsf{Smart}$ after the update (first disjunct); in models of the latter kind, Mary is still smart after the update. This is expressed by the second disjunct, where the role name r only serves the technical purpose of 'jumping' from John to Mary in updated interpretations.

It is not hard to establish the following analogue of Lemma 2.

Lemma 4. *Let $\mathcal{A}$ and $\mathcal{A}'$ $\mathcal{ALCQIO}^{@}$-ABoxes, $\mathcal{U}$ an update, $\mathcal{L}$ a description logic, and assume $\mathcal{A} \Longrightarrow^{\mathsf{p}}_{\mathcal{U}} \mathcal{A}_s$. Then*

1. $\mathcal{A} \longrightarrow^{\mathsf{p},\mathcal{L}}_{\mathcal{U}} \mathcal{A}'$ *iff for all $\mathcal{L}$-ABox assertions φ with $\mathsf{sig}(\varphi) \subseteq \overline{\mathsf{Fr}(\mathcal{A}') \cup \mathsf{Fr}(\mathcal{A}_s)}$, we have $\mathcal{A}' \models \varphi \Leftrightarrow \mathcal{A}_s \models \varphi$;*
2. $\mathcal{A} \longrightarrow^{\mathsf{p},\mathcal{L}}_{\mathcal{U}} \mathcal{A}_s$.

Concerning analogues of Lemmas 1 and 3, an obvious first observation is that, due to the use of fresh symbols, semantic projective updates need not be logically equivalent to each other, and neither do approximate projective updates. However, it is still a consequence of Definition 4 that semantic projective updates and approximate projective updates do not depend on the syntactic form of the original ABox.

Lemma 5. *Let $\mathcal{A}_1, \mathcal{A}_2$, and $\mathcal{A}'$ be $\mathcal{ALCQIO}^{@}$-ABoxes, $\mathcal{U}$ an update, and $\mathcal{L}$ a description logic. Then*

1. $\mathcal{A}_1 \equiv \mathcal{A}_2$ *and* $\mathcal{A}_1 \Longrightarrow^{\mathsf{p}}_{\mathcal{U}} \mathcal{A}'$ *imply* $\mathcal{A}_2 \Longrightarrow^{\mathsf{p}}_{\mathcal{U}} \mathcal{A}'$;
2. $\mathcal{A}_1 \equiv \mathcal{A}_2$ *and* $\mathcal{A}_1 \longrightarrow^{\mathsf{p},\mathcal{L}}_{\mathcal{U}} \mathcal{A}'$ *imply* $\mathcal{A}_2 \longrightarrow^{\mathsf{p},\mathcal{L}}_{\mathcal{U}} \mathcal{A}'$.

2.5. The '@' constructor, Boolean ABoxes and projective updates

The example in Section 2.2 illustrates that, sometimes, nominals can help to overcome the non-existence of updates. The example in Section 2.4 shows that the same is true for projective updates. Indeed, we will show that among the DLs introduced in Section 2.1, exactly the DLs that include nominals have semantic projective updates (but not necessarily non-projective updates). Interestingly, the positive effects of projective updates (but not those of nominals) can also be attained in two other ways: by adding the '@' concept constructor as introduced in Section 2.1 and by replacing ABoxes with *Boolean ABoxes*. The latter are sets of *Boolean ABox assertions*, i.e., combinations of ABox assertions expressed in terms of the connectives $\wedge$ and $\vee$ (recall that ABox assertions are already closed under negation).

For illustration, reconsider the example from Section 2.4, i.e. the ABox $\mathcal{F} = \{\mathsf{john} : \exists \mathsf{has\text{-}friend}.\mathsf{Smart}\}$ update $\mathcal{U} = \{\neg \mathsf{has\text{-}friend}(\mathsf{john}, \mathsf{mary})\}$. Recall that the semantic *projective* update $\mathcal{F}'$ of $\mathcal{F}$ consists of $\mathcal{U}$ and

$$\mathsf{john} : (\exists \mathsf{has\text{-}friend}.\mathsf{Smart}) \sqcup \exists \mathsf{r}.(\{\mathsf{mary}\} \sqcap \mathsf{Smart}).$$

To eliminate the auxiliary symbol r and thus obtain a semantic *non-projective* update, we can use the '@' concept constructor and replace the above assertion with

$$\mathsf{john} : (\exists \mathsf{has\text{-}friend}.\mathsf{Smart}) \sqcup @_{\mathsf{mary}}\mathsf{Smart}.$$

Alternatively, we can eliminate the symbol r by using Boolean ABoxes, replacing the above assertion with

$$(\mathsf{john} : \exists \mathsf{has\text{-}friend}.\mathsf{Smart}) \vee (\mathsf{mary} : \mathsf{Smart}).$$

It is due to the simplicity of this example that the two presentations of the semantic non-projective update do not involve nominals: in general, we might still have to use nominals even when the '@' constructor or Boolean ABoxes are admitted. Indeed, we will show that among the DLs introduced in Section 2.1, exactly those have semantic (non-projective!) updates that comprise nominals and the '@' constructor. A similar statement can be formulated for Boolean ABoxes.

The aim of the current section is to present some basic observations regarding the relationship between the '@' constructor, Boolean ABoxes, and projective updates. The following lemma shows that non-Boolean $\mathcal{L}^@$-ABoxes have exactly the same expressive power as Boolean $\mathcal{L}$-ABoxes provided that $\mathcal{L}$ contains nominals. This does not hold, e.g., for $\mathcal{ALC}$: while every $\mathcal{ALC}^@$-ABox can be translated into an equivalent Boolean $\mathcal{ALC}$-ABox, it can be proved that no non-Boolean $\mathcal{ALC}^@$-ABox is equivalent to the Boolean $\mathcal{ALC}$-ABox $\{A(a) \vee r(b,c)\}$. A proof of the following lemma can be found in Appendix A.

Lemma 6.

1. *Let $\mathcal{L}$ be a DL between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$. Then for every Boolean $\mathcal{L}^@$-ABox, there exists an equivalent Boolean $\mathcal{L}$-ABox;*
2. *Let $\mathcal{L}$ be a DL between $\mathcal{ALCO}$ and $\mathcal{ALCQIO}$. Then for every Boolean $\mathcal{L}$-ABox, there exists an equivalent non-Boolean $\mathcal{L}^@$-ABox.*

We remark that the translation of a Boolean $\mathcal{L}$-ABox into an $\mathcal{L}^@$-ABox involves an exponential blowup while the converse translation does not.

Finally, the relationship between the '@' constructor / Boolean ABoxes and projective updates can easily be established by simulating the '@' constructor with a fresh role, as in the ABox $\mathcal{F}'$ in the above example.

Lemma 7. *Let $\mathcal{L}$ be a DL between $\mathcal{ALCO}$ and $\mathcal{ALCQIO}$. Then for every $\mathcal{L}^@$-ABox $\mathcal{A}$, there exists an $\mathcal{L}$-ABox $\mathcal{A}'$ such that*

$$M(\mathcal{A})_{\restriction \overline{\{r\}}} = M(\mathcal{A}')_{\restriction \overline{\{r\}}}$$

where r is a role name that does not occur in $\mathcal{A}$.

Proof.(sketch) Let $\mathcal{A}$ be an $\mathcal{L}^@$-ABox and $\mathcal{U}$ an update. Construct an $\mathcal{L}$-ABox $\mathcal{A}'$ as follows. First convert all concepts in $\mathcal{A}$ into negation normal form (NNF), in which negation occurs only in front of concept names, but not in front of complex concepts [1]. Then replace every concept $@_a C$ with $\exists r.(\{a\} \sqcap C)$, r a role name not used in $\mathcal{A}$. It can be proved that the resulting ABox $\mathcal{A}'$ is as required. $\square$

3. Non-existence of Updates

We present two general non-existence results for updates. First, we show that among the DLs introduced in Section 2.1, those without nominals do not have approximate projective updates, thus no semantic updates, approximate updates, and semantic projective updates either.

Theorem 1. *Let $\mathcal{L}$ be a DL between $\mathcal{ALC}$ and $\mathcal{ALCQI}^{@}$. Then $\mathcal{L}$ does not have approximate projective updates.*

Proof. We exhibit an $\mathcal{ALC}$-ABox and update for which we show that no approximate projective update exists in any of the DLs listed in Theorem 1.

Let $\mathcal{A} = \{a : \exists r.A, r(b,a)\}$, $\mathcal{U} = \{\neg A(b)\}$, and

$$\mathcal{A}' = \{\neg A(b), r(b,a), a : \exists r.(A \sqcup \{b\})\}.$$

We first show that the $\mathcal{ALCO}$-ABox $\mathcal{A}'$ is a semantic update of $\mathcal{A}$ with $\mathcal{U}$. Let $\mathcal{I}$ be a model of $\mathcal{A}$. We have to show that $\mathcal{I}^{\mathcal{U}}$ is a model of $\mathcal{A}'$. By definition, $\mathcal{I}^{\mathcal{U}} \models r(b,a)$ and $\mathcal{I}^{\mathcal{U}} \models \neg A(b)$. It remains to show that $\mathcal{I}^{\mathcal{U}} \models a : \exists r.(A \sqcup \{b\})$. First assume that $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$. Then $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}^{\mathcal{U}}}$ and, therefore, $\mathcal{I}^{\mathcal{U}} \models a : \exists r.(A \sqcup \{b\})$. Now assume $(a^{\mathcal{I}}, b^{\mathcal{I}}) \notin r^{\mathcal{I}}$. Then, since $\mathcal{I} \models a : \exists r.A$, there exists $d \neq b^{\mathcal{I}}$ with $(a^{\mathcal{I}}, d) \in r^{\mathcal{I}}$ and $d \in A^{\mathcal{I}}$. But then $(a^{\mathcal{I}}, d) \in r^{\mathcal{I}^{\mathcal{U}}}$ and $d \in A^{\mathcal{I}^{\mathcal{U}}}$. Again $\mathcal{I}^{\mathcal{U}} \models a : \exists r.(A \sqcup \{b\})$. Conversely, assume that $\mathcal{I}'$ is a model of $\mathcal{A}'$. We have to show that there exists a model $\mathcal{I}$ of $\mathcal{A}$ such that $\mathcal{I}^{\mathcal{U}} = \mathcal{I}'$. Let $\mathcal{I}$ coincide with $\mathcal{I}'$ except that $b^{\mathcal{I}'} \in A^{\mathcal{I}}$ if $(a^{\mathcal{I}'}, b^{\mathcal{I}'}) \in r^{\mathcal{I}'}$. Then $\mathcal{I}$ is a model of $\mathcal{A}$ and $\mathcal{I}' = \mathcal{I}^{\mathcal{U}}$, as required.

We show that there exists no $\mathcal{ALCQI}^{@}$-ABox $\mathcal{B}$ that is an approximate projective update of $\mathcal{A}$ with $\mathcal{U}$ regarding $\mathcal{ALC}$. It follows that for all of the DLs $\mathcal{L}$ in Theorem 1, there is no $\mathcal{L}$-ABox that is an approximate projective update of $\mathcal{A}$ with $\mathcal{U}$ regarding $\mathcal{L}$.

Assume to the contrary that such a $\mathcal{B}$ exists. We start with a high-level sketch of the proof. Let $\exists r^n.C$ denote the n-fold nesting $\exists r. \cdots .\exists r.C$, with $\exists r^0.C = C$. We first observe that $a : \exists r.A$ is not an $\mathcal{ALC}$-consequence of $\mathcal{B}$, while $a : \exists r.(A \sqcup (\exists r^n.\top))$ is an $\mathcal{ALC}$-consequence of $\mathcal{B}$ for each $n > 0$. Since $\mathcal{B}$ is finite and formulated in $\mathcal{ALCQI}^{@}$, it cannot impose any constraints on domain element that exceed a certain 'distance' (in terms of the length of shortest role paths in an interpretation) from any individual name in $\mathcal{B}$. Thus, to entail all assertions $a : \exists r.(A \sqcup (\exists r^n.\top))$ without entailing $a : \exists r.A$, $\mathcal{B}$ must enforce an r-cycle. Using a careful modification of the well-known unraveling technique [18, 17] and certain guaranteed $\mathcal{ALC}$-consequences of $\mathcal{B}$, we show that no such cycle is actually enforced by $\mathcal{B}$.

We start with establishing some relevant (non)-entailments of $\mathcal{B}$.

Claim 1.

(i) $\mathcal{B} \not\models a : \exists r.A$;
(ii) $\mathcal{B} \models a : \exists r.(A \sqcup (\exists r^n.\top))$, for all $n \geq 0$;
(iii) $\mathcal{B} \not\models a : \exists r^2.\top$;

To prove (i), note that $\mathcal{A}' \not\models a : \exists r.A$. We obtain that $\mathcal{B} \not\models a : \exists r.A$ because $\mathcal{B}$ is an approximate projective update, $a, r, A \in \mathsf{sig}(\mathcal{A})$ and since $a : \exists r.A$ is an $\mathcal{ALC}$-assertion.

By the arguments used in the proof of (i), we know that (ii) can be proved by showing that $\mathcal{A}' \models a : \exists r.(A \sqcup (\exists r^n.\top))$ for all $n > 0$. Due to the fact that

$\mathcal{A}' \models a : \exists r.(A \sqcup \{b\})$, for every model $\mathcal{I}$ of $\mathcal{A}'$, we have that $a^\mathcal{I} \notin (\exists r.A)^\mathcal{I}$ implies $(a^\mathcal{I}, b^\mathcal{I}) \in r^\mathcal{I}$, which in turn yields $a \in (\exists r^n.\top)^\mathcal{I}$ for all $n \geq 1$ since $r(b,a) \in \mathcal{A}'$.

To prove (iii), note that the interpretation $\mathcal{I}$ with $\Delta^\mathcal{I} = \{a, b, x\}$, $a^\mathcal{I} = a$, $b^\mathcal{I} = b$, $r^\mathcal{I} = \{(b,a),(a,x)\}$, and $A^\mathcal{I} = \{x\}$ is a model of $\mathcal{A}'$ such that $a^\mathcal{I} \notin (\exists r^2.\top)^\mathcal{I}$. Hence $\mathcal{A}' \not\models a : \exists r^2.\top$. We obtain $\mathcal{B} \not\models a : \exists r^2.\top$ because $\mathcal{B}$ is an approximate projective update, $a, r \in \mathsf{sig}(\mathcal{A})$ and since $a : \exists r^2.\top$ is an $\mathcal{ALC}$-assertion.

Now take a model $\mathcal{I}$ of $\mathcal{B}$ with $\mathcal{I} \not\models a : \exists r.A$. We unravel $\mathcal{I}$ into a new model $\mathcal{J}$ of $\mathcal{B}$ that has a forest-like shape and still satisfies $\mathcal{J} \not\models a : \exists r.A$. After the unraveling, we further modify $\mathcal{J}$ which allows us to derive a contradiction to Point (ii) of Claim 1.

As we want to preserve all $\mathcal{ALCQI}^@$-concepts in concept assertions in $\mathcal{B}$, we apply an unraveling construction in which role-predecessors are not duplicated. In detail, let $\Delta^\mathcal{J}$ be the set of all words $w = d_0 s_0 d_1 s_1 \cdots s_{k-1} d_k$, $k \geq 0$, such that

1. $d_1, \ldots, d_k \in \Delta^\mathcal{I}$;
2. $s_0, \ldots, s_{k-1}$ are roles (i.e. role names or their inverses);
3. there is a $c \in \mathsf{N_I}$ such that $d_0 = c^\mathcal{I}$;
4. for all $i < k$, we have $(d_i, d_{i+1}) \in s_i^\mathcal{I}$ and if $s_i = s_{i+1}^-$, then $d_i \neq d_{i+2}$;
5. if $d_0 = c_0^\mathcal{I}$ and $d_1 = c_1^\mathcal{I}$ for $c_0, c_1 \in \mathsf{N_I}$, then $\mathcal{B} \not\models s_0(c_0, c_1)$.

Condition 4 is the standard approach for dealing with the presence of both number restrictions and inverse roles. Point 5 will be explained below. Define the interpretation of symbols in $\mathcal{J}$ as follows:

- $B^\mathcal{J} := \{d_0 \cdots d_k \in \Delta^\mathcal{J} \mid d_k \in B^\mathcal{I}\}$ for all $B \in \mathsf{N_C}$;
- for all $s \in \mathsf{N_R}$, $$s^\mathcal{J} := \{(c_1^\mathcal{I}, c_2^\mathcal{I}) \mid \mathcal{B} \models s(c_1, c_2)\} \cup \{(w, wsd) \mid w, wsd \in \Delta^\mathcal{J}\} \cup \{(ws^-d, w) \mid w, ws^-d \in \Delta^\mathcal{J}\}.$$
- $c^\mathcal{J} := c^\mathcal{I}$ for all $c \in \mathsf{N_I}$.

Note the careful definition of $s^\mathcal{J}$, where we do not include all pairs $(c_1^\mathcal{I}, c_2^\mathcal{I})$ with $(c_1^\mathcal{I}, c_2^\mathcal{I}) \in s^\mathcal{I}$ as is often done when unraveling models of ABoxes. Indeed, only this careful definition and Condition 5 above ensure that ABox elements are not duplicated during unraveling, which would cause conflicts with number restrictions.

Claim 2. For all $\mathcal{ALCQI}^@$-concepts C and all $w = d_0 \cdots d_k \in \Delta^\mathcal{J}$, we have $w \in C^\mathcal{J}$ iff $d_k \in C^\mathcal{I}$.

The proof is by induction on the structure of C. We only do the cases $C = @_c D$ and $C = (\leqslant n\ s\ D)$ from the induction step, leaving the remaining cases to the reader.

Let $w = d_0 \cdots d_k \in \Delta^{\mathcal{J}}$ and $C = @_c D$. Then $d_k \in (@_c D)^{\mathcal{I}}$ iff $c^{\mathcal{I}} \in D^{\mathcal{I}}$ iff (by IH) $c^{\mathcal{J}} \in D^{\mathcal{J}}$ iff $w \in (@_c D)^{\mathcal{J}}$.

Now let $C = (\leqslant n\ s\ D)$ and $w = d_0 \cdots d_k$. Assume first that $k > 0$. Let, for any $e \in \Delta^{\mathcal{J}}$, $s^{\mathcal{I}}(e) = \{d \in \Delta^{\mathcal{I}} \mid (e, d) \in s^{\mathcal{I}}\}$, and likewise for $s^{\mathcal{J}}(e)$. Now, using the IH and Condition 4, one can show that $s^{\mathcal{I}}(d_k) \cap D^{\mathcal{I}}$ has the same cardinality as $s^{\mathcal{J}}(w) \cap D^{\mathcal{J}}$. Therefore $d_k \in (\leqslant n\ s\ D)^{\mathcal{I}}$ iff $w \in (\leqslant n\ s\ D)^{\mathcal{J}}$, as required.

