TECHNISCHE UNIVERSITÄT DRESDEN

Dresden University of Technology
Institute for Theoretical Computer Science
Chair for Automata Theory

LTCS–Report

Updating Description Logic ABoxes

Hongkai Liu, Carsten Lutz, Maja Miličić, Frank Wolter

LTCS-Report 05-10

Lehrstuhl für Automatentheorie
Institut für Theoretische Informatik
TU Dresden
http://lat.inf.tu-dresden.de

Hans-Grundig-Str. 25
01062 Dresden
Germany

Updating Description Logic ABoxes

H. Liu[1], C. Lutz[1], M. Miličić[1], F. Wolter[2]

[1]Inst. für Theoretische Informatik
TU Dresden
Germany

[2]Dept. of Computer Science
University of Liverpool
UK

Abstract

Description logic (DL) ABoxes are a tool for describing the state of affairs in an application domain. In this paper, we consider the problem of updating ABoxes when the state changes. We assume that changes are 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 by investigating whether the updated ABox can be expressed in these DLs and, if so, whether it is computable and what is its size. It turns out that DLs have to include nominals and the "@" constructor of hybrid logic (or, equivalently, admit Boolean ABoxes) for updated ABoxes to be expressible. We devise algorithms to compute updated ABoxes in several expressive DLs and show that an exponential blowup in the size of the whole input (original ABox + update information) cannot be avoided unless every PTime problem is LogTime-parallelizable. We also exhibit ways to avoid an exponential blowup in the size of the original ABox, which is usually large compared to the update information.

1 Introduction

Description logics (DLs) are a prominent family of logic-based formalisms for the representation of and reasoning about conceptual knowledge [4]. In DLs, concepts are used to describe classes of individuals sharing common properties. For example, the following concept describes the class of all parents with only happy children:

$$\mathsf{Person} \sqcap \exists \mathsf{has\text{-}child}.\mathsf{Person} \sqcap \forall \mathsf{has\text{-}child}.(\mathsf{Person} \sqcap \mathsf{Happy})$$

This concept is formulated in $\mathcal{ALC}$, the basic DL that contains all Boolean operators [21]. Concepts are the most important ingredient of description logic

ABoxes, whose purpose is to describe a snapshot of the world. For example, the following ABox, which is also formulated in $\mathcal{ALC}$, says that John is a parent with only happy children, that Peter is his child, and that Mary is a person:

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

In many applications of DLs, an ABox is used to represent the current state of affairs in the application domain [4]. In such applications, it is necessary to update the ABox in the case that the world has changed. Such an update should incorporate the information about the new state while retaining all knowledge that is not affected by the change (as demanded by the principle of inertia, see e.g. [14]). For example, if Mary is not happy any longer, we should update the above ABox to the following one. This updated ABox is formulated in $\mathcal{ALCO}$, the extension of $\mathcal{ALC}$ with nominals (i.e., individual names inside concept descriptions):

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

Observe that new information concerning Mary also resulted in an update of the information concerning John because the semantics for ABoxes adopts the open world assumption and can therefore represent the domain in an incomplete way [4], Page 68. In the example above, we have no information about whether or not Mary is a child of John.

Surprisingly, formal theories of ABox updates have never been developed. In applications, ABoxes are usually updated in an ad-hoc way, and effects such as the information change for John above are simply ignored. The current paper aims at curing this deficiency. Its purpose is *to provide a first formal analysis of ABox updates in many common description logics, concentrating on the most basic kind of updates*. These basic updates are as follows: the new information to be incorporated into the ABox is a set of possibly negated assertions $a{:}A$ and $r(a, b)$, where A is an *atomic* concept. The motivation for considering this restricted form of updates is three-fold: first, there is a single, uncontroversial semantics for updates of this restricted form, whereas several different and equally natural semantics are available in the case of updates with complex concepts, see e.g. [26, 8, 11, 22]. Second, it follows from the results in [3] that, under Winslett-style PMA semantics [26], unrestricted ABox updates in relatively simple DLs such as $\mathcal{ALCFI}$ and its extensions are not computable. It seems very likely that the other available semantics suffer from similar computational problems. Finally, we believe that the massive non-determinism of

ABox updates with complex concepts, in particular those involving quantifiers nested in a complex way, leads to unintuitive results under all of the available semantics.

We consider restricted ABox updates in the expressive DL $\mathcal{ALCQIO}$ and its fragments. It turns out that, in many natural DLs such as $\mathcal{ALC}$, the updated ABox cannot be expressed. As an example, consider the $\mathcal{ALC}$ ABox given above. To express the ABox obtained by the (restricted) update with $\mathsf{mary}{:}\neg\mathsf{Happy}$, we had to resort to the more expressive DL $\mathcal{ALCO}$. But even the introduction of nominals does not suffice to guarantee that updated ABoxes are expressible. Only if we further add the "@" concept constructor from hybrid logic [1, 2] or Boolean ABoxes (we show that these two are equivalent in the presence of nominals), updated ABoxes can be expressed. Here, the @ constructor allows the formation of *concepts* of the form $@_aC$ expressing that the individual a satisfies C, and Boolean ABoxes are a generalization of standard ABoxes: while the latter can be thought of as a conjunction of ABox assertions of the form $a{:}C$ and $r(a, b)$, Boolean ABoxes are a Boolean combination of such ABox assertions. Our expressiveness results do not only concern $\mathcal{ALC}$: similar proofs as those given in this paper can be used to show that, in any standard DL in which nominals and the "@" constructor are not expressible, updated ABoxes cannot be expressed.

We show that updated ABoxes exist and are computable in $\mathcal{ALCQIO}^@$, the extension of $\mathcal{ALCQIO}$ (which includes nominals) with the @ constructor. The proposed algorithm can easily be adapted to the fragments $\mathcal{ALCIO}^@$ and $\mathcal{ALCQO}^@$. An important issue is the size of updated ABoxes: the updated ABoxes computed by our algorithm may be of size exponential both in the size of the original ABox and in the size of the new information (henceforth called the *update*). We show that an exponential blowup cannot be completely avoided by proving that, even in the case of propositional logic, updated theories are not polynomial in the size of the (combined) input unless every PTime-algorithm is LogTime-parallelizable (the "P vs. NP" question of parallel computation).[1] In the update literature, an exponential blowup in the size of the update is viewed as much more tolerable than an exponential blowup in the size of the original ABox since the former tend to be very small compared to the latter. We believe that, in the case of $\mathcal{ALCQIO}^@$ and its two fragments mentioned above, the exponential blowup in the size of the original ABox cannot be avoided. While we leave a proof as an open problem, we exhibit two ways around the blowup: the first is to allow the introduction of new concept definitions in an acyclic TBox

[1] In contrast to the results by Cadoli et al. [7], our result even applies to the restricted form of updates, i.e., updates in propositional logic where the update is a conjunction of literals. Thus, our argument provides further evidence for the claims in [7], where it is shown that, with unrestricted updates, an exponential blowup in the size of the update cannot be avoided unless the first levels of the polynomial hierarchy collapse.

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 $\mathcal{ALCQIO}$.

when computing the update. The second is to move to extensions of $\mathcal{ALCQIO}^@$ that allow Boolean operators on roles, thus eliminating the asymmetry between concepts and roles found in standard DLs. In both cases, we show how to compute updated ABoxes that are polynomial in the size of the original ABox (and exponential in the size of the update). Thus, the blowup induced by updates in these expressive DLs is not worse than in propositional logic. We also show that the blowup produced by iterated updates is not worse than the blowup produced by a single update.