Now assume that $k = 0$. Then $w = d_0 \in \Delta^{\mathcal{I}} \cap \Delta^{\mathcal{J}}$ and there exists a $c_0 \in \mathsf{N_I}$ with $c_0^{\mathcal{I}} = d_0$. If $d \in s^{\mathcal{I}}(d_0)$ and there is a $c \in \mathsf{N_I}$ with $c^{\mathcal{I}} = d$ and $\mathcal{B} \models s(c_0, c)$, then $(d_0, d) \in s^{\mathcal{J}}$ by definition of $s^{\mathcal{J}}$. If $d \in s^{\mathcal{I}}(d_0)$ and there is no such c, then $(d_0, d_0 s d) \in s^{\mathcal{J}}$. Thus, there exists a bijection between $s^{\mathcal{I}}(d_0)$ and $s^{\mathcal{J}}(d_0)$ and we obtain, by applying the IH to the pairs in this bijection, that $d_0 \in (\leqslant n\ s\ D)^{\mathcal{I}}$ iff $d_0 \in (\leqslant n\ s\ D)^{\mathcal{J}}$, as required.

Claim 2 implies that $\mathcal{J} \not\models a : \exists r.A$ as intended. Moreover, we have

Claim 3. $\mathcal{J}$ is a model of $\mathcal{B}$.

To prove Claim 3, let $\varphi \in \mathcal{B}$.

1. If $\varphi = C(c)$ for an $\mathcal{ALCQI}^{@}$-concept C, then we have $\mathcal{I} \models C(c)$, and thus $\mathcal{J} \models C(c)$ by Claim 2.
2. If $\varphi = s(c_1, c_2)$ for a role s name, then we have $\mathcal{B} \models s(c_1, c_2)$, and thus $\mathcal{J} \models s(c_1, c_2)$ by definition of $s^{\mathcal{J}}$,
3. If $\varphi = \neg s(c_1, c_2)$, then we have $\mathcal{B} \not\models s(c_1, c_2)$ (since $\mathcal{B}$ is consistent) and therefore $\mathcal{J} \not\models s(c_1, c_2)$ by definition of $s^{\mathcal{J}}$. Thus $\mathcal{J} \models \neg s(c_1, c_2)$.

This finishes the proof of Claim 3.

We define the *depth* $d(C)$ of an $\mathcal{ALCQI}^{@}$-concept C as the nesting depth of number restrictions in C, with a 'reset' triggered by the '@' constructor, i.e.,

$$
\begin{array}{rcrcl}
d(B) & = & d(@_a C) & = & 0 \\
d(C \sqcap D) & = & d(C \sqcup D) & = & \max\{d(C), d(D)\} \\
d(\leqslant n\ r\ C) & = & d(\geqslant n\ r\ C) & = & d(C) + 1 \\
 & & d(\neg C) & = & d(C)
\end{array}
$$

The *depth* $d(\mathcal{B})$ of $\mathcal{B}$ is defined as $\max\{d(C) \mid C(c) \in \mathcal{B}\}$. As the next step, we further modify $\mathcal{J}$ by 'cutting off' all paths in $\Delta^{\mathcal{J}}$ that are not in $\mathcal{B}$ at length $d(\mathcal{B})$. Thus, let $\Delta^{\mathcal{J}'} = \{d_0 \cdots d_k \in \Delta^{\mathcal{J}} \mid k \leq d(\mathcal{B})\}$, let $B^{\mathcal{J}'}$ and $s^{\mathcal{J}'}$ be the restrictions of $B^{\mathcal{J}}$ and $s^{\mathcal{J}}$ to $\Delta^{\mathcal{J}'}$ for all $B \in \mathsf{N_C}$ and $s \in \mathsf{N_R}$, and let $c^{\mathcal{J}'} = c^{\mathcal{J}}$ for all $c \in \mathsf{N_I}$ (clearly, no $c^{\mathcal{J}}$ is dropped by the restriction of $\Delta^{\mathcal{J}}$). One can show by induction on the structure of C:

Claim 4. For all $\mathcal{ALCQI}^{@}$-concepts C with $d(C) = i \leq d(\mathcal{B})$ and all $w = d_0 \cdots d_k \in \Delta^{\mathcal{J}}$ with $k \leq d(\mathcal{B}) - i$, we have $w \in C^{\mathcal{J}}$ iff $w \in C^{\mathcal{J}'}$.

Claims 3 and 4 imply that $\mathcal{J}'$ is a model of $\mathcal{B}$: role assertions $\varphi \in \mathcal{B}$ are clearly not invalidated when constructing $\mathcal{J}'$ from $\mathcal{J}$ and concept assertions $C(c) \in \mathcal{B}$ are satisfied by Claim 4 and since they were satisfied in $\mathcal{J}$.

By Point (ii) of Claim 1, to obtain a contradiction it thus remains to show that there exists an $n > 0$ such that $\mathcal{J}' \not\models a : \exists r.(A \sqcup (\exists r^n.\top))$. Set $n = d(\mathcal{B})+1$. First observe that $\mathcal{J}' \not\models a : \exists r.A$ because $\mathcal{J} \not\models a : \exists r.A$. It remains to show that $\mathcal{J}' \not\models a : \exists r^{n+1}.\top$. Observe that by Point (iii) of Claim 1, in $\mathcal{B}$ there is no r-chain of length larger than 1 starting from a (more precisely: we have $m \leq 1$ for any m with $r(a, c_1), r(c_1, c_2), \ldots, r(c_{m-1}, c_m) \in \mathcal{B}$ for some $c_1, \ldots, c_m \in \mathsf{N_I}$). Thus, by construction of $\mathcal{J}'$, all r-paths $d_0, \ldots, d_k$ in $\mathcal{J}'$ with $d_0 = a^{\mathcal{J}'}$ have length $k \leq d(\mathcal{B}) + 1$. Thus, $\mathcal{J}' \not\models a : \exists r^{n+1}.\top$, as required. $\square$

Our second non-existence result for updates states that among the DLs introduced in Section 2.1 that include nominals, those that lack the '@' constructor do not have approximate updates, thus no semantic updates either. In contrast to the DLs considered in the previous theorem, the DLs addressed here do have projective semantic updates (thus also projective approximate updates), see Section 4.

Theorem 2. *Let $\mathcal{L}$ be a DL between $\mathcal{ALCO}$ and $\mathcal{ALCQIO}$. Then $\mathcal{L}$ does not have approximate updates.*

Proof. Let $\mathcal{L}$ be a DL between $\mathcal{ALCO}$ and $\mathcal{ALCQIO}$. We construct an $\mathcal{ALC}$-ABox $\mathcal{A}$ and update $\mathcal{U}$ such that there is no $\mathcal{ALCQIO}$-ABox $\mathcal{A}'$ that is an approximate update of $\mathcal{A}$ with $\mathcal{U}$ regarding $\mathcal{ALC}$. Let $\mathcal{A} = \{a : \exists r.A\}$, $\mathcal{U} = \{\neg r(a, b)\}$ and

$$\mathcal{A}' = \{a : \exists r.A \sqcup @_b A, \neg r(a, b)\}.$$

It is not difficult to show that the $\mathcal{ALC}^{@}$-ABox $\mathcal{A}'$ is a semantic update of $\mathcal{A}$ with $\mathcal{U}$, thus also an approximate update. It suffices to show that there is no $\mathcal{ALCQIO}$-ABox $\mathcal{B}$ with $\mathcal{A}' \models \varphi$ iff $\mathcal{B} \models \varphi$ for all $\mathcal{ALC}$-assertions φ. Assume to the contrary that such a $\mathcal{B}$ exists, and choose a role name s that does not occur in $\mathcal{A}'$ and $\mathcal{B}$ (such a role name exists since $\mathcal{B}$ is finite). Now consider the interpretations $\mathcal{I}$ and $\mathcal{I}'$ displayed in Figure 2. We assume that the individual names a and b are mapped to the individuals of the same name as shown in the figure. To satisfy the UNA, we also assume that there is an infinite set of additional points that interpret the individual names distinct from a and b. On these additional points, all concept and role names are interpreted as empty. Note that $\mathcal{I}$ and $\mathcal{I}'$ are models of $\mathcal{A}'$. By Point 2 of Lemma 3, they are thus also models of $\mathcal{B}$. Consider the additional interpretation $\mathcal{I}''$ in Figure 3. We show that $\mathcal{I}'' \models \mathcal{B}$ and $\mathcal{I}'' \not\models \mathcal{B}$, thus derive a contradiction.

Claim 1. $\mathcal{I}'' \not\models \mathcal{B}$.

Assume $\mathcal{I}'' \models \mathcal{B}$. Define $C = \neg A \sqcap \exists s.(\{a\} \sqcap \forall r.\neg A)$. Clearly, $\mathcal{I}'' \models C(b)$. Since $\mathcal{I}'' \models \mathcal{B}$, it follows that $\mathcal{B} \not\models \neg C(b)$. Hence, $\mathcal{A}' \not\models \neg C(b)$. This is a contradiction to the fact that $\mathcal{A}' \models \neg C(b)$ (note that $\neg C \equiv A \sqcup \forall s.(\{a\} \rightarrow \exists r.A)$).

Claim 2. $\mathcal{I}'' \models \mathcal{B}$.

To prove this claim we require an observation regarding models of $\mathcal{ALCQIO}$-ABoxes. Assume that $\mathcal{I}_1$ and $\mathcal{I}_2$ are interpretations whose domains are split into two non-empty disjoint parts, say $\Delta^{\mathcal{I}_1} = \Delta_{1,1} \uplus \Delta_{1,2}$ and $\Delta^{\mathcal{I}_2} = \Delta_{2,1} \uplus \Delta_{2,2}$ such that

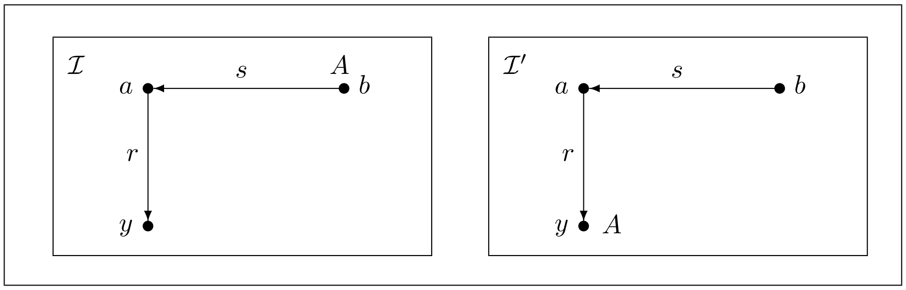

Figure 2: The interpretations $\mathcal{I}$ and $\mathcal{I}'$.

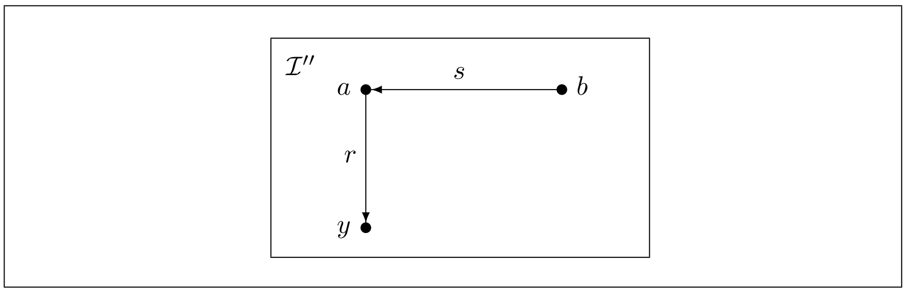

Figure 3: The interpretation $\mathcal{I}''$.

(i) the interpretation of individual names is split in the same way, i.e., $a^{\mathcal{I}_1} \in \Delta_{1,1}$ iff $a^{\mathcal{I}_2} \in \Delta_{2,1}$ for all $a \in \mathsf{N_I}$;

(ii) no role name in $\mathcal{I}_1$ connects $\Delta_{1,1}$ and $\Delta_{1,2}$, i.e., $s^{\mathcal{I}_1} \cap (\Delta_{1,1} \times \Delta_{1,2}) = \emptyset$ and $s^{\mathcal{I}_1} \cap (\Delta_{1,2} \times \Delta_{1,1}) = \emptyset$ for all $s \in \mathsf{N_R}$;

(iii) the same for $\mathcal{I}_2$.

Now swap the submodel of $\mathcal{I}_1$ induced by $\Delta_{1,2}$ with the submodel of $\mathcal{I}_2$ induced by $\Delta_{2,2}$ and use $\mathcal{I}_{1,2}$ to denote the resulting interpretation with domain $\Delta_{1,1} \uplus \Delta_{2,2}$ and $\mathcal{I}_2$ to denote the resulting interpretation with domain $\Delta_{2,1} \uplus \Delta_{1,2}$. It can be proved that

($*$) for all $\mathcal{ALCQIO}$-ABox assertions φ, we have that $\mathcal{I}_1 \models \varphi$ and $\mathcal{I}_2 \models \varphi$ iff $\mathcal{I}_{1,2} \models \varphi$ and $\mathcal{I}_{2,1} \models \varphi$.

Actually, ($*$) is easily verified for (possibly negated) role assertions φ. To establish it for concept assertions, one can prove by induction on the structure of C that for all $d \in \Delta_{i,1}$, $i \in \{1,2\}$, we have $d \in C^{\mathcal{I}_i}$ iff $d \in C^{\mathcal{I}_{i,\bar{i}}}$; and for all $d \in \Delta_{i,2}$, $i \in \{1,2\}$, we have $d \in C^{\mathcal{I}_i}$ iff $d \in C^{\mathcal{I}_{\bar{i},i}}$ (where $\bar{1} = 2$ and $\bar{2} = 1$).

To apply the above observation, we first modify $\mathcal{I}$, $\mathcal{I}'$, and $\mathcal{I}''$ by dropping the s-edge from b to a. Call the resulting interpretations $\mathcal{J}$, $\mathcal{J}'$, and $\mathcal{J}''$, respectively. As s does not occur in $\mathcal{B}$, we have $\mathcal{J} \models \mathcal{B}$ and $\mathcal{J}' \models \mathcal{B}$, and to show $\mathcal{I}'' \models \mathcal{B}$ it suffices to show that $\mathcal{J}'' \models \mathcal{B}$. Observe that $\mathcal{J}''$ is the result of swapping the submodel $\mathcal{J}_b$ of $\mathcal{J}$ induced by the domain $\{b\}$ with the submodel $\mathcal{J}'_b$ of $\mathcal{J}'$ induced by the domain $\{b\}$. Thus, $(*)$ yields that $\mathcal{J}''$ is a model of $\mathcal{B}$ as required. □

4. Computing semantic updates

The main result established in this section is that adding nominals and the '@' constructor to the DLs $\mathcal{ALC}$, $\mathcal{ALCI}$, $\mathcal{ALCQ}$, and $\mathcal{ALCQI}$ suffices to have semantic updates. We also analyze the size of the updated ABoxes showing that our proof incurs a blowup that is exponential in the role depth of concepts used in the original ABox and in the size of the update, both in the case of a single update and of iterated updates. We then show that this blowup is very likely to be unavoidable, and that the somewhat unusual extension of $\mathcal{ALCQIO}$ with Boolean role constructors and 'nominal roles' can be used to avoid the exponential blowup in the role depth of the original ABox (but not the blowup in the size of the update).

4.1. Semantic updates in DLs with nominals and '@'

We provide a detailed proof that $\mathcal{ALCQIO}^{@}$ has semantic updates, thus also approximate updates (and the projective versions of both). The proof can easily be adapted to the fragments $\mathcal{ALCO}^{@}$, $\mathcal{ALCIO}^{@}$, and $\mathcal{ALCQO}^{@}$.

As a preliminary, we observe that assertions that are already contained in the original ABox $\mathcal{A}$ can be dropped from the update $\mathcal{U}$.

Lemma 8. *Let $\mathcal{A}$, $\mathcal{A}'$ be $\mathcal{ALCQIO}^{@}$-ABoxes and $\mathcal{U}$ an update. Then $\mathcal{A} \Longrightarrow_{\mathcal{U}} \mathcal{A}'$ iff $\mathcal{A} \Longrightarrow_{\mathcal{U}\setminus\mathcal{A}} \mathcal{A}'$.*

Proof. Immediate consequence of the fact that for all models $\mathcal{I}$ of $\mathcal{A}$, we have $\mathcal{I}^{\mathcal{U}} = \mathcal{I}^{\mathcal{U}\setminus\mathcal{A}}$. □

Consequently, from now on we assume w.l.o.g. that $\mathcal{A} \cap \mathcal{U} = \emptyset$ whenever $\mathcal{A}$ is updated with $\mathcal{U}$.

Our construction of semantic updates is an extension of the corresponding construction for propositional logic described in [4]. We start with addressing the update of single concept assertions, where the central technical construction consists of converting a concept C into a concept $C^{\mathcal{U}}$ that can be used after the update with $\mathcal{U}$ to describe the set of exactly those domain elements that have been in the extension of C *before* the update. The conversion proceeds by induction on the structure of C as detailed in Figure 4. It seems appropriate to remind the reader that $r^-(a,b) \in \mathcal{U}$ is simply an abbreviation for $r(b,a) \in \mathcal{U}$, and likewise for negated role assertions. To get to grips with the somewhat intricate translation of number restrictions, the reader may find it easier to first consider the more specialized clauses for existential restrictions and universal restrictions, which are for convenience given in Figure 5. As an example, consider the concept $C = \exists r.A$ and the update $\mathcal{U} = \{\neg A(b), r(b,b)\}$. Modulo some minor simplifications, we obtain

$$C^{\mathcal{U}} = \big(\neg\{b\} \sqcap \exists r.(A \sqcup \{b\})\big) \ \sqcup \ \big(\{b\} \sqcap \exists r.(\neg\{b\} \sqcap A)\big).$$

$$
\begin{aligned}
A^{\mathcal{U}} &= (A \sqcup \bigsqcup_{\neg A(a)\in\mathcal{U}} \{a\}) \sqcap \neg(\bigsqcup_{A(a)\in\mathcal{U}} \{a\}) & \{a\}^{\mathcal{U}} &= \{a\} \\
(@_a C)^{\mathcal{U}} &= @_a C^{\mathcal{U}} & (\neg C)^{\mathcal{U}} &= \neg C^{\mathcal{U}} \\
(C \sqcap D)^{\mathcal{U}} &= C^{\mathcal{U}} \sqcap D^{\mathcal{U}} & (C \sqcup D)^{\mathcal{U}} &= C^{\mathcal{U}} \sqcup D^{\mathcal{U}}
\end{aligned}
$$

$$
\begin{aligned}
(\geqslant m\ r\ C)^{\mathcal{U}} = {} & ((\bigsqcap_{a\in\mathsf{Ind}(\mathcal{U})} \neg\{a\}) \sqcap (\geq m\ r\ C^{\mathcal{U}})) \\
& \sqcup \bigsqcup_{a\in\mathsf{Ind}(\mathcal{U})} \Big(\{a\} \sqcap \bigsqcup_{\substack{m_1 + m_2 = m \\ m_2 \leq |\mathsf{Ind}(\mathcal{U})|}} \big((\geq m_1\ r\ (\bigsqcap_{r(a,b)\in\mathcal{U}} \neg\{b\}) \sqcap C^{\mathcal{U}}) \\
& \qquad \sqcap \bigsqcup_{S\subseteq\{b\mid\neg r(a,b)\in\mathcal{U}\},|S|=m_2} \bigsqcap_{b\in S} @_b C^{\mathcal{U}}\big)\Big) \\
(\leqslant m\ r\ C)^{\mathcal{U}} = {} & ((\bigsqcap_{a\in\mathsf{Ind}(\mathcal{U})} \neg\{a\}) \sqcap (\leq m\ r\ C^{\mathcal{U}})) \\
& \sqcup \bigsqcup_{a\in\mathsf{Ind}(\mathcal{U})} \Big(\{a\} \sqcap \bigsqcup_{\substack{m_1 + m_2 = m \\ m_2 \leq |\mathsf{Ind}(\mathcal{U})|}} \big((\leq m_1\ r\ (\bigsqcap_{r(a,b)\in\mathcal{U}} \neg\{b\}) \sqcap C^{\mathcal{U}}) \\
& \qquad \sqcap \bigsqcap_{S\subseteq\{b\mid\neg r(a,b)\in\mathcal{U}\},|S|=m_2+1} \bigsqcup_{b\in S} \neg @_b C^{\mathcal{U}}\big)\Big)
\end{aligned}
$$

Figure 4: Constructing the concept update $C^{\mathcal{U}}$

The following lemma formally states the main property of the constructed concepts $C^{\mathcal{U}}$, where we use $\neg\mathcal{U}$ to denote $\{\dot\neg\varphi \mid \varphi \in \mathcal{U}\}$ and $\dot\neg\varphi$ is obtained from $\neg\varphi$ by eliminating double negation (i.e., it denotes ψ if $\varphi = \neg\psi$ for some ψ and $\neg\varphi$ otherwise). We will see later how to overcome the restriction that $\mathcal{I}$ has to violate all assertions in $\mathcal{U}$.

Lemma 9. *For all interpretations $\mathcal{I}$ such that $\mathcal{I} \models \neg\mathcal{U}$, we have $C^{\mathcal{I}} = (C^{\mathcal{U}})^{\mathcal{I}^{\mathcal{U}}}$.*

Proof. Let $\mathcal{I}$ be an interpretation such that $\mathcal{I} \models \neg\mathcal{U}$ and E an $\mathcal{ALCQIO}^{@}$-concept. By induction on the structure of E, we show that $(E^{\mathcal{U}})^{\mathcal{I}^{\mathcal{U}}} = E^{\mathcal{I}}$.

- If $E = A$, for A a concept name, then $(A^{\mathcal{U}})^{\mathcal{I}^{\mathcal{U}}}$ is

$$
\begin{aligned}
& \Big(A^{\mathcal{I}^{\mathcal{U}}} \cup \bigcup_{\neg A(a)\in\mathcal{U}} \{a^{\mathcal{I}^{\mathcal{U}}}\}\Big) \setminus \bigcup_{A(a)\in\mathcal{U}} \{a^{\mathcal{I}^{\mathcal{U}}}\} \\
&= \Big(\Big(A^{\mathcal{I}} \cup \bigcup_{A(a)\in\mathcal{U}} \{a^{\mathcal{I}}\} \setminus \bigcup_{\neg A(a)\in\mathcal{U}} \{a^{\mathcal{I}}\}\Big) \cup \bigcup_{\neg A(a)\in\mathcal{U}} \{a^{\mathcal{I}^{\mathcal{U}}}\}\Big) \setminus \bigcup_{A(a)\in\mathcal{U}} \{a^{\mathcal{I}^{\mathcal{U}}}\} \\
&= \Big(\Big(A^{\mathcal{I}} \cup \bigcup_{A(a)\in\mathcal{U}} \{a^{\mathcal{I}}\} \setminus \bigcup_{\neg A(a)\in\mathcal{U}} \{a^{\mathcal{I}}\}\Big) \cup \bigcup_{\neg A(a)\in\mathcal{U}} \{a^{\mathcal{I}}\}\Big) \setminus \bigcup_{A(a)\in\mathcal{U}} \{a^{\mathcal{I}}\} \\
&= A^{\mathcal{I}}
\end{aligned}
$$

  The last equality holds since, due to $\mathcal{I} \models \neg\mathcal{U}$, we have $A^{\mathcal{I}} \cap \bigcup_{A(a)\in\mathcal{U}} \{a^{\mathcal{I}}\} = \emptyset$ and $\bigcup_{\neg A(a)\in\mathcal{U}} \{a^{\mathcal{I}}\} \subseteq A^{\mathcal{I}}$.

- The case $E = \{a\}$ is immediate since $\mathcal{I}$ and $\mathcal{I}^{\mathcal{U}}$ interpret individual names in the same way.
- The cases $E = \neg C$, $E = C \sqcup D$, $E = C \sqcap D$, and $E = @_a C$ are straightforward using the semantics and induction hypothesis.
- It remains to consider the cases $E = (\geq m\ r\ C)$ and $E = (\leq m\ r\ C)$. The central observation is that we have $(d, e) \in r^{\mathcal{I}}$ iff either
  - $d \neq a^{\mathcal{I}}$ for all $a \in \mathsf{Ind}(\mathcal{U})$ and $(d, e) \in r^{\mathcal{I}^{\mathcal{U}}}$; or
  - $d = a^{\mathcal{I}}$ for an $a \in \mathsf{Ind}(\mathcal{U})$ and exactly one of the following holds:
    * $e \neq b^{\mathcal{I}}$ for all $r(a, b) \in \mathsf{Ind}(\mathcal{U})$ and $(d, e) \in r^{\mathcal{I}^{\mathcal{U}}}$
    * $e = b^{\mathcal{I}}$ for a $b \in \mathsf{Ind}(\mathcal{U})$ with $\neg r(a, b) \in \mathcal{U}$ (which implies $(d, e) \in r^{\mathcal{I}}$ since $\mathcal{I} \models \neg\mathcal{U}$ and excludes the previous case since $(d, e) \notin r^{\mathcal{I}^{\mathcal{U}}}$).

  This case distinction is directly reflected in the translation of the concepts $(\geqslant m\ r\ C)$ and $(\leqslant m\ r\ C)$. Using this and the induction hypothesis, it is possible to verify that $(E^{\mathcal{U}})^{\mathcal{I}^{\mathcal{U}}} = E^{\mathcal{I}}$, as required.

$\square$

The concepts $C^{\mathcal{U}}$ are used as a central building block for defining updates of ABoxes. Let $\mathcal{A}$ be an $\mathcal{ALCQIO}^{@}$-ABox and $\mathcal{U}$ an update. Define the ABox $\mathcal{A}^{\mathcal{U}}$ by setting

$$\begin{array}{rcl} \mathcal{A}^{\mathcal{U}} & = & \{C^{\mathcal{U}}(a) \mid C(a) \in \mathcal{A}\} \cup \\ & & \{r(a, b) \mid r(a, b) \in \mathcal{A} \wedge \neg r(a, b) \notin \mathcal{U}\} \cup \\ & & \{\neg r(a, b) \mid \neg r(a, b) \in \mathcal{A} \wedge r(a, b) \notin \mathcal{U}\}. \end{array}$$

We now establish an analogue of Lemma 9, but formulated for ABoxes instead of concepts.

Lemma 10. *Let $\mathcal{A}$ be an ABox and $\mathcal{U}$ an update. For every interpretation $\mathcal{I}$ with $\mathcal{I} \models \neg\mathcal{U}$, we have $\mathcal{I} \models \mathcal{A}$ iff $\mathcal{I}^{\mathcal{U}} \models \mathcal{A}^{\mathcal{U}}$.*

Proof. "$\Rightarrow$" Let $\mathcal{I} \models \mathcal{A}$. We show that $\mathcal{I}^{\mathcal{U}} \models \mathcal{A}^{\mathcal{U}}$. Let $\varphi \in \mathcal{A}^{\mathcal{U}}$. If $\varphi = r(a, b)$ or $\varphi = \neg r(a, b)$, then, by the definition of $\mathcal{A}^{\mathcal{U}}$ and $\mathcal{I}^{\mathcal{U}}$, $\mathcal{I}^{\mathcal{U}} \models \varphi$. If $\varphi = E^{\mathcal{U}}(a)$ for $E(a) \in \mathcal{A}$, Lemma 9 yields $\mathcal{I}^{\mathcal{U}} \models E^{\mathcal{U}}(a)$.

"$\Leftarrow$" Let $\mathcal{I}^{\mathcal{U}} \models \mathcal{A}^{\mathcal{U}}$. We show that $\mathcal{I} \models \mathcal{A}$. Take $\varphi \in \mathcal{A}$. First for the case $\varphi = r(a, b)$. There are two subcases:

1. $\neg r(a, b) \in \mathcal{U}$. Then $r(a, b) \in \neg\mathcal{U}$ and since $\mathcal{I} \models \neg\mathcal{U}$, we obtain that $\mathcal{I} \models r(a, b)$;
2. $\neg r(a, b) \notin \mathcal{U}$. Then $r(a, b) \in \mathcal{A}^{\mathcal{U}}$, thus $\mathcal{I}^{\mathcal{U}} \models r(a, b)$. We have $r(a, b) \notin \mathcal{U}$ since we assume $\mathcal{A} \cap \mathcal{U} = \emptyset$. By definition of $\mathcal{I}^{\mathcal{U}}$, this yields $\mathcal{I} \models r(a, b)$.

The case $\varphi = \neg r(a, b)$ is analogous to the previous one, and the case $\varphi = E(a)$ is immediate by Lemma 9. $\square$

$$(\exists r.C)^{\mathcal{U}} = ((\bigsqcap_{a \in \mathsf{Ind}(\mathcal{U})} \neg\{a\}) \sqcap \exists r.C^{\mathcal{U}}) \sqcup$$
$$\bigsqcup_{a \in \mathsf{Ind}(\mathcal{U})} \big(\{a\} \sqcap (\exists r.((\bigsqcap_{r(a,b) \in \mathcal{U}} \neg\{b\}) \sqcap C^{\mathcal{U}}) \sqcup \bigsqcup_{\neg r(a,b) \in \mathcal{U}} @_b C^{\mathcal{U}})\big)$$
$$(\forall r.C)^{\mathcal{U}} = ((\bigsqcap_{a \in \mathsf{Ind}(\mathcal{U})} \neg\{a\}) \rightarrow \forall r.C^{\mathcal{U}}) \sqcap$$
$$\bigsqcap_{a \in \mathsf{Ind}(\mathcal{U})} \big(\{a\} \rightarrow (\forall r.((\bigsqcap_{r(a,b) \in \mathcal{U}} \neg\{b\}) \rightarrow C^{\mathcal{U}}) \sqcap \bigsqcap_{\neg r(a,b) \in \mathcal{U}} @_b C^{\mathcal{U}})\big)$$

Figure 5: Constructing $C^{\mathcal{U}}$ for existential and universal restrictions

Similar to the concepts $C^{\mathcal{U}}$, the construction of the ABox $\mathcal{A}^{\mathcal{U}}$ relies on the fact that the model $\mathcal{I}$ of $\mathcal{A}$ violates all assertions in $\mathcal{U}$. For a fixed model $\mathcal{I}$ of $\mathcal{A}$, we can overcome this problem by replacing $C^{\mathcal{U}}$ with $C^{\mathcal{U}'}$, where $\mathcal{U}' = \{\varphi \in \mathcal{U} \mid \mathcal{I} \not\models \varphi\}$ is the update that consists of those assertions from $\mathcal{U}$ that are violated in $\mathcal{I}$. However, the original ABox $\mathcal{A}$ can have many different models $\mathcal{I}$, which may give rise to different adjustments $\mathcal{U}'$ of the update $\mathcal{U}$. We address this issue by considering *all* subsets $\mathcal{U}' \subseteq \mathcal{U}$ of assertions that can potentially be violated in a model of $\mathcal{A}$, and then taking the disjunction of all the resulting updated ABoxes $\mathcal{A}^{\mathcal{U}'}$.

Let $\mathcal{A}$ be an ABox and $\mathcal{U}$ an update. Define the updated ABox $\mathcal{A}'$ as the Boolean ABox

$$\mathcal{A}' \;=\; \bigwedge \mathcal{U} \wedge \bigvee_{\mathcal{U}' \subseteq \mathcal{U}} \bigwedge \mathcal{A}^{\mathcal{U}'}.$$

Here, we use Boolean ABox operators only as an abbreviation for the "@" constructor, see Lemma 6.

Lemma 11. *$\mathcal{A} \Longrightarrow_{\mathcal{U}} \mathcal{A}'$.*

Proof. We prove that both inclusions of $M(\mathcal{A}') = \{\mathcal{I}^{\mathcal{U}} \mid \mathcal{I} \in M(\mathcal{A})\}$ hold.

"$\supseteq$" Let $\mathcal{I} \models \mathcal{A}$. We have to show that $\mathcal{I}^{\mathcal{U}} \models \mathcal{A}'$. By definition of $\mathcal{I}^{\mathcal{U}}$, $\mathcal{I}^{\mathcal{U}} \models \mathcal{U}$. Define a subset $\mathcal{U}' \subseteq \mathcal{U}$ as $\mathcal{U}' = \{\varphi \in \mathcal{U} \mid \mathcal{I} \not\models \varphi\}$. By Lemma 10, we have $\mathcal{I}^{\mathcal{U}'} \models \mathcal{A}^{\mathcal{U}'}$. Moreover, using the definition of $\mathcal{U}'$ and $\mathcal{I}^{\mathcal{U}'}$, it can be verified that $\mathcal{I}^{\mathcal{U}'} = \mathcal{I}^{\mathcal{U}}$, thus $\mathcal{I}^{\mathcal{U}} \models \mathcal{A}^{\mathcal{U}'}$ which yields $\mathcal{I}^{\mathcal{U}} \models \mathcal{A}'$.

"$\subseteq$" Let $\mathcal{I}' \models \mathcal{A}'$. We need to show that there exists an interpretation $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{A}$ and $\mathcal{I}' = \mathcal{I}^{\mathcal{U}}$. Since $\mathcal{I}' \models \mathcal{A}'$, there is a $\mathcal{U}' \subseteq \mathcal{U}$ such that $\mathcal{I}' \models \mathcal{A}^{\mathcal{U}'}$. Let $\mathcal{I} = (\mathcal{I}')^{\neg(\mathcal{U}')}$, i.e., we undo all the modifications in the selected adjustment $\mathcal{U}'$. Then $\mathcal{I}' = \mathcal{I}^{\mathcal{U}'} = \mathcal{I}^{\mathcal{U}}$. Moreover, Lemma 10 yields $\mathcal{I} \models \mathcal{A}$. $\square$

The presented construction of semantic updates can be adapted to the DLs $\mathcal{ALCO}^{@}$, $\mathcal{ALCIO}^{@}$, and $\mathcal{ALCQO}^{@}$. For the former two, we have to treat existential and universal restrictions in the $C^{\mathcal{U}}$ translation rather than number restrictions. The corresponding clauses are shown in Figure 5. The lemmas proved above for $\mathcal{ALCQIO}^{@}$ are then easily adapted.

Theorem 3. *The DLs $\mathcal{ALCO}^{@}$, $\mathcal{ALCIO}^{@}$, $\mathcal{ALCQO}^{@}$, and $\mathcal{ALCQIO}^{@}$ have semantic updates.*

4.2. The size of semantic updates

We show that the above construction yields semantic updates whose size is at most exponential in the size of the role depth of the original ABox and the size of the update. If the role depth of the original ABoxes is fixed, then the size of the update is polynomial in the size of the original ABox and exponential in the size of the update.

The *length* of a concept C, denoted by $|C|$, is the number of symbols needed to write C. Numbers inside number restrictions can be coded in unary or in binary, which yields $|(\leqslant\ n\ r\ C)| \in \mathcal{O}(n)$ and $|(\leqslant\ n\ r\ C)| \in \mathcal{O}(\log n)$, respectively. Since all our results hold independently of the chosen coding scheme, we from now on assume binary coding. The *role depth* of a concept C, denoted by $\mathsf{rd}(C)$, is the nesting depth of number restrictions in C, i.e., $\mathsf{rd}(A) = 0$, $\mathsf{rd}(\neg C) = \mathsf{rd}(C)$, $\mathsf{rd}(C \sqcap D) = \mathsf{rd}(C \sqcup D) = \mathsf{max}(\mathsf{rd}(C), \mathsf{rd}(D)) + 1$, and $\mathsf{rd}(\geqslant n\ r\ C) = \mathsf{rd}(\leqslant n\ r\ C) = \mathsf{rd}(C) + 1$.

The *size* of an ABox assertion $C(a)$ is $|C|$, the size of $r(a, b)$ and $\neg r(a, b)$ is 1. The *size* of an ABox $\mathcal{A}$, denoted by $|\mathcal{A}|$, is the sum of the sizes of all assertions in $\mathcal{A}$. The *role depth* of an ABox $\mathcal{A}$, denotes by $\mathsf{rd}(\mathcal{A})$, is $\mathsf{max}\{\mathsf{rd}(C) \mid C(a) \in \mathcal{A}\}$.

A close inspection of our construction of semantic updates reveals the following result.

Theorem 4. *Let $\mathcal{L} \in \{\mathcal{ALCO}^{@}, \mathcal{ALCIO}^{@}, \mathcal{ALCQO}^{@}, \mathcal{ALCQIO}^{@}\}$. Then for every $\mathcal{L}$-ABox $\mathcal{A}$ and update $\mathcal{U}$, the semantic update $\mathcal{A}'$ of $\mathcal{A}$ with $\mathcal{U}$ computed by our algorithm satisfies*

$$|\mathcal{A}'| \leq |\mathcal{A}| \cdot 2^{\mathcal{O}(\log(|\mathcal{A}|)\cdot|\mathcal{U}|\cdot\mathsf{rd}(\mathcal{A}))} \cdot 2^{2|\mathcal{U}|}$$

and can be computed in time polynomial in $|\mathcal{A}'|$.

Proof. By inspection of the construction given in Section 4.1. We distinguish two cases. First, assume that $\mathsf{rd}(\mathcal{A}) = 0$. Then the size of each concept $C^{\mathcal{U}}$ is $|C| \cdot |\mathcal{U}|$ and each ABox $\mathcal{A}^{\mathcal{U}'}$ is of size at most $|\mathcal{A}| \cdot |\mathcal{U}|$. The semantic update $\mathcal{A}'$ comprises $2^{|\mathcal{U}|}$ such ABoxes $\mathcal{A}^{\mathcal{U}'}$ plus the update $\mathcal{U}$, thus the overall size is clearly dominated by the given expression. Now assume that $\mathsf{rd}(\mathcal{A}) > 0$. View $C^{\mathcal{U}}$ as a syntax tree in which all purely Boolean subtrees are collapsed into a single node, where a subtree is Boolean if none of its nodes is labeled with the '@' constructor or a number restriction. By construction, it follows that $C^{\mathcal{U}}$ has outdegree at most $|C| \cdot 2^{\mathcal{O}(|\mathcal{U}|)}$ and depth at most $\mathsf{rd}(C)$. Since every collapsed node represents at most $|C| \cdot 2^{\mathcal{O}(|\mathcal{U}|)}$ syntax tree nodes, the size of $C^{\mathcal{U}}$ is bounded by $2^{\mathcal{O}(\log(|C|)\cdot|\mathcal{U}|\cdot\mathsf{rd}(C))}$. Analogously, the size of each ABox $\mathcal{A}^{\mathcal{U}'}$ is bounded by $|\mathcal{A}| \cdot 2^{\mathcal{O}(\log(|\mathcal{A}|)\cdot|\mathcal{U}|\cdot\mathsf{rd}(\mathcal{A}))}$. The semantic update $\mathcal{A}'$ comprises $2^{|\mathcal{U}|}$ such ABoxes $\mathcal{A}^{\mathcal{U}'}$ plus the update $\mathcal{U}$, thus the overall size is again dominated by the given expression. □

Note that the bound stated in Theorem 4 is polynomial in the overall size of the original ABox (which is potentially large), and exponential only in the role depth of the original ABox and the size of the update, which are typically small. In particular, if the input ABox does not comprise any number restrictions (and neither existential and universal restrictions), then the size of updated ABoxes is exactly as in propositional logic [9].