This paper is organized as follows. In Section 2, we introduce the relevant DLs, formally define ABox updates, and establish the result on the exponential blowup induced by restricted updates in propositional logic. We then prove in Section 3 that updated ABoxes cannot be expressed in $\mathcal{ALC}$, in $\mathcal{ALCO}$, $\mathcal{ALC}^@$, and $\mathcal{ALC}$ with Boolean ABoxes. In Section 4, we show how to compute ABox updates in $\mathcal{ALCQIO}^@$ and analyze the size of the computed ABoxes. Finally, Section 5 is concerned with the computation of updated ABoxes whose size is only polynomial in the size of the original ABox. We conclude in Section 6, which is also used to discuss potential further work.

2 Preliminaries

2.1 Description Logics

In DLs, *concepts* are inductively defined with the help of a set of *constructors*, starting with a set $\mathsf{N_C}$ of *concept names* and a set $\mathsf{N_R}$ of *role names*, and (possibly) a set $\mathsf{N_I}$ of *individual names*. In this section, we introduce the DL $\mathcal{ALCQIO}^@$, whose concepts are formed using the constructors shown in Figure 1. There, the inverse constructor is the only role constructor, whereas the remain-

Symbol	Constructor	$\mathcal{ALC}$	$\mathcal{ALCO}$	$\mathcal{ALCQ}$	$\mathcal{ALCI}$	$\mathcal{ALCQO}$	$\mathcal{ALCIO}$	$\mathcal{ALCQI}$
$\mathcal{Q}$	$(\leqslant n\ r\ C)$ $(\geqslant n\ r\ C)$			x		x		x
$\mathcal{I}$	r^-				x		x	x
$\mathcal{O}$	$\{a\}$		x			x	x	

Figure 2: Fragments of $\mathcal{ALCQIO}$.

ing seven constructors are concept constructors. In Figure 1 and throughout this 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 (i.e., role names and inverses thereof), A, B to denote concept names, and C, D to denote (possibly complex) 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 DL 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 sublanguages as indicated in Figure 2. For each language $\mathcal{L}$ listed in Figure 2, we have an analogue $\mathcal{L}^{@}$ obtained by adding the @ constructor.

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 individual $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. The extension of $\cdot^{\mathcal{I}}$ to inverse roles and arbitrary concepts is inductively defined as shown in the third column of Figure 1.

An $\mathcal{ALCQIO}^{@}$ *assertional box (ABox)* is a finite set of *concept assertions* $C(a)$ and *role assertions* $r(a, b)$ and $\neg r(a, b)$ (where r may be an inverse role). For readability, we sometimes write concept assertions as $a{:}C$. Observe that there is no need for explicitly introducing negated concept assertions as negation is available as a concept constructor in $\mathcal{ALCQIO}^{@}$. 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.

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 role assertion $\neg r(a, b)$ iff $(a^{\mathcal{I}}, b^{\mathcal{I}}) \notin r^{\mathcal{I}}$. We denote satisfaction of an ABox assertion ψ by an intepretation $\mathcal{I}$ with $\mathcal{I} \models \psi$. An interpretation $\mathcal{I}$ is a *model* of an ABox $\mathcal{A}$ (written $\mathcal{I} \models \mathcal{A}$) if it satisfies all assertions in $\mathcal{A}$. An ABox is *consistent* iff it has a model. Two ABoxes $\mathcal{A}$ and

$\mathcal{A}'$ are *equivalent* (written $\mathcal{A} \equiv \mathcal{A}'$) iff they have the same models.

The *length* of a concept C, denoted by $|C|$, is the number of symbols needed to write C. The *size* of an ABox assertion $C(a)$ is $|C|$, the size of $r(a,b)$ and $\neg r(a,b)$ is 1. Finally, the *size* of an ABox $\mathcal{A}$, denoted by $|\mathcal{A}|$, is the sum of the sizes of all assertions in $\mathcal{A}$.

2.2 ABox Updates

We introduce a simple form of ABox update where complex concepts are not allowed in the update information.