In many applications, the state of affairs evolves continuously which makes it necessary to update the ABox over and over again. It is then clearly important that the exponential blowups of the repeated updates do not add up, which would result in a non-elementary growth of the produced semantic updates. The following theorem shows that this is indeed not the case.

Theorem 5. *Let* $\mathcal{L} \in \{\mathcal{ALCO}^{@}, \mathcal{ALCIO}^{@}, \mathcal{ALCQO}^{@}, \mathcal{ALCQIO}^{@}\}$*,* $\mathcal{A}_0, \dots, \mathcal{A}_n$ $\mathcal{L}$*-ABoxes,* $\mathcal{U}_1, \dots, \mathcal{U}_n$ *updates, and* $\mathcal{A}_{i+1}$ *the semantic update of* $\mathcal{A}_i$ *with* $\mathcal{U}_{i+1}$ *computed by our algorithm, for* $0 \leq i < n$*. Then*

$$|\mathcal{A}_n| \leq |\mathcal{A}_0| \cdot 2^{\mathcal{O}(\log(|\mathcal{A}_0|)\cdot(|\mathcal{U}_1|+\cdots+|\mathcal{U}_n|)\cdot \mathsf{rd}(\mathcal{A}_0))} \cdot 2^{2(|\mathcal{U}_1|+\cdots+|\mathcal{U}_n|)}.$$

Proof. The argument is analogous to the proof of Theorem 4. In particular, viewing a concept $((C^{\mathcal{U}_1})^{\cdots})^{\mathcal{U}_n}$ as a syntax tree with collapsed nodes as in that proof, it is not hard to see that the outdegree is at most $|C| \cdot 2^{\mathcal{O}(|\mathcal{U}_1|+\cdots+|\mathcal{U}_n|)}$ and the depth is at most $\mathsf{rd}(C)$. Since every collapsed Boolean node represents at most $\mathcal{O}(|C| \cdot 2^{\mathcal{O}(|\mathcal{U}_1|+\cdots+|\mathcal{U}_n|)})$ syntax tree nodes, the size of $C^{\mathcal{U}}$ is bounded by $2^{\mathcal{O}(\log(|C|)\cdot(|\mathcal{U}_1|+\cdots+|\mathcal{U}_n|)\cdot\mathsf{rd}(C))}$. Analogously, the size of each ABox $((\mathcal{A}_0^{\mathcal{U}'_1})^{\cdots})^{\mathcal{U}'_n}$, with $\mathcal{U}'_1 \subseteq \mathcal{U}_1, \dots, \mathcal{U}'_n \subseteq \mathcal{U}_n$, is bounded by $|\mathcal{A}_0| \cdot 2^{\mathcal{O}(\log(|\mathcal{A}_0|)\cdot(|\mathcal{U}_1|+\cdots+|\mathcal{U}_n|)\cdot\mathsf{rd}(\mathcal{A}_0))}$. The semantic update $\mathcal{A}_n$ is a Boolean combination of $2^{|\mathcal{U}_1|+\cdots+|\mathcal{U}_n|}$ such ABoxes and $2^{|\mathcal{U}_2|+\cdots+|\mathcal{U}_n|}$ ABoxes of the form $((\mathcal{U}_i^{\mathcal{U}'_{i+1}})^{\cdots})^{\mathcal{U}'_n}$. The size of each of the latter ABoxes is bounded by $|\mathcal{U}_1| \cdot \cdots \cdot |\mathcal{U}_n|$. In summary, the size of $\mathcal{A}_n$ is thus bounded by the expression given in Theorem 5. □

4.3. A lower bound for the size of semantic updates

We show that, in $\mathcal{ALCQIO}^{@}$ and its fragments, an exponential blowup of semantic updates cannot be avoided unless

$$\text{NP} \cap \text{co-NP} \subseteq \text{NC}_1,$$

where NC_1 is the class of problems that is solvable by a family of circuits of polynomial size and logarithmic depth. We work with the non-uniform version of NC_1 here, i.e., we do not demand that the circuit for a given input length can be computed within certain resource bounds (or at all!). The stated inclusion is widely believed to not hold. It is intimately related to the important open question whether every problem that is efficiently solvable can be effectively parallelized, i.e., whether PTime is a subset of uniform NC_1, which is also not believed to be the case. In particular, there are reasons to assume that (non)-uniformity is irrelevant for inclusions of this sort, see e.g. [19].

We obtain our result by relating semantic updates to Craig interpolants in propositional logic, which allows us to transfer known lower bounds for the size of such interpolants, see e.g. [20, 21, 22]. In what follows, we will deliberately confuse propositional formulas and concepts that use only the Boolean constructors $\neg$, $\sqcap$, and $\sqcup$. We use $\mathsf{sig}(\varphi)$ to denote the set of propositional letters that occur in the propositional formula φ, and $\mathsf{sig}(\mathcal{A})$ to denote the concept and role names that occur in the ABox $\mathcal{A}$. Recall that, given two propositional formulas φ and ψ with $\varphi \models \psi$, a *Craig interpolant* of φ and ψ is a formula ϑ with $\varphi \models \vartheta \models \psi$ and such that $\mathsf{sig}(\vartheta) \subseteq \mathsf{sig}(\varphi) \cap \mathsf{sig}(\psi)$. For a propositional formula φ and a set of propositional letters S, we use $\varphi[S/\top]$ to denote the result of replacing each letter from S in φ with $\top$ (i.e., logical truth).

Lemma 12. *Let φ, ψ, and ϑ be propositional formulas with $\varphi \models \psi$. If $\mathcal{A} = \{\varphi(a)\}$, $\mathcal{U} = \{p(a) \mid p \in \mathsf{sig}(\varphi) \setminus \mathsf{sig}(\psi)\}$, and $\mathcal{A} \Longrightarrow_{\mathcal{U}} \{\vartheta(a)\}$, then $\vartheta[S/\top]$ is a Craig interpolant of φ and ψ where $S = \mathsf{sig}(\vartheta) \setminus (\mathsf{sig}(\varphi) \cap \mathsf{sig}(\psi))$.*

Proof. Clearly, $\vartheta[S/\top]$ contains only propositional letters from $\mathsf{sig}(\varphi) \cap \mathsf{sig}(\psi)$ as required. It thus remains to show that $\varphi \models \vartheta[S/\top] \models \psi$. We start with noting that

$$(*) \quad \vartheta[S/\top] \equiv \vartheta[\mathsf{sig}(\mathcal{U})/\top]$$

To see $(*)$, note that $(S \setminus \mathsf{sig}(\mathcal{U})) \cap (\mathsf{sig}(\mathcal{A}) \cup \mathsf{sig}(\mathcal{U})) = \emptyset$, and thus $\mathcal{A} \Longrightarrow_{\mathcal{U}} \{\vartheta(a)\}$ implies that whenever $\mathcal{I}$ is a model of ϑ and $\mathcal{I}$ and $\mathcal{J}$ differ only in the interpretation of symbols from $S \setminus \mathsf{sig}(\mathcal{U})$, then $\mathcal{J}$ is also a model of ϑ. It follows that $\vartheta[\mathsf{sig}(\mathcal{U})/\top]$ has the same property. Thus, replacing all symbols from $S \setminus \mathsf{sig}(\mathcal{U})$ in $\vartheta[\mathsf{sig}(\mathcal{U})/\top]$ with $\top$, which yields $\vartheta[S/\top]$, is an equivalence preserving operation.

We now show that $\varphi \models \vartheta[S/\top] \models \psi$.

- $\varphi \models \vartheta[S/\top]$.

  Let $\mathcal{I} \models \varphi(a)$. Then $\mathcal{I}^{\mathcal{U}} \models \vartheta(a)$. Moreover, $\mathcal{I}^{\mathcal{U}} \models p(a)$ for all $p \in \mathsf{sig}(\mathcal{U})$. It follows that $\mathcal{I}^{\mathcal{U}} \models \vartheta[\mathsf{sig}(\mathcal{U})/\top](a)$, thus $\mathcal{I}^{\mathcal{U}} \models \vartheta[S/\top](a)$ by $(*)$. Since $\mathcal{I}$ and $\mathcal{I}^{\mathcal{U}}$ agree on the interpretation of all propositional letters from $\mathsf{sig}(\varphi) \cap \mathsf{sig}(\psi)$, we get $\mathcal{I} \models \vartheta[S/\top](a)$.

- $\vartheta[S/\top] \models \psi$.

  Let $\mathcal{J} \models \vartheta[S/\top](a)$. By $(*)$, $\mathcal{J} \models \vartheta[\mathsf{sig}(\mathcal{U})/\top](a)$. Let $\mathcal{J}'$ be obtained from $\mathcal{J}$ by interpreting all $p \in \mathsf{sig}(\mathcal{U})$ as true, i.e., $p^{\mathcal{J}'} = \Delta^{\mathcal{J}'}$. Clearly, $\mathcal{J}' \models \vartheta(a)$. Thus there is a model $\mathcal{I}$ of $\mathcal{A}$ with $\mathcal{I}^{\mathcal{U}} = \mathcal{J}'$. Since $\mathcal{I} \models \varphi(a)$, we have $\mathcal{I} \models \psi(a)$. As $\mathcal{I}$, $\mathcal{J}'$, and $\mathcal{J}$ agree on the interpretation of all propositional letters from $\mathsf{sig}(\psi)$, we get $\mathcal{J} \models \psi(a)$.

$\square$

As a side remark, we note that the formula $\vartheta[S/\top]$ in Lemma 12 is even a uniform interpolant for φ and $\mathsf{sig}(\psi)$: for any ψ' with $\mathsf{sig}(\psi') \cap \mathsf{sig}(\varphi) \subseteq \mathsf{sig}(\psi)$ and $\varphi \models \psi'$ we have $\vartheta[S/\top] \models \psi'$.

The semantic update $\{\vartheta(a)\}$ considered in Lemma 12 is of a rather particular form. To establish a lower bound for the size of semantic updates in $\mathcal{ALCQIO}^@$, we observe that one cannot express ABoxes of the form $\{\vartheta(a)\}$, ϑ a propositional formula, more succinctly in $\mathcal{ALCQIO}^@$. The following lemma states this fact even for first-order logic (with equality), of which $\mathcal{ALCQIO}^@$ is a fragment when concept names are confused with unary predicates and role names with binary predicates [23]. The lemma can easily be proved by standard manipulations of FO formulas; details are left to the reader.

Lemma 13. *Let $\vartheta(a)$ be an ABox assertion, where ϑ is a propositional formula, and let φ be a first-order sentence that is equivalent to $\vartheta(a)$. Then there exists a propositional formula ϑ' that is equivalent to ϑ such that $|\vartheta'(a)| \leq |\varphi|$.*

Together with the observation from [20] that the size of Craig interpolants cannot be bounded by a polynomial unless $\mathrm{NP} \cap \text{co-NP} \subseteq \mathrm{NC}_1$, Lemma 12 and 13 yield the desired result.

Theorem 6. *If there exists a polynomial p such that, for all propositional ABoxes $\mathcal{A}$ and updates $\mathcal{U}$, there exists an $\mathcal{ALCQIO}^@$-ABox $\mathcal{A}'$ such that $\mathcal{A} \Longrightarrow_{\mathcal{U}} \mathcal{A}'$ and $|\mathcal{A}'| \leq p(|\mathcal{A}| \cdot |\mathcal{U}|)$, then $\mathrm{NP} \cap \text{co-NP} \subseteq \mathrm{NC}_1$.*

Proof. Assume there is a polynomial p as stated in Theorem 6. We show that then p^2 bounds the size of Craig interpolants in propositional logic, which implies $\mathrm{NP} \cap \text{co-NP} \subseteq \mathrm{NC}_1$ as observed in [20]. Let φ and ψ be propositional formulas and take the ABox $\mathcal{A} = \{\varphi(a)\}$ and update $\mathcal{U} = \{p(a) \mid p \in \mathsf{sig}(\varphi) \setminus \mathsf{sig}(\psi)\}$. Then there is an $\mathcal{ALCQIO}^@$-ABox $\mathcal{A}'$ with $\mathcal{A} \Longrightarrow_{\mathcal{U}} \mathcal{A}'$ and $|\mathcal{A}'| \leq p(|\mathcal{A}| \cdot |\mathcal{U}|)$. By our algorithm computing semantic updates, there is a propositional formula ϑ such that $\mathcal{A} \Longrightarrow_{\mathcal{U}} \{\vartheta(a)\}$. By Lemma 1, $\mathcal{A}'$ and $\{\vartheta(a)\}$ are logically equivalent. By Lemma 13, $\mathcal{A}'$ is logically equivalent to an ABox $\mathcal{A}'' = \{\vartheta'(a)\}$ with ϑ' a propositional formula and $|\mathcal{A}''| \leq |\mathcal{A}'|$. Finally, by Lemma 12 this implies that $\vartheta'[S/\top]$, $S = \mathsf{sig}(\vartheta') \setminus (\mathsf{sig}(\varphi) \cap \mathsf{sig}(\psi))$, is a Craig interpolant of φ and ψ, whose size is bounded by $p(|\varphi|^2)$ (and independent of $|\psi|$). $\square$

As shown in [22], it is possible to replace the complexity-theoretic proviso in Theorem 6 with $\mathrm{UP} \subseteq \mathrm{P}/_{\mathsf{poly}}$, where UP is the class of problems in NP accepted by a nondeterministic Turing machine with unique accepting paths and $\mathrm{P}/_{\mathsf{poly}}$ is the non-uniform version of PTime. Just like the statement used in Theorem 6, it is strongly believed that $\mathrm{UP} \subseteq \mathrm{P}/_{\mathsf{poly}}$ does not hold.

We note that the result stated as Theorem 6 is closely related to a similar result proved by Cadoli et al. which states that for semantic updates of propositional theories, an exponential blowup cannot be avoided unless the polynomial hierarchy collapses [9]. However, Cadoli et al.'s technique does not appear to work with the restricted form of updates $\mathcal{U}$ considered in this paper, where we allow only literals but no compound concepts/formulas.

4.4. Smaller updates in $\mathcal{ALCQIO}^+$

An inspection of the construction of semantic updates presented in Section 4.1 reveals that, in the case where the update $\mathcal{U}$ contains only concept assertions but no role assertions, computing the concepts $C^{\mathcal{U}}$ becomes a lot simpler: we only have to replace every concept name A in C with

$$(A \sqcup \bigsqcup_{\neg A(a)\in\mathcal{U}} \{a\}) \sqcap \neg(\bigsqcup_{A(a)\in\mathcal{U}} \{a\}).$$

In particular, the resulting semantic update $\mathcal{A}^{\mathcal{U}}$ is then only exponential in $|\mathcal{U}|$, but no longer in the role depth of $\mathcal{A}$. To understand why such a simple rewriting is not possible when roles are updated, note that the above construction makes essential use of nominals, the '@' constructor, and the Boolean concept constructors. In standard DLs, none of these constructors is available for roles: we can neither construct the union of roles, nor their complement, nor a "nominal role" $\{(a, b)\}$ with a and b individual names. In this section, we consider a DL that comprises such slightly unusual role constructors and show that it admits simple semantic updates of the above form also in the case when the update comprises role assertions.

Denote by $\mathcal{ALCQIO}^+$ the DL that extends $\mathcal{ALCQIO}^@$ by means of the role constructors $\cup$ (role union), $\setminus$ (set-theoretic difference of roles), and $\{(a, b)\}$ (nominal roles). In this language, compound roles are constructed by starting from role names and nominal roles, and then applying $\cup$, $\setminus$, and the inverse role constructor $\cdot^-$. The semantics of compound roles is as expected:

- $\{(a, b)\}^{\mathcal{I}} = \{(a^{\mathcal{I}}, b^{\mathcal{I}})\}$, for all $a, b \in \mathsf{N_I}$;
- $(r_1 \cup r_2)^{\mathcal{I}} = r_1^{\mathcal{I}} \cup r_2^{\mathcal{I}}$;
- $(r_1 \setminus r_2)^{\mathcal{I}} = r_1^{\mathcal{I}} \setminus r_2^{\mathcal{I}}$.

We note that $\mathcal{ALCQIO}^+$ is of almost the same expressive power as C^2, the two-variable fragment of first-order logic with counting quantifiers [24]. In particular, $\mathcal{ALCQIO}^+$-ABoxes can easily be translated into formulas of C^2. The following is the main result of this section.

Theorem 7. *There is a polynomial p such that, for every $\mathcal{ALCQIO}^+$-ABox $\mathcal{A}$ and every update $\mathcal{U}$, there is an $\mathcal{ALCQIO}^+$-ABox $\mathcal{A}'$ such that*

1. $\mathcal{A} \Longrightarrow^{\mathcal{U}} \mathcal{A}'$*;*
2. $|\mathcal{A}'| \leq |\mathcal{A}| \cdot 2^{p(|\mathcal{U}|)}$*;*
3. $\mathcal{A}'$ *can be computed in time* $p(|\mathcal{A}'|)$*.*

Proof. We modify the construction from Section 4.1. The construction of the concepts $C^{\mathcal{U}}$ is now as follows: replace every concept name A in C with

$$(A \sqcup \bigsqcup_{\neg A(a)\in\mathcal{U}} \{a\}) \sqcap \neg(\bigsqcup_{A(a)\in\mathcal{U}} \{a\})$$

and every role name r in C with

$$(r \cup \bigcup_{\neg r(a,b)\in\mathcal{U}} \{(a,b)\}) \setminus \bigcup_{r(a,b)\in\mathcal{U}} \{(a,b)\}.$$

The concepts $C^{\mathcal{U}}$ are thus of size polynomial in $|\mathcal{A}| \cdot |\mathcal{U}|$. The ABox $\mathcal{A}'$ can then be constructed in the same way as in Section 4.1. $\square$

Clearly, Theorem 7 is independent of the coding of numbers, and iterated updates retain the same size bound, with $|\mathcal{U}|$ replaced by $|\mathcal{U}_1| + \cdots + |\mathcal{U}_n|$. An alternative to working with a description logic such as $\mathcal{ALCQIO}^+$ is to work directly with the two-variable fragment with counting C^2. Then, a result analogous to Theorem 7 is easily obtained.

5. Computing projective updates

We consider projective updates and show that they are more well-behaved than semantic ones: first, projective updates are enjoyed by all DLs between $\mathcal{ALCO}$ and $\mathcal{ALCQIO}^{@}$, including those that do not comprise the '@' constructor; and second, projective updates can be constructed in polynomial time and without an exponential blowup. We prove this in detail for $\mathcal{ALCQIO}^{@}$ using an approach that can easily be adapted to all DLs between $\mathcal{ALCO}^{@}$ and $\mathcal{ALCQIO}^{@}$. These results then transfer to the corresponding DLs without the '@' constructor thanks to Lemma 7.

Let $\mathcal{A}$ be an $\mathcal{ALCQIO}^{@}$-ABox and $\mathcal{U}$ an update. We show how to construct an $\mathcal{ALCQIO}^{@}$-ABox $\mathcal{A}'$ such that $\mathcal{A} \Longrightarrow^{\mathsf{p}}_{\mathcal{U}} \mathcal{A}'$. Let $\mathsf{sub}(\mathcal{A})$ denote the closure under subconcepts of $\{C \mid C(a) \in \mathcal{A}\}$. The general approach to constructing $\mathcal{A}'$ shares a lot of similarity with the construction of semantic updates in Section 4.1. However, we need some subtle technical tricks to avoid the exponential blowups that occur there, namely (i) during the construction of the concepts $C^{\mathcal{U}}$ and (ii) due to the final disjunction over all $\mathcal{U}' \subseteq \mathcal{U}$. The central idea to overcome both blowups is to use fresh concept names X_C and fresh role names ρ_r to *explicitly* reconstruct in $\mathcal{A}'$ the extension of *all* concepts $C \in \mathsf{sub}(\mathcal{A})$ and *all* roles name $r \in \mathsf{role}(\mathcal{A})$ *before* the update. This allows us to eliminate blowup (i) because the concept names X_C enable 'structure sharing', thus addressing the multiple occurrences of concepts $C^{\mathcal{U}}$ on the right-hand side of the clauses in Figure 4; moreover, the role names ρ_r help to avoid the exponential case distinction 'for all $S \subseteq \{b \mid \neg r(a,b) \in \mathcal{U}\}$' in the clauses for number restrictions. The use of the X_C and ρ_r also allows us to eliminate blowup (ii) as the case distinction 'for all $\mathcal{U}' \subseteq \mathcal{U}$' can be replaced with some 'freedom' that we will leave in the interpretation of the X_C and ρ_r, and that intuitively corresponds to an existential quantification over all $\mathcal{U}' \subseteq \mathcal{U}$. More details are given below after the construction of $\mathcal{A}'$. In what follows, we use ρ_{r^-} to denote $(\rho_r)^-$.

The projective update $\mathcal{A}'$ will be the union of four ABoxes. First, $\mathcal{A}'$ contains the update $\mathcal{U}$. Second, we set up an ABox $\mathcal{A}_{\mathsf{init}}$ that stores the original ABox

$\mathcal{A}$ using the concept names X_C for C and role names ρ_r for r:

$$\begin{array}{rcl}\mathcal{A}_{\mathsf{init}} & = & \{X_C(a) \mid C(a) \in \mathcal{A}\} \cup \\ & & \{\rho_r(a,b) \mid a,b \in \mathsf{Ind}(\mathcal{U}), r(a,b) \in \mathcal{A}\} \cup \\ & & \{\neg\rho_r(a,b) \mid a,b \in \mathsf{Ind}(\mathcal{U}), \neg r(a,b) \in \mathcal{A}\} \cup \\ & & \{r(a,b) \mid \{a,b\} \not\subseteq \mathsf{Ind}(\mathcal{U}), r(a,b) \in \mathcal{A}\} \cup \\ & & \{\neg r(a,b) \mid \{a,b\} \not\subseteq \mathsf{Ind}(\mathcal{U}), \neg r(a,b) \in \mathcal{A}\}\end{array}$$

The remaining two ABoxes establish the relationship between ρ_r and r and X_C and C. First, we state that the interpretation of ρ_r coincides with r for all ABox individuals that are not affected by the update $\mathcal{U}$:

$$\mathcal{A}_{\mathsf{r}} = \{(\exists\rho_r.\{b\} \leftrightarrow \exists r.\{b\})(a) \mid a,b \in \mathsf{Ind}(\mathcal{U}), r \in \mathsf{role}(\mathcal{A}), r(a,b) \notin \mathcal{U}, \neg r(a,b) \notin \mathcal{U}\}.$$

Second, we ensure that each concept name X_C, $C \in \mathsf{sub}(\mathcal{A})$ represents the extension of C before the update (and thus behaves like the concept $C^{\mathcal{U}}$) by taking the conjunction C_{bi} of all concepts in Figure 6, i.e., one biimplication for each fresh concept name X_C, $C \in \mathsf{sub}(\mathcal{A})$. Note that the biimplication for X_A, $A \in \mathsf{N_C}$, which is given in the first line, states that X_A is interpreted like A except on the individuals where an update of A occurred. In particular, if $A \notin \mathsf{sub}(\mathcal{U})$, then $(\bigsqcap_{\substack{A(a) \in \mathcal{U} \text{ or} \\ \neg A(a) \in \mathcal{U}}} \neg\{a\})$ is equivalent to $\top$ and, therefore, the concept in the first line is equivalent to $X_A \leftrightarrow A$. On those individuals where an update occurred, the only constraints for X_A are those given in $\mathcal{A}_{\mathsf{init}}$. It can thus be verified that the possible extensions of X_A are precisely the possible extensions of A before the update (in general, there is more than one possibility, e.g. if $A(a) \in \mathcal{U}$ and $\{A(a), \neg A(a)\} \cap \mathcal{A} = \emptyset$). The fresh roles ρ_r can be understood similarly, with the ABox $\mathcal{A}_{\mathsf{r}}$ playing the role of the biimplications for X_A. However, there is also one major difference: since we cannot express that ρ_r has the same extension as r on non-ABox domain elements, we use r instead of ρ_r when dealing with such elements. This is reflected by the use of both r and ρ_r in the biimplications for number restrictions.

Unfortunately, $\mathcal{ALCQIO}^@$ also lacks the expressive power to enforce that C_{bi} is satisfied by all domain elements. We thus resort to enforcing that C_{bi} is satisfied by all 'relevant' domain elements, i.e., by all domain elements that can be reached from an ABox individual in $\mathcal{A}$ by a sequence of roles that occurs in some concept $C \in \mathsf{sub}(\mathcal{A})$. Formally, we inductively associate with each concept $C \in \mathsf{sub}(\mathcal{A})$ a set $\mathsf{path}(C)$ of words $r_1 \ldots r_n \in \mathsf{N_R}^*$ as follows, where ε denotes

$$
\begin{array}{rcl}
\multicolumn{3}{l}{(\displaystyle\bigsqcap_{\substack{A(a)\in\mathcal{U}\text{ or}\\ \neg A(a)\in\mathcal{U}}} \neg\{a\}) \rightarrow (X_A \leftrightarrow A)} \\
X_{\{a\}} & \leftrightarrow & \{a\} \\
X_{@_a C} & \leftrightarrow & @_a X_C \\
X_{\neg C} & \leftrightarrow & \neg X_C \\
X_{C\sqcap D} & \leftrightarrow & X_C \sqcap X_D \\
X_{C\sqcup D} & \leftrightarrow & X_C \sqcup X_D \\
X_{(\geqslant m\ r\ C)} & \leftrightarrow & ((\displaystyle\bigsqcap_{a\in\mathsf{Ind}(\mathcal{U})} \neg\{a\}) \sqcap (\geq m\ r\ X_C)) \sqcup \\
& & \Big((\displaystyle\bigsqcup_{a\in\mathsf{Ind}(\mathcal{U})}\{a\}) \sqcap \bigsqcup_{\substack{m_1+m_2=m\\ m_2\leq|\mathsf{Ind}(\mathcal{U})|}} ((\geq m_1\ r\ (\bigsqcap_{b\in\mathsf{Ind}(\mathcal{U})} \neg\{b\}) \sqcap X_C) \\
& & \qquad \sqcap (\geq m_2\ \rho_r\ (\displaystyle\bigsqcup_{b\in\mathsf{Ind}(\mathcal{U})}\{b\}) \sqcap X_C))\Big) \\
X_{(\leqslant m\ r\ C)} & \leftrightarrow & ((\displaystyle\bigsqcap_{a\in\mathsf{Ind}(\mathcal{U})} \neg\{a\}) \sqcap (\leq m\ r\ X_C)) \sqcup \\
& & \Big((\displaystyle\bigsqcup_{a\in\mathsf{Ind}(\mathcal{U})}\{a\}) \sqcap \bigsqcup_{\substack{m_1+m_2=m\\ m_2\leq|\mathsf{Ind}(\mathcal{U})|}} ((\leq m_1\ r\ (\bigsqcap_{b\in\mathsf{Ind}(\mathcal{U})} \neg\{b\}) \sqcap X_C) \\
& & \qquad \sqcap (\leq m_2\ \rho_r\ (\displaystyle\bigsqcup_{b\in\mathsf{Ind}(\mathcal{U})}\{b\}) \sqcap X_C))\Big)
\end{array}
$$

Figure 6: The conjuncts of C_{bi}.

the empty sequence:

$$
\begin{array}{rcl}
\mathsf{path}(\{a\}) & = & \{\varepsilon\} \\
\mathsf{path}(A) & = & \{\varepsilon\} \\
\mathsf{path}(C_1 \sqcap C_2) & = & \mathsf{path}(C_1) \cup \mathsf{path}(C_2) \\
\mathsf{path}(C_1 \sqcup C_2) & = & \mathsf{path}(C_1) \cup \mathsf{path}(C_2) \\
\mathsf{path}(\neg C) & = & \mathsf{path}(C) \\
\mathsf{path}(@_a C) & = & \mathsf{path}(C) \\
\mathsf{path}(\geqslant m\ r\ C) & = & \{rw \mid w \in \mathsf{path}(C)\} \cup \{\varepsilon\} \\
\mathsf{path}(\leqslant m\ r\ C) & = & \{rw \mid w \in \mathsf{path}(C)\} \cup \{\varepsilon\}
\end{array}
$$

Let $\mathsf{path}(\mathcal{A}) = \bigcup\{\mathsf{path}(C) \mid C \in \mathsf{sub}(\mathcal{A})\}$. Now choose a fresh individual name a^* and a fresh role name u, and set

$$\mathcal{A}_{\mathsf{rel}} = \{\forall uw.C_{\mathsf{bi}}(a^*) \mid w \in \mathsf{path}(\mathcal{A})\} \cup \{u(a^*, b) \mid b \in \mathsf{Ind}(\mathcal{A})\},$$

where $\forall w.C$ abbreviates $\forall r_1.\cdots.\forall r_n.C$ when $w = r_1 \cdots r_n$. Finally, let

$$\mathcal{A}' = \mathcal{A}_{\mathsf{init}} \cup \mathcal{A}_{\mathsf{r}} \cup \mathcal{A}_{\mathsf{rel}} \cup \mathcal{U}.$$

Before proving that $\mathcal{A} \Longrightarrow^{\mathsf{p}}_{\mathcal{U}} \mathcal{A}'$, we give an example that also illustrates our approach to avoiding the exponential blowup of type (ii) described above. Let

$\mathcal{A} = \{a : \exists r.A, a : \exists s.\neg A\}$ and $\mathcal{U} = \{A(a_1), \neg A(a_2)\}$ and note that, due to the UNA and since r does not occur in $\mathcal{U}$, we can simplify $a : (\exists r.A)^{\mathcal{U}}$ to the equivalent assertion $a : \exists r.A^{\mathcal{U}}$. Thus, the conjunction of $\mathcal{U}$ with the following assertion is a simplified version of the semantic update computed by our algorithm:

$$
\begin{aligned}
& a : \exists r.A \wedge a : \exists s.\neg A \\
\vee\ & a : \exists r.(A \sqcap \neg\{a_1\}) \wedge a : \exists s.\neg(A \sqcap \neg\{a_1\}) \\
\vee\ & a : \exists r.(A \sqcup \{a_2\}) \wedge a : \exists s.\neg(A \sqcup \{a_2\}) \\
\vee\ & a : \exists r.((A \sqcup \{a_2\}) \sqcap \neg\{a_1\}) \wedge a : \exists s.\neg((A \sqcup \{a_2\}) \sqcap \neg\{a_1\})
\end{aligned}
$$

Here, the four disjuncts are due to the fact that we have to take the disjunction over all $\bigwedge \mathcal{A}^{\mathcal{U}'}$ for $\mathcal{U}' \subseteq \mathcal{U}$. Intuitively, this disjunction reflects the fact that each of $A(a_1)$ and $\neg A(a_2)$ might or might not be satisfied already before the update. The projective update is the union of $\mathcal{U}$ and the following ABoxes (where we have simplified C_{bi} by taking into account that for any role $r \notin \mathsf{sig}(\mathcal{U})$, the right hand side of the biimplication for $\exists r.C$ is equivalent to $\exists r.X_C$):

$$
\begin{aligned}
\mathcal{A}_{\mathsf{init}} &= \{a : X_{\exists r.A}, a : X_{\exists s.\neg A}\} \\
\mathcal{A}_{\mathsf{r}} &= \emptyset \\
\mathcal{A}_{\mathsf{rel}} &= \{u(a^*, a)\} \cup \{\forall u.C_{\mathsf{bi}}(a^*), \forall u.\forall r.C_{\mathsf{bi}}(a^*), \forall u.\forall s.C_{\mathsf{bi}}(a^*)\} \text{ where} \\
C_{\mathsf{bi}} &= (X_{\exists r.A} \leftrightarrow \exists r.X_A) \sqcap (X_{\exists s.\neg A} \leftrightarrow \exists s.X_{\neg A}) \sqcap (X_{\neg A} \leftrightarrow \neg X_A) \\
&\quad \sqcap ((\neg\{a_1\} \sqcap \neg\{a_2\}) \rightarrow (X_A \leftrightarrow A))
\end{aligned}
$$

Observe that the interpretation of $X_{\exists r.A}$ and $X_{\exists s.\neg A}$ on a_1 and a_2 is not constrained using a case distinction but by demanding that $\mathcal{A}_{\mathsf{init}}$ is satisfied.

Lemma 14. $\mathcal{A} \Longrightarrow^{\mathsf{p}}_{\mathcal{U}} \mathcal{A}'$.

Proof. Assume first that $\mathcal{I} \in M(\mathcal{A})^{\mathcal{U}}$. We have to show that there exists a model $\mathcal{I}'$ of $\mathcal{A}'$ that coincides with $\mathcal{I}$ for all symbols distinct from u, a^*, the role names ρ_r, $r \in \mathsf{role}(\mathcal{A})$, and the concept names X_C, $C \in \mathsf{sub}(\mathcal{A})$. By definition, there exists a model $\mathcal{I}_0$ of $\mathcal{A}$ such that $\mathcal{I} = \mathcal{I}_0^{\mathcal{U}}$. Now define $\mathcal{I}'$ in the same way as $\mathcal{I}$ but extended by setting

$$
\begin{aligned}
(a^*)^{\mathcal{I}'} &:= d_0, \text{ for some } d_0 \in \Delta^{\mathcal{I}}, \\
u^{\mathcal{I}'} &:= \{(d_0, b^{\mathcal{I}}) \mid b \in \mathsf{Ind}(\mathcal{A})\}, \\
\rho_r^{\mathcal{I}'} &:= r^{\mathcal{I}_0}, \text{ for } r \in \mathsf{role}(\mathcal{A}), \\
X_C^{\mathcal{I}'} &:= C^{\mathcal{I}_0}, \text{ for } C \in \mathsf{sub}(\mathcal{A}).
\end{aligned}
$$

It is easily verified that $\mathcal{I}'$ is a model of $\mathcal{A}'$ and, therefore, as required.

Conversely, assume that $\mathcal{I}$ is a model of $\mathcal{A}'$. We construct a model $\mathcal{I}_0$ of $\mathcal{A}$ such that $\mathcal{I} = \mathcal{I}_0^{\mathcal{U}}$. Define $\mathcal{I}_0$ as follows: for all concept names $A \notin \mathsf{sub}(\mathcal{A}) \cap \mathsf{sub}(\mathcal{U})$, set $A^{\mathcal{I}_0} = A^{\mathcal{I}}$. For all concept names $A \in \mathsf{sub}(\mathcal{A}) \cap \mathsf{sub}(\mathcal{U})$, set $d \in A^{\mathcal{I}_0}$ iff:

1. $d \in A^{\mathcal{I}}$ and $d \notin \{a^{\mathcal{I}} \mid a \in \mathsf{Ind}(\mathcal{U})\}$; or

2. $d \in X_A^{\mathcal{I}}$ and $d \in \{a^{\mathcal{I}} \mid a \in \mathsf{Ind}(\mathcal{U})\}$.

Similarly, for all role names $r \notin \mathsf{role}(\mathcal{A}) \cap \mathsf{role}(\mathcal{U})$, set $r^{\mathcal{I}_0} = r^{\mathcal{I}}$. For all role names $r \in \mathsf{roles}(\mathcal{A}) \cap \mathsf{role}(\mathcal{U})$, set $(d_1, d_2) \in r^{\mathcal{I}_0}$ iff:

1. $(d_1, d_2) \in r^{\mathcal{I}}$ and $\{d_1, d_2\} \not\subseteq \{a^{\mathcal{I}} \mid a \in \mathsf{Ind}(\mathcal{U})\}$; or
2. $(d_1, d_2) \in \rho_r^{\mathcal{I}}$ and $\{d_1, d_2\} \subseteq \{a^{\mathcal{I}} \mid a \in \mathsf{Ind}(\mathcal{U})\}$.

We show that $\mathcal{I}_0$ is a model of $\mathcal{A}$ and $\mathcal{I} = \mathcal{I}_0^{\mathcal{U}}$.

Claim 1. For all $C(a) \in \mathcal{A}$, $\mathcal{I}_0 \models C(a)$ iff $\mathcal{I} \models X_C(a)$.

We show that for all $C \in \mathsf{sub}(\mathcal{A})$, subconcepts E of C, and words w and $d \in \Delta^{\mathcal{I}}$ such that $(b^{\mathcal{I}}, d) \in w^{\mathcal{I}}$ for some $b \in \mathsf{Ind}(\mathcal{A})$ and $\{wv \mid v \in \mathsf{path}(E)\} \subseteq \mathsf{path}(C)$, we have

$$d \in E^{\mathcal{I}_0} \text{ iff } d \in X_E^{\mathcal{I}}.$$

Claim 1 then follows immediately by taking $w = \varepsilon$. The proof is by structural induction on E and uses $\mathcal{A}_{\mathsf{rel}}$. We consider the cases where E is a concept name or of the form $(\geqslant m\ r\ F)$. First let $E = A$ for a concept name A. Since $\mathsf{path}(A) = \{\varepsilon\}$, we have to consider all d with $(b^{\mathcal{I}}, d) \in w^{\mathcal{I}}$ for some $b \in \mathsf{Ind}(\mathcal{A})$ and $w \in \mathsf{path}(C)$. Assume d is given. If $d \notin \{a^{\mathcal{I}} \mid a \in \mathsf{Ind}(\mathcal{U})\}$, then $d \in X_A^{\mathcal{I}}$ iff $d \in A^{\mathcal{I}}$ (first biimplication in Figure 6) iff $d \in A^{\mathcal{I}_0}$, by definition of $A^{\mathcal{I}_0}$. Otherwise, if $d \in \{a^{\mathcal{I}} \mid a \in \mathsf{Ind}(\mathcal{U})\}$, then $d \in A^{\mathcal{I}_0}$ iff $d \in X_A^{\mathcal{I}}$, again by definition of $A^{\mathcal{I}_0}$.