Definition 1 (Interpretation Update). An *update* $\mathcal{U}$ is a simple ABox that is consistent. Let $\mathcal{U}$ be an update and $\mathcal{I}$, $\mathcal{I}'$ interpretations such that $\Delta^{\mathcal{I}} = \Delta^{\mathcal{I}'}$ and $\mathcal{I}$ and $\mathcal{I}'$ agree on the interpretation of individual names. Then $\mathcal{I}'$ is the *result of updating* $\mathcal{I}$ *with* $\mathcal{U}$, *written* $\mathcal{I} \Longrightarrow_{\mathcal{U}} \mathcal{I}'$, if the following hold:

- for all concept names A, $A^{\mathcal{I}'} = (A^{\mathcal{I}} \cup \{a^{\mathcal{I}} \mid A(a) \in \mathcal{U}\}) \setminus \{a^{\mathcal{I}} \mid \neg A(a) \in \mathcal{U}\}$;
- for all role names r,

$$r^{\mathcal{I}'} = (r^{\mathcal{I}} \cup \{(a^{\mathcal{I}}, b^{\mathcal{I}}) \mid r(a,b) \in \mathcal{U}\}) \setminus \{(a^{\mathcal{I}}, b^{\mathcal{I}}) \mid \neg r(a,b) \in \mathcal{U}\}.$$

$\triangle$

Now let $\mathcal{A}$ be an ABox and $\mathcal{U}$ an update. Then, up to equivalence, there exists at most one ABox $\mathcal{A}'$ satisfying the conditions

(U1) $\forall \mathcal{I}, \mathcal{I}' : \big(\, (\, \mathcal{I} \models \mathcal{A} \; \wedge \; \mathcal{I} \Longrightarrow_{\mathcal{U}} \mathcal{I}' \,) \; \rightarrow \; \mathcal{I}' \models \mathcal{A}' \, \big)$ and

(U2) $\forall \mathcal{I}' : \big(\, \mathcal{I}' \models \mathcal{A}' \; \rightarrow \; \exists \mathcal{I} : (\, \mathcal{I} \models \mathcal{A} \; \wedge \; \mathcal{I} \Longrightarrow_{\mathcal{U}} \mathcal{I}' \,) \, \big)$.

In other words, whenever ABoxes $\mathcal{A}'$ and $\mathcal{A}''$ satisfy (U1) and (U2), then $\mathcal{A}' \equiv \mathcal{A}''$. This observation justifies the following definition.

Definition 2 (ABox Update). Let $\mathcal{A}$ be an ABox and $\mathcal{U}$ an update. An ABox $\mathcal{A}'$ is the *result of updating* $\mathcal{A}$ *with* $\mathcal{U}$, in symbols $\mathcal{A} * \mathcal{U} \equiv \mathcal{A}'$, if $\mathcal{A}'$ satisfies the conditions (U1) and (U2). We then call $\mathcal{A}$ the *original ABox* and $\mathcal{A}'$ the *updated ABox*. $\triangle$

As mentioned in the introduction, there are two technical reasons for restricting ourselves to updates of this simple form. First, it allows us to use the uncontroversial semantics given above, which coincides with all standard semantics for updates considered in the literature, see e.g., [23, 20, 26, 18]. In contrast, for unrestricted updates involving complex concepts there exist several competing semantics such as the ones proposed by Winslett [25] and Forbus [8]. Semantics for theory revision are closely related as well, but yield different results even in our restricted setting [5, 19]. Second, we consider it quite likely that, under many of these semantics and for many DLs, unrestricted ABox updates are not computable even if the updated ABoxes exist. Some evidence is given by the results in [3], which imply that this is the case under Winslett-style semantics for the DL $\mathcal{ALCFI}$ and all of its extensions.[2]