Now consider the case $E = (\geqslant m\ r\ F)$. Assume $(b^{\mathcal{I}}, d) \in w^{\mathcal{I}}$ for some $b \in \mathsf{Ind}(\mathcal{A})$ and $wv \in \mathsf{path}(C)$ for all $v \in \mathsf{path}(E)$. Then $wrv \in \mathsf{path}(C)$ for all $v \in \mathsf{path}(F)$.

We distinguish the following cases:

- $d \notin \{a^{\mathcal{I}} \mid a \in \mathsf{Ind}(\mathcal{U})\}$. By the biimplication for $X_{(\geqslant m\ r\ F)}$ in Figure 6, we have that $d \in X_E^{\mathcal{I}}$ iff $d \in (\geqslant m\ r\ X_F)^{\mathcal{I}}$. For all x we have that $(d, x) \in r^{\mathcal{I}}$ iff $(d, x) \in r^{\mathcal{I}_0}$. Moreover, for all x such that $(d, x) \in r^{\mathcal{I}}$, we have that $(b^{\mathcal{I}}, x) \in (wr)^{\mathcal{I}}$. By IH, it holds that $x \in F^{\mathcal{I}_0}$ iff $x \in X_F^{\mathcal{I}}$. Thus we obtain $d \in X_E^{\mathcal{I}}$ iff $d \in (\geqslant m\ r\ F)^{\mathcal{I}_0} = E^{\mathcal{I}_0}$.

- $d \in \{a^{\mathcal{I}} \mid a \in \mathsf{Ind}(\mathcal{U})\}$. Again by the biimplication for $X_{(\geqslant m\ r\ F)}$, we have that $d \in X_E^{\mathcal{I}}$ iff for some m_1, m_2 such that $m_1 + m_2 = m$ and $m_2 \leq |\mathsf{Ind}(\mathcal{U})|$ we have both $d \in (\geq m_1\ r \bigsqcap_{b \in \mathsf{Ind}(\mathcal{U})} \neg\{b\} \sqcap X_F)^{\mathcal{I}}$ and $d \in (\geq m_2\ \rho_r \bigsqcup_{b \in \mathsf{Ind}(\mathcal{U})} \{b\} \sqcap X_F)^{\mathcal{I}}$.

  Then $d \in (\geq m_1\ r \bigsqcap_{b \in \mathsf{Ind}(\mathcal{U})} \neg\{b\} \sqcap X_F)^{\mathcal{I}}$ iff $d \in (\geq m_1\ r \bigsqcap_{b \in \mathsf{Ind}(\mathcal{U})} \neg\{b\} \sqcap F)^{\mathcal{I}_0}$ can be proved analogously to the previous case. Note that if $x = c^{\mathcal{I}}$ for some $c \in \mathsf{Ind}(\mathcal{A})$, then $(d, x) \in r^{\mathcal{I}_0}$ iff $(d, x) \in \rho_r^{\mathcal{I}}$. Moreover, by IH, $c^{\mathcal{I}} \in F^{\mathcal{I}_0}$ iff $c^{\mathcal{I}} \in X_F^{\mathcal{I}}$. Thus $d \in (\geq m_2\ \rho_r \bigsqcup_{b \in \mathsf{Ind}(\mathcal{U})} \{b\} \sqcap X_F)^{\mathcal{I}}$ iff $d \in (\geq m_2\ r \bigsqcup_{b \in \mathsf{Ind}(\mathcal{U})} \{b\} \sqcap F)^{\mathcal{I}_0}$. Summing up the previus equivalences, we obtain that $d \in X_E^{\mathcal{I}}$ iff $d \in (\geqslant m\ r\ F)^{\mathcal{I}_0} = E^{\mathcal{I}_0}$, as required.

From Claim 1, the condition that $\mathcal{I}$ is a model of $\mathcal{A}_{\mathsf{init}}$, and the definition of $r^{\mathcal{I}_0}$, we obtain that $\mathcal{I}_0$ is a model of $\mathcal{A}$.

It remains to show that $\mathcal{I} = \mathcal{I}_0^{\mathcal{U}}$. First, interpretations of concept and role names which do not appear in $\mathcal{U}$ are identical in $\mathcal{I}_0$ and $\mathcal{I}$. Second, $\mathcal{I}$ and $\mathcal{I}_0$ interpret all role and concept names which appear in $\mathcal{U}$ in the same way on the part of the domain $\Delta^{\mathcal{I}}$ unaffected by the update $\mathcal{U}$: for concept names A, the definition of $A^{\mathcal{I}_0}$, and the first biimplication in Figure 6 imply that for all $x \in \Delta^{\mathcal{I}} \setminus \{a^{\mathcal{I}} \mid A(a) \in \mathcal{U} \text{ or } \neg A(a) \in \mathcal{U}\}$ it holds that $x \in A^{\mathcal{I}_0}$ iff $x \in A^{\mathcal{I}}$. Smilarly, for role names r, the definition of $r^{\mathcal{I}_0}$ and $\mathcal{I} \models \mathcal{A}_{\mathsf{r}}$ imply that for all $(x, y) \in (\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}) \setminus \{(a^{\mathcal{I}}, b^{\mathcal{I}}) \mid r(a, b) \in \mathcal{U} \text{ or } \neg r(a, b) \in \mathcal{U}\}$ it holds that $(x, y) \in r^{\mathcal{I}_0}$ iff $(x, y) \in r^{\mathcal{I}}$. Thus, since $\mathcal{I} \models \mathcal{U}$, we obtain that $\mathcal{I} = \mathcal{I}_0^{\mathcal{U}}$. $\square$

We now analyze the size of $\mathcal{A}'$ in terms of the size of $\mathcal{A}$ and $\mathcal{U}$. Obviously, $|\mathcal{A}_{\mathsf{init}}| \leq |\mathcal{A}|$ and $|\mathcal{A}_{\mathsf{r}}| = \mathcal{O}(|\mathcal{U}|^3)$. Since $|C_{\mathsf{bi}}| = \mathcal{O}(|\mathcal{U}|^3 \cdot |\mathcal{A}|)$ (independently from the coding of numbers inside number restrictions) and $|\mathsf{path}(\mathcal{A})| \leq |\mathcal{A}|^2$, we obtain that $|\mathcal{A}_{\mathsf{rel}}| = \mathcal{O}(|\mathcal{A}|^3 \cdot |\mathcal{U}|^3)$. Summing up, we obtain $|\mathcal{A}'| = \mathcal{O}(|\mathcal{U}|^3 \cdot |\mathcal{A}|^3)$.

Together with Lemma 7, we thus obtain the following result, which in particular implies that all DLs between $\mathcal{ALCO}$ and $\mathcal{ALCQIO}^{@}$ have projective updates. It is independent of the coding of numbers inside number restrictions.

Theorem 8. *Let $\mathcal{L}$ be a DL between $\mathcal{ALCO}$ and $\mathcal{ALCQIO}^{@}$. Then there is a polynomial p such that, for every $\mathcal{L}$-ABox $\mathcal{A}$ and every update $\mathcal{U}$, there is an $\mathcal{L}$-ABox $\mathcal{A}'$ such that*

1. $\mathcal{A} \Longrightarrow_{\mathcal{U}}^{\mathsf{p}} \mathcal{A}'$;
2. $|\mathcal{A}'| \leq p(|\mathcal{A}| \cdot |\mathcal{U}|)$;
3. $\mathcal{A}'$ *can be computed in time* $p(|\mathcal{A}'|)$.

In a context where also TBoxes are available, it might be more appropriate to store the conjuncts of C_{bi} in a TBox rather than in $\mathcal{A}'$. In this way, we do not need to introduce the new individual a^* and can avoid the paths of role names altogether. It can be seen that an acyclic TBox suffices, please see [15] to get a more concrete idea.

We close this section with a brief discussion of iterated updates in the projective case. To start with, we note that it is possible to repeatedly compute projective updates using the presented construction by simply treating fresh symbols introduced by earlier updates as 'normal' symbols during all subsequent updates. These updates are then even stronger than necessary since later updates preserve the meaning of fresh symbols introduced by earlier updates. Unfortunately, it is easy to see that the projective update $\mathcal{A}'$ that is obtained by starting with an ABox $\mathcal{A}$ and then consecutively applying updates $\mathcal{U}_1, \ldots, \mathcal{U}_n$ using our construction is exponential in n. In particular, if $\mathcal{A} = \mathcal{A}_0, \ldots, \mathcal{A}_n = \mathcal{A}'$ are the generated projective updates, then it is easy to see that each $\mathcal{A}_i$ contains $k^i \cdot |\mathsf{sub}(\mathcal{A})|$ concept names X_C, for some constant k. Despite this problem, it is still straightforward to carry out repeated updates without ever obtaining an ABox of exponential size. The simple workaround is to keep the original

ABox $\mathcal{A}$ in memory and then to repeatedly update the updates $\mathcal{U}_i$ instead of the projective updates $\mathcal{A}_i$. More precisely, define a cumulative update

$$\widehat{\mathcal{U}}_i = \mathcal{U}_i \cup (\widehat{\mathcal{U}}_{i-1} \setminus \neg\mathcal{U}_{i+1}).$$

for $1 \leq i \leq n$. When processing the stream of updates $\mathcal{U}_1, \ldots, \mathcal{U}_n$, only keep $\mathcal{A}$ and the latest cumulative update $\widehat{\mathcal{U}}_i$ in memory. At any time, the 'current' projective update is obtained by applying our construction to $\mathcal{A}$ and $\widehat{\mathcal{U}}_i$. Clearly, all projective updates obtained in this way are of size polynomial in $|\mathcal{A}|$ and $|\mathcal{U}_1| + \cdots + |\mathcal{U}_n|$.

6. Conditional Updates and Reasoning about Action

Up to now, we have considered updates $\mathcal{U}$ that are unconditional in the sense that all assertions in $\mathcal{U}$ are necessarily true after the update, not subject to any conditions. In some applications, though, it is more useful to allow *conditional updates* that are able to express statements such as '$A(a)$ is true after the update if $C(b)$ was true before'. In particular, such a generalization is important for reasoning about actions, which has recently been studied in a DL context [12] and where ABox updates play a crucial role for implementing the reasoning pattern of 'progression', as opposed to 'regression' approaches [7]. In this section, we introduce conditional updates and show that all results that we have proved for unconditional updates, both semantic and projective, are also true for the corresponding version of conditional updates. As an application, we put conditional updates to work for reasoning about action, using the progression approach to reprove the optimal upper complexity bounds for the projection problem of DL actions that were first established using regression in [12]. The latter results will be based on projective updates.

Let $\mathcal{L}$ be a DL between $\mathcal{ALC}$ and $\mathcal{ALCQIO}^@$. A *conditional $\mathcal{L}$-update $\mathcal{U}$* is a finite set of expressions φ/ψ, where the *precondition* φ is an $\mathcal{L}$-ABox assertion and the *postcondition* ψ is, as in the unconditional case, an assertion of one of the forms

$$A(a), \neg A(a), r(a,b), \neg r(a,b).$$

Intuitively, $\varphi/\psi \in \mathcal{U}$ means that if φ holds before the update, then ψ holds after it. Analogously to the case of unconditional updates, we require a consistency condition: if φ/ψ and $\varphi'/\neg\psi$ are both in $\mathcal{U}$, then the ABox $\{\varphi, \varphi'\}$ has to be inconsistent. We now adapt the notion of an interpretation update to the case of conditional updates.

Definition 5 (Conditional Interpretation Update). *Let $\mathcal{U}$ be a conditional update and $\mathcal{I}$ an interpretation. Define an interpretation $\mathcal{I}^{\mathcal{U}}$ by setting for all* *individual names a, concept names A, and role names r:*

$$
\begin{aligned}
\Delta^{\mathcal{I}^{\mathcal{U}}} &= \Delta^{\mathcal{I}} \\
a^{\mathcal{I}^{\mathcal{U}}} &= a^{\mathcal{I}} \\
A^{\mathcal{I}^{\mathcal{U}}} &= (A^{\mathcal{I}} \cup \{a^{\mathcal{I}} \mid \varphi/A(a) \in \mathcal{U} \text{ and } \mathcal{I} \models \varphi\}) \setminus \{a^{\mathcal{I}} \mid \varphi/\neg A(a) \in \mathcal{U} \text{ and } \mathcal{I} \models \varphi\} \\
r^{\mathcal{I}^{\mathcal{U}}} &= (r^{\mathcal{I}} \cup \{(a^{\mathcal{I}}, b^{\mathcal{I}}) \mid \varphi/r(a,b) \in \mathcal{U} \text{ and } \mathcal{I} \models \varphi\}) \\
&\qquad\qquad\qquad\qquad \setminus \{(a^{\mathcal{I}}, b^{\mathcal{I}}) \mid \varphi/\neg r(a,b) \in \mathcal{U} \text{ and } \mathcal{I} \models \varphi\}.
\end{aligned}
$$

$\mathcal{I}^{\mathcal{U}}$ *is called the* result of updating $\mathcal{I}$ with $\mathcal{U}$*.*

The conditional versions of semantic and projective updates are defined in the same way as for unconditional updates. We repeat the definition for the reader's convenience.

Definition 6 (Semantic (Projective) Conditional Updates). *Let $\mathcal{A}$ be an $\mathcal{ALCQIO}^@$-ABox and $\mathcal{U}$ a conditional update.*

An $\mathcal{ALCQIO}^@$-ABox $\mathcal{A}'$ is a semantic update *of $\mathcal{A}$ with $\mathcal{U}$, in symbols $\mathcal{A} \Longrightarrow_{\mathcal{U}} \mathcal{A}'$, if*

$$M(\mathcal{A}') = \{\mathcal{I}^{\mathcal{U}} \mid \mathcal{I} \in M(\mathcal{A})\}$$

A description logic $\mathcal{L}$ has semantic conditional updates *if for every $\mathcal{L}$-ABox $\mathcal{A}$ and conditional $\mathcal{L}$-update $\mathcal{U}$, there is an $\mathcal{L}$-ABox $\mathcal{A}'$ with $\mathcal{A} \Longrightarrow_{\mathcal{U}} \mathcal{A}'$.*

An $\mathcal{ALCQIO}^@$-ABox $\mathcal{A}'$ is a projective update *of $\mathcal{A}$ with $\mathcal{U}$, in symbols $\mathcal{A} \Longrightarrow^{\mathsf{p}}_{\mathcal{U}} \mathcal{A}$, if*

$$M(\mathcal{A}')_{\restriction \overline{\mathsf{Fr}(\mathcal{A}')}} = \{\mathcal{I}^{\mathcal{U}} \mid \mathcal{I} \in M(\mathcal{A})\}_{\restriction \overline{\mathsf{Fr}(\mathcal{A}')}},$$

where $\mathsf{Fr}(\mathcal{A}') = \mathsf{sig}(\mathcal{A}') \setminus (\mathsf{sig}(\mathcal{A}) \cup \mathsf{sig}(\mathcal{U}))$ is the set of fresh symbols in $\mathcal{A}'$. A description logic $\mathcal{L}$ has semantic projective conditional updates *if for every $\mathcal{L}$-ABox $\mathcal{A}$ and conditional $\mathcal{L}$-update $\mathcal{U}$, there is an $\mathcal{L}$-ABox $\mathcal{A}'$ with $\mathcal{A} \Longrightarrow^{\mathsf{p}}_{\mathcal{U}} \mathcal{A}'$.*

Note that conditional updates generalize unconditional ones since assertions ψ of unconditional updates can be expressed as $\top(a)/\psi$, with a an arbitrary individual name. It follows that all non-existence results that we have proved in Section 3 for unconditional updates hold for conditional updates as well. Regarding results about the existence of updates, our first aim is to generalize Theorems 3, 4, and 8 from the unconditional to the conditional case. We leave the straightforward generalization of other results such as Theorem 7 to the interested reader. We start with extending Theorem 3.

Theorem 9. *$\mathcal{ALCO}^@$, $\mathcal{ALCIO}^@$, $\mathcal{ALCQO}^@$, and $\mathcal{ALCQIO}^@$ have semantic conditional updates.*

Proof. Let $\mathcal{A}$ be an ABox formulated in one of the DLs in Theorem 9 and $\mathcal{U}$ a conditional update. The assertions that occur on the left-hand-side of update statements in $\mathcal{U}$ is $\mathsf{lhs}(\mathcal{U}) = \{\varphi \mid \varphi/\psi \in \mathcal{U}\}$. Each $\mathcal{D} \subseteq \mathsf{lhs}(\mathcal{U})$ corresponds to one possible choice of preconditions that are true before the update. For each such $\mathcal{D}$, the ABox

$$\mathsf{Pre}_{\mathcal{D}} = \mathcal{D} \cup \{\neg\varphi \mid \varphi \in \mathsf{lhs}(\mathcal{U}) \setminus \mathcal{D}\}$$

identifies the models that realize the choice $\mathcal{D}$ and

$$\mathsf{Post}_{\mathcal{D}} = \{\psi \mid \varphi/\psi \in \mathcal{U}, \varphi \in \mathcal{D}\}$$

is the unconditional update that has to be executed in those models. By Lemma 11, we have

$$(\mathcal{A} \cup \mathsf{Pre}_{\mathcal{D}}) \Longrightarrow_{\mathsf{Post}_{\mathcal{D}}} \mathcal{A}_{\mathcal{D}}$$

where

$$\mathcal{A}_{\mathcal{D}} = \bigwedge \mathsf{Post}_{\mathcal{D}} \wedge \bigvee_{\mathcal{U}' \subseteq \mathsf{Post}_{\mathcal{D}}} \bigwedge (\mathcal{A} \cup \mathsf{Pre}_{\mathcal{D}})^{\mathcal{U}'}.$$

By the semantics of ABox updates, it follows that $\mathcal{A} \Longrightarrow_{\mathcal{U}} \mathcal{A}'$ where

$$\mathcal{A}' = \bigvee_{\mathcal{D} \subseteq \mathsf{lhs}(\mathcal{U})} \Big(\bigwedge \mathsf{Post}_{\mathcal{D}} \wedge \bigvee_{\mathcal{U}' \subseteq \mathsf{Post}_{\mathcal{D}}} \bigwedge (\mathcal{A} \cup \mathsf{Pre}_{\mathcal{D}})^{\mathcal{U}'} \Big).$$

□

It follows from the construction of $\mathcal{A}'$ in the above proof that the upper bound on the size of semantic updates given in Theorem 4 still applies in the conditional case. We now consider projective conditional updates, generalizing Theorem 8.