Practically, our restriction means that the user has to describe updates at an atomic level. It is clear that more complex updates such as Boolean combinations of concept names can be useful in applications. On the other hand, we believe that the utility of *arbitrarily* complex concepts in updates is limited since using such concepts together with a standard update semantics introduces massive non-determinism into updates. For example, the very simply update $\{\forall r.A(a)\}$ applied to an interpretation $\mathcal{I}$ under most standard semantics for updates means that for each individual $x \in A^{\mathcal{I}}$ with $(a^{\mathcal{I}}, x) \in r^{\mathcal{I}}$, we have to decide whether to change $(a^{\mathcal{I}}, x) \in r^{\mathcal{I}}$ to $(a^{\mathcal{I}}, x) \notin r^{\mathcal{I}}$ or $x \in A^{\mathcal{I}}$ to $x \notin A^{\mathcal{I}}$ (but we are not allowed apply both changes). With complex nested concepts, this non-determinism quickly grows out of bounds.

We now give another example of updating ABoxes. The following $\mathcal{ALCO}$ ABox expresses that John and Mary are married. We also know that one of them is happy, and the other is not. However, we do not know which of the two is unhappy. Moreover, Peter and Sarah both have happy parents:

$$\begin{array}{l}
\mathsf{spouse}(\mathsf{john}, \mathsf{mary}) \\
\mathsf{peter}{:}\exists \mathsf{parent}.\mathsf{Happy} \\
\mathsf{sarah}{:}\exists \mathsf{parent}.\mathsf{Happy} \\
\mathsf{john}{:}\big(\mathsf{Happy} \sqcap \exists \mathsf{spouse}.(\{\mathsf{mary}\} \sqcap \neg\mathsf{Happy})\big) \sqcup \\
\qquad \big(\neg\mathsf{Happy} \sqcap \exists \mathsf{spouse}.(\{\mathsf{mary}\} \sqcap \mathsf{Happy})\big)
\end{array}$$

Suppose that, because one of them is unhappy, John and Mary have an argument. This results in both John and Mary being unhappy. Hence, we should apply the following update to the above ABox:

$$\neg\mathsf{Happy}(\mathsf{john}), \quad \neg\mathsf{Happy}(\mathsf{mary}).$$

Then, the updated ABox can be expressed in $\mathcal{ALCO}^{@}$ as follows:[3]

$$\begin{array}{l}
\neg\mathsf{Happy}(\mathsf{john}) \\
\neg\mathsf{Happy}(\mathsf{mary}) \\
\mathsf{spouse}(\mathsf{john}, \mathsf{mary}) \\
\mathsf{john}{:}\big(@_{\mathsf{peter}}\exists \mathsf{parent}.(\mathsf{Happy} \sqcup \{\mathsf{john}\}) \sqcap @_{\mathsf{sarah}}\exists \mathsf{parent}.(\mathsf{Happy} \sqcup \{\mathsf{john}\})\big) \sqcup \\
\qquad \big(@_{\mathsf{peter}}\exists \mathsf{parent}.(\mathsf{Happy} \sqcup \{\mathsf{mary}\}) \sqcap @_{\mathsf{sarah}}\exists \mathsf{parent}.(\mathsf{Happy} \sqcup \{\mathsf{mary}\})\big)
\end{array}$$

[2] $\mathcal{ALCFI}$ is the extension of $\mathcal{ALCI}$ with at-least and at-most restrictions that admit only the number 1.

[3] To save brackets, we assume that the @ constructor has higher precedence than conjunction.

The only surprising assertion in the updated ABox is the last one. Intuitively, it represents the update of the last two assertions of the original ABox: the first disjunct captures the case where John was the unhappy person, and the second disjunct captures the case when Mary was the unhappy person. There is no update of the second line of the original ABox as this assertion is completely invalidated by the update. We shall later prove that the updated ABox cannot be expressed in $\mathcal{ALCO}$. This illustrates that, as was already noted in the introduction, the presence of nominals alone does not suffice to guarantee the existence of updates.

2.3 A Lower Bound on the Size of Updates

Later on, we will see that the existence and size of updated ABoxes strongly depends on the underlying description logic. In this section, we establish a general lower bound on the size of the updated ABox: even in propositional logic, updated ABoxes can become exponential in the size of the whole input, which consists of the original ABox and the update. At least, this holds unless every PTime algorithm is LogTime-parallelizable, i.e., unless the complexity classes PTime and NC are identical. As discussed by Papadimitriou in [16], this is believed to be similarly unlikely as PTime = NP. This lower bound on the size of updated ABoxes transfers to all DLs considered in this paper. Note that our result complements the one from [7], where it is shown that an exponential blowup of propositional updates cannot be avoided if *arbitrary formulas* are allowed as updates unless the first levels of the polynomial hierarchy collapses. Our argument uses a much more restricted form of updates (conjunctions of literals) and refers to a different complexity-theoretic assumption.

For the following definitions, we fix an individual name a. A *propositional ABox* $\mathcal{A}$ is of the form $\{C(a)\}$ with C a *propositional concept*, i.e., a concept that uses only the concept constructors $\neg$, $\sqcap$, and $\sqcup$. A *propositional update* $\mathcal{U}$ contains only assertions of the form $A(a)$ and $\neg A(a)$. Observe that propositional ABoxes and propositional updates are only allowed to refer to the single, fixed individual name a.

For the semantics, we fix a single individual x. Since we are dealing with propositional ABoxes and updates, we assume that interpretations do not inteprete role names, and that interpretation domains have only a single element x with $a^{\mathcal{I}} = x$. We introduce a couple of notions. For a concept C, let $\mathsf{C}(C)$ denote the set of concept names used in C. For an interpretation $\mathcal{I}$ and a set of concept names Γ, let $\mathcal{I}|_\Gamma$ denote the restriction of $\mathcal{I}$ that interpretes only the concept names in Γ. Let C be a concept and $\Gamma \subseteq \mathsf{C}(C)$. Then a concept D is called a *uniform* Γ*-interpolant* of C iff $\mathsf{C}(D) \subseteq \Gamma$ and $\{\mathcal{I}|_\Gamma \mid x \in C^{\mathcal{I}}\} = \{\mathcal{I}|_\Gamma \mid x \in D^{\mathcal{I}}\}$. It is easily seen that, for any propositional concept C and subset $\Gamma \subseteq \mathsf{C}(C)$, the uniform Γ-interpolant of C exists and is unique up to equivalence. The following lemma establishes a tight connection between uniform interpolants and propositional updates.

Lemma 3. *Let $\mathcal{A} = \{C(a)\}$ be a propositional ABox, $\mathcal{U}$ a propositional update, Γ the set of concept names in C not occurring in $\mathcal{U}$, $\widehat{C}$ the shortest uniform Γ-interpolant of C, and*

$$\mathcal{A}' = \{a : (\widehat{C} \sqcap \prod_{A(a) \in \mathcal{U}} A)\}.$$

Then we have the following:

(i) $\mathcal{A} * \mathcal{U} \equiv \mathcal{A}'$;

(ii) if $\mathcal{A} * \mathcal{U} \equiv \mathcal{A}''$, *then* $|\mathcal{A}'| \leq |\mathcal{U}| + |\mathcal{A}''|$.

Proof. Let $\mathcal{A} = \{C(a)\}$, $\mathcal{U}$, Γ, $\widehat{C}$, and $\mathcal{A}'$ be as in the lemma. To prove (i), we have to show that $\mathcal{A}'$ satisfies Conditions (U1) and (U2) from Definition 2:

(U1) Let $\mathcal{I}$, $\mathcal{I}'$ be interpretations such that $\mathcal{I} \models \mathcal{A}$ and $\mathcal{I} \Longrightarrow_{\mathcal{U}} \mathcal{I}'$. We have to show that $\mathcal{I}' \models \mathcal{U}$ and $\mathcal{I}' \models \widehat{C}(a)$. By definition of "$\Longrightarrow_{\mathcal{U}}$" and since the concept names in Γ do not appear in $\mathcal{U}$, we have $\mathcal{I}' \models \mathcal{U}$ and $\mathcal{I}|_\Gamma = \mathcal{I}'|_\Gamma$. The latter together with $\mathcal{I} \models C(a)$ and the fact that $\widehat{C}$ is the uniform Γ-interpolant of C yields that $\mathcal{I}' \models \widehat{C}(a)$ as required.