Theorem 10. *Let $\mathcal{L}$ be a DL between $\mathcal{ALCO}$ and $\mathcal{ALCQIO}^{@}$. Then there is a polynomial p such that, for every $\mathcal{L}$-ABox $\mathcal{A}$ and every conditional $\mathcal{L}$-update $\mathcal{U}$, there is an $\mathcal{L}$-ABox $\mathcal{A}'$ such that*

1. $\mathcal{A} \Longrightarrow^{\mathsf{p}}_{\mathcal{U}} \mathcal{A}'$;
2. $|\mathcal{A}'| \leq p(|\mathcal{A}| \cdot |\mathcal{U}|)$;
3. *$\mathcal{A}'$ can be computed in time $p(|\mathcal{A}'|)$.*

Proof. (sketch) Assume that $\mathcal{A}$ is an $\mathcal{ALCQIO}^{@}$-ABox and $\mathcal{U}$ a conditional $\mathcal{ALCQIO}^{@}$-update. To define a projective update, introduce fresh concept names X_A for all $A \in \mathsf{sub}(\mathcal{A} \cup \mathsf{lhs}(\mathcal{U}))$ and fresh role names ρ_A for all $r \in \mathsf{role}(\mathcal{A} \cup \mathsf{lhs}(\mathcal{U}))$. The additional names for concepts and roles in $\mathsf{lhs}(\mathcal{U})$ are used to represent the preconditions of $\mathcal{U}$ that hold in the original interpretation. The component $\mathcal{A}_{\mathsf{init}}$ is defined in the same way as in the proof of Theorem 8 by setting

$$\begin{array}{rcl} \mathcal{A}_{\mathsf{init}} & = & \{X_C(a) \mid C(a) \in \mathcal{A}\} \cup \\ & & \{\rho_r(a,b) \mid r(a,b) \in \mathcal{A}\} \cup \\ & & \{\neg\rho_r(a,b) \mid \neg r(a,b) \in \mathcal{A}\} \end{array}$$

Set

$$\begin{array}{rcl} \mathcal{A}_{\mathsf{r}} & = & \{(\exists \rho_r.\{b\} \leftrightarrow \exists r.\{b\})(a) \mid a,b \in \mathsf{Ind}(\mathcal{U}), r \in \mathsf{role}(\mathcal{A} \cup \mathsf{lhs}(\mathcal{U})), \\ & & \qquad r(a,b) \notin \mathsf{rhs}(\mathcal{U}), \neg r(a,b) \notin \mathsf{rhs}(\mathcal{U})\} \end{array}$$

and define C_{bi} in the same way as in Theorem 8 with the exception that C_{bi} contains one biimplication for each X_C, $C \in \mathsf{sub}(\mathcal{A} \cup \mathsf{lhs}(\mathcal{U}))$ and $\mathcal{U}$ is replaced by $\mathsf{rhs}(\mathcal{U})$ in the implication for X_A and the biimplications for qualified number restrictions. Let, as before, a^* be a fresh individual name and u a fresh role name, and set

$$\mathcal{A}_{\mathsf{rel}} = \{\forall uw.C_{\mathsf{bi}}(a^*) \mid w \in \mathsf{path}(\mathcal{A} \cup \mathsf{lhs}(\mathcal{U}))\} \cup \{u(a^*, b) \mid b \in \mathsf{Ind}(\mathcal{A} \cup \mathsf{lhs}(\mathcal{U}))\}.$$

Instead of including $\mathcal{U}$ in $\mathcal{A}'$ as in the case of unconditional updates, we have to make sure that only those updates are triggered whose preconditions are satisfied. This can be achieved by the ABox

$$\begin{aligned}\mathcal{A}_{\mathsf{cond}} \quad = \quad & \bigcup_{\varphi/\psi \in \mathcal{U}} \{a^* : \widehat{\mathsf{p}}(\varphi) \rightarrow \mathsf{p}(\psi)\} \cup \\ & \bigcup_{\psi \in \mathsf{rhs}(\mathcal{U}) \cap O} \{a^* : \big(\bigsqcap_{\varphi/\psi \in \mathcal{U}} \neg \widehat{\mathsf{p}}(\varphi)\big) \rightarrow (\widehat{\mathsf{p}}(\psi) \leftrightarrow \mathsf{p}(\psi))\},\end{aligned}$$

where O denotes the set of ABox assertions using concept and role names from $\mathcal{A} \cup \mathsf{lhs}(\mathcal{U})$ only,

$$\mathsf{p}(\varphi) = \begin{cases} \exists u.(\{a\} \sqcap C) & \text{if } \varphi = C(a) \\ \exists u.(\{a\} \sqcap \exists r.\{b\}) & \text{if } \varphi = r(a,b) \\ \exists u.(\{a\} \sqcap \forall r.\neg\{b\}) & \text{if } \varphi = \neg r(a,b) \end{cases}$$

and $\widehat{\mathsf{p}}$ is defined like p, but with C replaced by X_C and r by ρ_r. The first line of $\mathcal{A}_{\mathsf{cond}}$ states that if φ holds in the original interpretation, then ψ holds in the updated interpretation, for every $\varphi/\psi \in \mathcal{U}$. The second line says that if none of the preconditions of an assertion ψ holds in the original interpretation, then ψ holds in the updated interpretation if, and only if, it holds in the original interpretation. Now set

$$\mathcal{A}' = \mathcal{A}_{\mathsf{init}} \cup \mathcal{A}_{\mathsf{r}} \cup \mathcal{A}_{\mathsf{rel}} \cup \mathcal{A}_{\mathsf{cond}}.$$

In the same way as in the proof of Theorem 8, one can show that $\mathcal{A} \Longrightarrow^{\mathsf{p}}_{\mathcal{U}} \mathcal{A}'$. Moreover, by construction, there is a polynomial p such that $|\mathcal{A}'| \leq p(|\mathcal{A}| \cdot |\mathcal{U}|)$ and $\mathcal{A}'$ can be computed in time $p(|\mathcal{A}'|)$. $\square$

We now apply Theorem 10 to the projection problem in reasoning about action as introduced in a DL context in [12]; see also [26, 27, 14] for related work. The projection problem means to decide whether a given action achieves a given goal in a given situation, i.e., whether the goal necessarily holds true after the execution of the action. It is one of the fundamental problems in reasoning about action, and many other important reasoning problems can be reduced to it [7]. In the context of a DL $\mathcal{L}$, the *projection problem for* $\mathcal{L}$ can be formalized as deciding, given

- an $\mathcal{L}$-ABox $\mathcal{A}$ that describes the situation in which the action is executed,
- a conditional $\mathcal{L}$-update $\mathcal{U}$ that describes the action, and
- an $\mathcal{L}$-ABox assertion φ that represents the goal,

whether for every model $\mathcal{I}$ of $\mathcal{A}$ it holds that $\mathcal{I}^{\mathcal{U}} \models \varphi$. In [12], algorithms for the projection problem for various DLs have been given based on the approach of *regression*, according to which one rewrites φ to a new assertion φ' such that φ' is a consequence of the initial ABox if, and only if, φ holds after the conditional update. In effect, one thus reduces the projection problem to a standard reasoning problem for the initial ABox. The resulting algorithms yield tight upper complexity bounds for the projection problem for all DLs between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$.

Theorem 11 ([12]). *The projection problem is*

- PSpace-*complete for* $\mathcal{ALC}$, $\mathcal{ALCO}$, *and* $\mathcal{ALCQO}$;
- ExpTime-*complete for* $\mathcal{ALCI}$ *and* $\mathcal{ALCIO}$;
- co-NExpTime-*complete for* $\mathcal{ALCQI}$ *and* $\mathcal{ALCQIO}$.

Interestingly, we obtain an alternative proof of the upper bounds stated in Theorem 11 from our results on projective conditional updates. Let $\mathcal{L}$ be one of the DLs mentioned in the theorem. Given an $\mathcal{L}$-ABox $\mathcal{A}$, a conditional $\mathcal{L}$-update $\mathcal{U}$, and an $\mathcal{L}$-ABox assertion φ, we can simply compute in polynomial time a projective update $\mathcal{A}'$ of $\mathcal{A}$ with $\mathcal{U}$ that is of size polynomial in the sizes of $\mathcal{A}$ and $\mathcal{U}$ and formulated in the extension $\mathcal{LO}$ of $\mathcal{L}$ with nominals (if not already present in $\mathcal{L}$), and then decide whether $\mathcal{A}' \models \varphi$. We thus obtain a polynomial time reduction from projection in $\mathcal{L}$ to ABox consequence in $\mathcal{LO}$, a problem that is

- in PSpace if $\mathcal{LO}$ is $\mathcal{ALCO}$ or $\mathcal{ALCQO}$ [28];
- in ExpTime if $\mathcal{LO}$ is $\mathcal{ALCIO}$ [29];
- in co-NExpTime if $\mathcal{LO}$ is $\mathcal{ALCQIO}$ [30].

Thus, the upper bounds of Theorem 11 follow immediately. From the perspective of reasoning about action, this approach corresponds to *projection*, i.e., instead of 'regressing' the goal φ back to the original ABox, we 'progress' the original ABox towards the goal.

We remark that the setup in [12] is somewhat more general than the one considered here as it adds acyclic TBoxes, so-called occlusions as part of an action description that allows some concept/role memberships to change freely during the execution of the action, and establishes the algorithms and complexity bounds for sequences of actions rather than single ones. However, although it is out of the scope of the current paper to go into any details, we conjecture that our progression approach can be generalized in a straightforwards way to handle all of these extensions. In particular, sequences of actions do not increase the complexity of progression-based projection, c.f. the remark that closes Section 5.

Our proof of the upper bounds in Theorem 11 and the matching lower bounds provided in [12] also have an interesting consequence from the ABox update perspective taken in this paper. To express projective updates of ABoxes in the DLs $\mathcal{ALC}$, $\mathcal{ALCQ}$, $\mathcal{ALCI}$, and $\mathcal{ALCQI}$, thus overcoming the problems identified in Section 3, we have added nominals to the respective languages in Section 5. In the cases of $\mathcal{ALCI}$ and $\mathcal{ALCQI}$, this actually means switching to a language in which the standard reasoning problems 'ABox consistency' and 'ABox consequence' have higher computational complexity: in $\mathcal{ALCI}$, ABox consequence is PSpace-complete whereas it is ExpTime-complete in $\mathcal{ALCIO}$; in $\mathcal{ALCQI}$, ABox consequence is ExpTime-complete whereas it is co-NExpTime-complete in $\mathcal{ALCQIO}$. Now, our proof of Theorem 11 and the lower bounds stated there entail that such an increase in complexity between the language for initial ABoxes and the target language for projective updates is *unavoidable* (modulo the assumption that the involved complexity classes are distinct) if one wants projective updates to be of polynomial size and computable in polynomial time: for $\mathcal{ALCI}$, a target language with PSpace complexity would prove that the projection problem for $\mathcal{ALCI}$ is in PSpace, thus showing PSpace = ExpTime; for $\mathcal{ALCQI}$ a target language with ExpTime complexity would prove that the projection problem for $\mathcal{ALCQI}$ is in ExpTime, thus showing ExpTime = co-NExpTime.

7. Extensions and Related Work

We discuss some natural extensions of the framework considered in this paper, in particular with TBoxes and more general forms of update. We also survey the relevant literature on updates in description logic and, to some reasonable extent, updates in propositional logic.

7.1. Extensions

As laid out in the introduction, this paper has concentrated on the rather special case of ABox updates where no domain constraints are present (i.e., no TBox) and updates can only consist of ground literals. Both restrictions are severe from the point of view of many applications, and thus it is natural to try and alleviate them. In both cases, this gives rise to significant new research challenges, and we only make some basic observations in what follows.

We first consider updates that admit compound ABox assertions, i.e., updates are sets of possibly negated assertions $C(a)$ and $r(a, b)$, where C can be a compound concept. Due to the presence of disjunction and existential restrictions, updates can now be non-deterministic. Even in the propositional case, there is no one-and-only generally accepted semantics for non-deterministic updates, which has led to many different proposals [8, 4, 31, 32, 33, 34, 35, 7]. For the case of DLs, the benefits and drawbacks of the available semantics still remain to be investigated. Unfortunately, at least under the rather natural Winslett PMA semantics [4], which is based on the idea of minimizing the changes between models of the original ABox and models of the updated ABox, it is known that it is impossible to compute the result of updating an $\mathcal{ALCQI}$-ABox and represent it in a formalism for which the consequence problem is decidable—no matter whether semantic, projective, approximate, or projective approximate updates are considered. This observation is a direct consequence of the result obtained in [28] that the projection problem for $\mathcal{ALCQI}$ (as defined in Section 6) under Winslett PMA semantics is undecidable. It does not seem unlikely that other model-based update semantics induce similar computational problems. It remains an interesting open problem, however, whether undecidability results can be established already for $\mathcal{ALC}$ and $\mathcal{ALCI}$.

Next, we consider the addition of TBoxes to the framework studied in this paper. While doing so, we assume that updates have the original restricted form, i.e., are sets of ground literals. We start with the simple case of acyclic TBoxes, where only primitive concept names are allowed in the update, but no defined ones—see [1] for details on these notions. It has been shown in [15] how the construction of semantic updates presented in Section 4.1 can be adapted to this case, and how acyclic TBoxes can help to achieve a more succinct presentation of updated ABoxes through structure sharing. All results presented in this paper can be easily extended to acyclic TBoxes under the described restrictions of updates. When this restriction is dropped, TBoxes (no matter whether acyclic, cyclic, or general) induce the same semantic and computational problems as compound concepts in the update. In fact, instead of putting $C(a)$ into the update with C a compound concept, one can equivalently define the abbreviation $A \equiv C$ in the TBox and then use the ground literal $A(a)$ in the update. Thus, we need an advanced semantics such as Winslett PMA and encounter the same computational problems that were described above for updates with compound ABox assertions. We refer the interested reader to [36] for a pragmatic approach to this problem in the context of projection.

7.2. Related work in propositional logic

We discuss the relationship between updates of DL ABoxes as investigated in this paper and the existing literature on updates and revisions of propositional logic theories. Since propositional logic is expressively complete, which means that every class of models can be described by a formula, the problem of non-expressibility of updates that we address in the context of DLs does not exist there. For the same reason, in propositional logic there is no difference between approximate and semantic updates. In contrast, the problem of determining the size of updated or revised propositional theories is of great interest and has been extensively investigated. First examples of exponential blowups in the representation of revised propositional theories were given by Nebel [37] and Winslett [4]. A systematic discussion of succinctness issues for a large range of different update and revision operators is provided by Cadoli et al. in [9]. In fact, [9] seems to be the first paper to make the distinction between semantic and projective updates and to study the impact that this distinction has on the size of updates.[2] We note, however, that the special case of updates by literals

[2] Note that Cadoli et al. use a slightly different terminology; e.g., they call an update operator *query compactable* if it has projective updates of polynomial size and *logically compactable* if it has semantic updates of polynomial size. Thus, in the terminology of [9], we have shown

(as studied in this paper) is not considered in [9], where all considered forms of update may involve any propositional formula.

7.3. Related work in description logic

The update, revision and evolution of description logic knowledge bases has recently received considerable attention. In our discussion, we focus on the update literature; summarizing the work on revision and evolution [42, 43] is outside the scope of this paper, but see [38, 39, 40, 41] and [42, 43], respectively.

Besides of the work presented here that is based on and extends [15], instance level updates have also been investigated for knowledge bases formulated in variants of the DL-Lite family of description logics [11, 44]. This family consists of inexpressive DLs tailored towards capturing conceptual modeling constructs while keeping reasoning, in particular conjunctive query answering, of very low complexity [2]. Similarly to what we do in the current paper, Calvanese et al. [11, 44] investigate the problem of updating a DL-Lite ABox by ground literals. In addition, DL-Lite TBoxes serve as domain constraints. The work presented in [11] assumes a model-based Winslett style semantics for updates and gives a variety of results on the existence and size of semantic and approximate updates for the description logic DL-Lite$_\mathcal{F}$. Unfortunately, as observed in [44], the fundamental algorithm *computeUpdate* of [11] is unsound and some expressivity results for Winslett style updates in DL-Lite$_\mathcal{F}$ claimed in [11] do not hold. To solve the resulting expressivity problems (as well as examples of non-intuitive updates resulting from the interaction between the TBox and the updates), two formula-based approaches to instance level updates under TBoxes in DL-Lite are proposed in [44]. For these approaches, updates of polynomial size exist. It would be of interest to investigate in how far such formula-based approach can be extended to the expressive DLs considered in this paper.

TBox level updates have received much less attention than instance level updates. A main reason may be that modifications of the TBox are typically not the result of changes in the application domain, but rather invoked due to the TBox engineer changing her *understanding* of the application domain. Thus, on the TBox level the belief revision problem seems much more relevant than the update problem, and the former is governed by different principles. Some pros and cons of model-based and formula-based semantics in this context are presented in [45].

8. Conclusion

We have studied updates of description logic ABoxes in the restricted, yet fundamental case where no compound concepts are admitted in the ABox and no TBoxes are present. Our results show that, while many description logics do

that for DLs between $\mathcal{ALCO}$ and $\mathcal{ALCQIO}^{@}$, the update operator considered in this paper is query compactable.

not have updates, by choosing the right DL (one that includes nominals), it is possible to guarantee the existence of updates. Moreover, by choosing the right notion of update (projective semantic), it is even possible to compute updated ABoxes in polynomial time. We have also described an application of our results in reasoning about action.

Regarding future work, it would be interesting to study less restricted cases where the update $\mathcal{U}$ is allowed to contain compound concepts and TBoxes are admitted. Note, though, that this involves some rather serious challenges that we have identified and discussed in Section 6. It would also be interesting to consider ABox revision instead of ABox update, for which a number of competing semantics are available; see [8, 9] and references therein. We believe that the results and techniques established in this paper would also be useful to deal with those semantics.

Acknowledgements

We would like to thank Franz Baader, Yursi Bong, Conrad Drescher, and Michael Thielscher for ideas and discussions. The first author was supported by the DFG project BA1122/10-2 and the third author by the DFG Graduiertenkolleg 334. Finally, we would like to thank the anonymous referees for helpful comments.

References

[1] F. Baader, D. L. McGuinness, D. Nardi, P. Patel-Schneider, The Description Logic Handbook: Theory, implementation and applications, Cambridge University Press, 2003.

[2] D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, R. Rosati, Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family, Journal of Automated Reasoning 39 (3) (2007) 385–429.

[3] R. Kontchakov, C. Lutz, D. Toman, F. Wolter, M. Zakharyaschev, The combined approach to query answering in DL-Lite, in: Proceedings of the 12th International Conference on the Principles of Knowledge Representation and Reasoning (KR10), AAAI Press, 2010, pp. 247–257.

[4] M. Winslett, Updating Logical Databases, Cambridge University Press, 1990.

[5] R. Fagin, G. M. Kuper, J. D. Ullman, M. Y. Vardi, Updating logical databases, Advances in Computing Research 3 (1986) 1–18.

[6] H. Katsuno, A. O. Mendelzon, On the difference between updating a knowledge base and revising it, in: Proceedings of the 2nd International Conference on the Principles of Knowledge Representation and Reasoning (KR91), Morgan Kaufmann, 1991, pp. 387–394.

[7] R. Reiter, Knowledge in Action, MIT Press, 2001.

[8] T. Eiter, G. Gottlob, On the complexity of propositional knowledge base revision, updates, and counterfactuals, Artificial Intelligence 57 (2-3) (1992) 227–270.

[9] M. Cadoli, F. M. Donini, P. Liberatore, M. Schaerf, The size of a revised knowledge base, Artificial Intelligence 115 (1) (1999) 25–64.

[10] J. Lang, F. Lin and P. Marquis, Causal Theories of Action: A Computational Core, in: Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI03), AAAI Press, 2003, pp. 1073-1078.

[11] G. D. Giacomo, M. Lenzerini, A. Poggi, R. Rosati, On instance-level update and erasure in description logic ontologies, Journal of Logic and Computation 19 (5) (2009) 745–770.

[12] F. Baader, C. Lutz, M. Milicic, U. Sattler, F. Wolter, Integrating description logics and action formalisms: First results, in: Proceedings of the 20th National Conference on Artificial Intelligence (AAAI05), AAAI Press / The MIT Press, 2005, pp. 572–577.

[13] C. Drescher, H. Liu, F. Baader, S. Guhlemann, U. Petersohn, P. Steinke, M. Thielscher, Putting ABox updates into action, in: Proceedings of the 7th International Symposium on Frontiers of Combining Systems (FroCos09), Vol. 5749 of LNCS, Springer, pp. 214–229.

[14] C. Drescher, M. Thielscher, Integrating action calculi and description logics, in: Proceedings of the 30th Annual German Conference on AI (KI07), Vol. 4667 of LNCS, Springer, 2007, pp. 68–83.

[15] H. Liu, C. Lutz, M. Milicic, F. Wolter, Updating description logic ABoxes, in: Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR06), AAAI Press, 2006, pp. 46–56.

[16] C. Areces and B. ten Cate, Hybrid Logic, in: P. Blackburn, J. van Benthem, and F. Wolter (Eds.), Handbook of Modal Logic, Elsevier 2007, pp. 821–868.

[17] C. Lutz, D. Toman, F. Wolter, Conjunctive query answering in the description logic $\mathcal{EL}$ using a relational database system, in: Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI09), AAAI Press, 2009, pp. 2070–2075.

[18] P. Blackburn, J. van Benthem, Modal logic: a semantic perspective, in: P. Blackburn, J. van Benthem, F. Wolter (Eds.), Handbook of Modal Logic, Elsevier, 2007, pp. 2–79.

[19] R. M. Karp, R. J. Lipton, Some connections between nonuniform and uniform complexity classes, in: Proceedings of the 12th Annual ACM Symposium on Theory of Computing (STOC80), ACM, 1980, pp. 302–309.

[20] D. Mundici, Tautologies with a unique Craig interpolant, uniform vs. nonuniform complexity, Annals of Pure and Applied Logic 27 (1984) 265–273.

[21] J. Krajícek, Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic, Journal of Symbolic Logic 62 (2) (1997) 457–486.

[22] U. Schöning, J. Torán, A note on the size of craig interpolants, in: Vol. 06451 of Dagstuhl Seminar Proceedings, Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2006.

[23] A. Borgida, On the relative expressiveness of description logics and predicate logics, Artificial Intelligence 82 (1 - 2) (1996) 353–367.

[24] C. Lutz, U. Sattler, F. Wolter, Modal logics and the two-variable fragment, in: Annual Conference of the European Association for Computer Science Logic CSL'01, Vol. 2142 of LNCS, Springer, 2001, pp. 247–261.

[25] H. Liu, Computing updates in description logics, Ph.D. thesis, TU Dresden (2010).

[26] Y. Gu, M. Soutchanski, A description logic based situation calculus, Annals of Mathematics and Artificial Intelligence DOI 10.1007/s10472-010-9176-z (2010) 1–81.

[27] L. Chang, F. Lin, Z. Shi, A dynamic description logic for representation and reasoning about actions, in the 2nd International Conference on Knowledge Science, Engineering and Management (KSEM07), Vol. 4798 of LNCS, Springer, 2007, pp. 115–127.

[28] F. Baader, M. Milicic, C. Lutz, U. Sattler, F. Wolter, Integrating description logics and action formalisms for reasoning about web services, LTCS-Report LTCS-05-02, Chair for Automata Theory, Institute for Theoretical Computer Science, Dresden University of Technology, Germany. http://lat.inf.tu-dresden.de/research/reports.html. (2005).

[29] C. Areces, P. Blackburn, M. Marx, A road-map on complexity for hybrid logics, in: Proceedings of the 13th Conference for Computer Science Logic (CSL99), Vol. 1683 of LNCS, Springer, 1999, pp. 307–321.

[30] S. Tobies, The complexity of reasoning with cardinality restrictions and nominals in expressive description logics, Journal of Artificial Intelligence Research 12 (2000) 199–217.

[31] K. D. Forbus, Introducing actions into qualitative simulations, in: Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI89), Morgan Kaufmann, 1989, pp. 1273–1279.

[32] F. Lin, Embracing causality in specifying the indeterminate effects of actions, in: Proceedings of the 14th National Conference on Artificial Intelligence (AAAI-96), MIT Press, 1996, pp. 670–676.

[33] M. Thielscher, Nondeterministic actions in the fluent calculus: Disjunctive state update axioms, in: Intellectics and Computational Logic, Kluwer Academic, 2000, pp. 327–345.

[34] Y. Zhang, N. Foo, Updating knowledge bases with disjunctive information, Computational Intelligence 16 (2000) 1–22.

[35] A. Herzig, The PMA revisited, in: Proceedings of the 5th International Conference on the Principles of Knowledge Representation and Reasoning (KR96), Morgan Kaufmann, 1996, pp. 40–50.

[36] H. Liu, C. Lutz, M. Milicic, F. Wolter, Reasoning about actions using description logics with general TBoxes, in: 10th European Conference on Logics in Artificial Intelligence (JELIA06), Vol. 4160 of LNCS, Springer, 2006, pp. 266–279.

[37] B. Nebel, Base revision operations and schemes: Semantics, representation and complexity, in: Proceedings of the 11th European Conference on Artificial Intelligence (ECAI94), John Wiley and Sons, 1994, pp. 341–345.

[38] G. Qi, W. Liu, D. A. Bell, Knowledge base revision in description logics, in: 10th European Conference on Logics in Artificial Intelligence (JELIA06), Vol. 4160 of LNCS, Springer, 2006, pp. 386–398.

[39] G. Qi, F. Yang, A survey of revision approaches in description logics, in: Proceedings of the 2008 International Workshop on Description Logics (DL08), CEUR Workshop Proceedings Volume 353, 2008.

[40] G. Qi, J. Du, Model-based revision operators for terminologies in description logics, in: Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI09), AAAI Press, 2009, pp. 891–897.

[41] Z. Wang, K. Wang, R. W. Topor, Revising general knowledge bases in description logics, in: in: Proceedings of the 12th International Conference on the Principles of Knowledge Representation and Reasoning (KR10), AAAI Press, 2010, pp. 599–601.

[42] G. Flouris, M. d'Aquin, G. Antoniou, J. Z. Pan, D. Plexousakis, Special issue on ontology dynamics, Journal of Logic and Computation 19 (5) (2009) 717–719.

[43] G. Flouris, D. Manakanatas, H. Kondylakis, D. Plexousakis, G. Antoniou, Ontology change: classification and survey, Knowledge Engineering Review 23 (2) (2008) 117–152.

[44] D. Zheleznyakov, D. Calvanese, E. Kharlamov, W. Nutt, Updating TBoxes in DL-Lite, in: Proceedings of the 2010 International Workshop on Description Logics (DL10), CEUR Workshop Proceedings Volume 573, 2010.

[45] D. Calvanese, E. Kharlamov, W. Nutt, D. Zheleznyakov, Updating ABoxes in DL-Lite, in: Proceeding of the 4th Alberto Mendelzon International Workshop on Foundations of Data Management (AMW10), 2010.

Appendix A. Proofs for Section 2

Observation 1. $\mathcal{E} \Longrightarrow_{\mathcal{U}} \mathcal{E}'$.

Proof. Recall that

$$\begin{array}{rcl} \mathcal{E} & = & \{\mathsf{john} : \exists\mathsf{has\text{-}child}.\mathsf{Happy},\ \mathsf{mary} : \mathsf{Happy} \sqcap \mathsf{Clever}\} \\ \mathcal{U} & = & \{\neg\mathsf{Happy}(\mathsf{mary})\} \\ \mathcal{E}' & = & \{\mathsf{john} : \exists\mathsf{has\text{-}child}.(\mathsf{Happy} \sqcup \{\mathsf{mary}\}),\ \mathsf{mary} : \neg\mathsf{Happy} \sqcap \mathsf{Clever}\} \end{array}$$

First, let $\mathcal{I}$ be a model of $\mathcal{E}$. By definition of $\mathcal{I}^{\mathcal{U}}$, we have $\mathsf{mary}^{\mathcal{I}^{\mathcal{U}}} \notin \mathsf{Happy}^{\mathcal{I}_{\mathcal{U}}}$ and $\mathsf{mary}^{\mathcal{I}^{\mathcal{U}}} \in \mathsf{Clever}^{\mathcal{I}_{\mathcal{U}}}$ and thus the second assertion of $\mathcal{E}'$ is satisfied. For the first assertion, first assume that there is a $d \in \mathsf{Happy}^{\mathcal{I}}$ with $(\mathsf{john}^{\mathcal{I}}, d) \in \mathsf{has\text{-}child}^{\mathcal{I}}$ and $d \neq \mathsf{mary}^{\mathcal{I}}$. By definition of $\mathcal{I}^{\mathcal{U}}$, we have $\mathsf{john}^{\mathcal{I}_{\mathcal{U}}} \in (\exists\mathsf{has\text{-}child}.\mathsf{Happy})^{\mathcal{I}_{\mathcal{U}}}$ and thus the first assertion is satisfied. If there is no such d, then we must have $(\mathsf{john}^{\mathcal{I}}, \mathsf{mary}^{\mathcal{I}}) \in \mathsf{has\text{-}child}^{\mathcal{I}} = \mathsf{has\text{-}child}^{\mathcal{I}_{\mathcal{U}}}$. Thus $\mathsf{john}^{\mathcal{I}_{\mathcal{U}}} \in (\exists\mathsf{has\text{-}child}.\{\mathsf{mary}\})^{\mathcal{I}_{\mathcal{U}}}$ and the first assertion is satisfied.

Now let $\mathcal{J}$ be a model of $\mathcal{E}'$. Let $\mathcal{I}$ be the interpretation obtained from $\mathcal{J}$ by setting $\mathsf{Happy}^{\mathcal{I}} = \mathsf{Happy}^{\mathcal{J}} \cup \{\mathsf{mary}^{\mathcal{J}}\}$. By definition, $\mathcal{I}^{\mathcal{U}} = \mathcal{J}$ and $\mathcal{I}$ satisfies the second assertion in $\mathcal{E}$. Moreover, it is obvious that both $\mathsf{john}^{\mathcal{J}} \in (\exists\mathsf{has\text{-}child}.\mathsf{Happy})^{\mathcal{J}}$ and $\mathsf{john}^{\mathcal{J}} \in (\exists\mathsf{has\text{-}child}.\{\mathsf{mary}\})^{\mathcal{J}}$, one of which must be the case, imply that $\mathcal{I}$ satisfies the first assertion in $\mathcal{E}$. ☐

Observation 2. $\mathcal{E} \longrightarrow^{\mathcal{U}}_{\mathcal{ALC}} \mathcal{E}''$.

Proof. By Point 2 of Lemma 2, it is sufficient to show that the ABoxes

$$\begin{array}{rcl} \mathcal{E}' & = & \{\mathsf{john} : \exists\mathsf{has\text{-}child}.(\mathsf{Happy} \sqcup \{\mathsf{mary}\}),\ \mathsf{mary} : \neg\mathsf{Happy} \sqcap \mathsf{Clever}\} \\ \mathcal{E}'' & = & \{\mathsf{john} : \exists\mathsf{has\text{-}child}.(\mathsf{Happy} \sqcup \mathsf{Clever}),\ \mathsf{mary} : \neg\mathsf{Happy} \sqcap \mathsf{Clever}\} \end{array}$$

have the same $\mathcal{ALC}$-consequences, i.e., $\mathcal{E}' \models \varphi$ iff $\mathcal{E}'' \models \varphi$ for all $\mathcal{ALC}$-ABox assertions $\mathcal{E}$. We have that $\mathcal{E}' \not\models \varphi$ implies $\mathcal{E}'' \not\models \varphi$ since every model of $\mathcal{E}'$ is also a model of $\mathcal{E}''$. For the converse, let $\mathcal{E}'' \not\models \varphi$. If φ is a (possibly negated) role assertion, we are done since $\mathcal{E}'$ does not entail any such assertion. Hence, let $\varphi = C_0(a_0)$ be a concept assertion and take a model $\mathcal{I}$ of $\mathcal{E}''$ with $a_0^{\mathcal{I}} \notin C_0^{\mathcal{I}}$. Define a new model $\mathcal{J}$ with $\Delta^{\mathcal{J}} = \Delta^{\mathcal{I}} \times \mathsf{Ind}(\mathcal{A})$ by interpreting concept names A, role names r, and individual names a as follows:

$$\begin{array}{rcl} A^{\mathcal{J}} & = & \{(d,a) \mid d \in A^{\mathcal{I}} \text{ and } a \in \mathsf{Ind}(\mathcal{E})\} \\ r^{\mathcal{J}} & = & \{((d,a),(e,a)) \mid (d,e) \in r^{\mathcal{I}} \text{ and } a \in \mathsf{Ind}(\mathcal{E})\} \\ a^{\mathcal{J}} & = & (a^{\mathcal{I}}, a) \end{array}$$

It can be proved by induction on the structure of C that $d \in C^{\mathcal{I}}$ iff $(d,a) \in C^{\mathcal{I}}$ for all $d \in \Delta^{\mathcal{I}}$, $a \in \mathsf{Ind}(\mathcal{E})$, and $\mathcal{ALC}$-concepts C. It follows that $\mathcal{J} \models \mathcal{E}''$ and $a_0^{\mathcal{J}} \notin C_0^{\mathcal{J}}$. If $\mathcal{J}$ is a model of $\mathcal{E}'$, we are done. Otherwise, $\mathcal{J} \models \mathcal{E}''$ and $\mathcal{J} \not\models \mathcal{E}'$ jointly imply that there is a $d \in (\neg\mathsf{Happy} \sqcap \mathsf{Clever})^{\mathcal{J}}$ such that $(\mathsf{john}^{\mathcal{J}}, d) \in \mathsf{has\text{-}child}^{\mathcal{J}}$. Distinguish the following two cases:

1. $a_0 \neq \mathsf{mary}$.
   Then let $\mathcal{J}'$ be obtained from $\mathcal{J}$ by setting $\mathsf{mary}^{\mathcal{J}'} = d$. By construction and since $\mathcal{J} \models \mathcal{A}''$, we have $\mathcal{J}' \models \mathcal{E}'$. Moreover, $a_0^{\mathcal{J}'} \notin C_0^{\mathcal{J}'}$ since $a_0^{\mathcal{J}} \notin C_0^{\mathcal{J}}$, $a_0^{\mathcal{J}'} = a_0^{\mathcal{J}}$, and C_0 does not contain nominals.
2. $a_0 = \mathsf{mary}$.
   Then let $\mathcal{J}'$ be obtained from $\mathcal{J}$ by setting $\mathsf{has\text{-}child}^{\mathcal{J}'} = \mathsf{has\text{-}child}^{\mathcal{J}} \cup \{(\mathsf{john}^{\mathcal{J}}, \mathsf{mary}^{\mathcal{J}})\}$. We have $\mathcal{J}' \models \mathcal{E}'$ and it remains to show that $\mathsf{mary}^{\mathcal{J}'} \notin C_0^{\mathcal{J}'}$. Let Ω denote the elements reachable in $\mathcal{J}'$ from mary, i.e., Ω is the smallest set such that $\mathsf{mary}^{\mathcal{J}'} \in \Omega$ and if $d \in \Omega$ and $(d, e) \in r^{\mathcal{I}}$ for some $r \in \mathsf{N_R}$, then $e \in \Omega$. It can be shown by induction on the structure of C that $d \in C^{\mathcal{J}}$ iff $d \in C^{\mathcal{J}'}$ for all $d \in \Omega$ and $\mathcal{ALC}$-concepts C; a crucial element of the proof is the observation that, due to the definition of $\mathcal{J}$ and $\mathcal{J}'$, $\mathsf{john}^{\mathcal{J}'} \notin \Omega$. Obviously, $a_0^{\mathcal{J}} \notin C^{\mathcal{J}}$ then yields $\mathsf{mary}^{\mathcal{J}'} \notin C_0^{\mathcal{J}'}$ as required.

□

Lemma 6.

1. Let $\mathcal{L}$ be a DL between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$. Then for every Boolean $\mathcal{L}^{@}$-ABox, there exists an equivalent Boolean $\mathcal{L}$-ABox;
2. Let $\mathcal{L}$ be a DL between $\mathcal{ALCO}$ and $\mathcal{ALCQIO}$. Then for every Boolean $\mathcal{L}$-ABox, there exists an equivalent non-Boolean $\mathcal{L}^{@}$-ABox.

Proof. Concerning (i), let $\mathcal{A}$ be a Boolean $\mathcal{L}^{@}$-ABox, and let φ be an assertion from $\mathcal{A}$ such that $@_b D$ is a subconcept of some concept occurring in φ. Then the ABox $\mathcal{A}'$ is obtained from $\mathcal{A}$ by replacing φ with $(D(b) \wedge \varphi[\top/@_b D]) \vee (\neg D(b) \wedge \varphi[\bot/@_b D])$, where $C[X/@_b D]$ denotes the concept obtained from φ by replacing all occurrences of $@_b D$ with X. Using the semantics, it is easy to see that $\mathcal{A}'$ is equivalent to $\mathcal{A}$. By iterating this replacement, we will eventually obtain a Boolean $\mathcal{L}$-ABox.

Concerning (ii), define a mapping $\cdot^*$ from ABox assertions in $\mathcal{L}$ to $\mathcal{L}^{@}$-concepts as follows:

$$
\begin{array}{rcl}
C(a)^* & := & @_a C \\
r(a, b)^* & := & @_a \exists r.\{b\} \\
\neg r(a, b)^* & := & @_a \forall r.\neg\{b\}
\end{array}
$$

The mapping is extended to Boolean ABox assertion φ as follows: φ^* is the $\mathcal{L}^{@}$-concept obtained by replacing $\wedge$ with $\sqcap$, $\vee$ with $\sqcup$, and every assertion ψ with ψ^*. Now, let $\mathcal{A} = \{\varphi_1, \ldots, \varphi_n\}$ be a Boolean $\mathcal{L}$-ABox. Define a non-Boolean $\mathcal{L}^{@}$-ABox $\mathcal{A}' := \{(\varphi_1^* \sqcap \cdots \sqcap \varphi_n^*)(a)\}$, where a is an arbitrary individual name. Using the semantics, it is easy to see that $\mathcal{A}'$ is equivalent to $\mathcal{A}$. □