(U2) Let $\mathcal{I}'$ be an interpretation such that $\mathcal{I} \models \mathcal{A}'$. In particular, $\mathcal{I}' \models \widehat{C}(a)$. Since $\widehat{C}$ is the uniform Γ-interpolant of C, there is thus an interpretation $\mathcal{I}$ such that $a^{\mathcal{I}} \in C^{\mathcal{I}}$ and $\mathcal{I}|_\Gamma = \mathcal{I}'|_\Gamma$. We have to show that $\mathcal{I} \Longrightarrow_{\mathcal{U}} \mathcal{I}'$ and $\mathcal{I} \models \mathcal{A}$. The latter is clear since $a^{\mathcal{I}} \in C^{\mathcal{I}}$. For the former, we have to show that (i) $a^{\mathcal{I}} \in A^{\mathcal{I}'} \setminus A^{\mathcal{I}}$ implies $A(a) \in \mathcal{U}$, and (ii) $a^{\mathcal{I}} \in A^{\mathcal{I}} \setminus A^{\mathcal{I}'}$ implies $\neg A(a) \in \mathcal{U}$. For (i), let $a^{\mathcal{I}} \in A^{\mathcal{I}'} \setminus A^{\mathcal{I}}$. As $\mathcal{I}|_\Gamma = \mathcal{I}'|_\Gamma$, we have $A \notin \Gamma$. Therefore, A appears in $\mathcal{U}$. This can be either in the form $A(a)$ or $\neg A(a)$. As the second yields a contradiction to $a^{\mathcal{I}} \in A^{\mathcal{I}'}$ and $\mathcal{I}' \models \mathcal{U}$, we are done. Case (ii) is symmetric.

Now for Point (ii). Suppose $\mathcal{A} * \mathcal{U} \equiv \mathcal{A}''$. Then $\mathcal{A}'' = \{a : D\}$ for come concept D. We may assume that all concept names occuring in D occur in $\mathcal{A} \cup \mathcal{U}$ as well. Now, for all concept names A such that $a : A \in \mathcal{U}$ replace every occurence of A in D by $\top$. For $a : \neg A \in \mathcal{U}$, replace every occurence of A in D by $\bot$. Denote the resulting concept by D'. Then $\mathcal{A} * \mathcal{U} \equiv \{a : D'\} \cup \mathcal{U}$. Moreover, as D' and $\mathcal{U}$ do not have any concept names in common and $\mathcal{A}' \equiv \{a : D'\} \cup \mathcal{U}$, we have $\{a : \widehat{C}\} \equiv \{a : D\}$. It follows that D' is a Γ-interpolant for C. We derive $|\widehat{C}| \leq |D'|$ because $\widehat{C}$ is the shortest Γ-interpolant for C. But then

$$|\mathcal{A}'| \leq |\widehat{C}| + |\mathcal{U}| \leq |D'| + |\mathcal{U}| \leq |D| + |\mathcal{U}| \leq |\mathcal{A}''| + |\mathcal{U}|.$$

❑

It thus remains to show that the size of (smallest) uniform interpolants of propositional concepts is not bounded polynomially in the size of the interpolated concept unless PTIME = NC.

The size of uniform interpolants of propositional concepts is closely related to the relative succinctness of propositional logic (PL) formulas and Boolean circuits. We remind that both PL formulas and Boolean circuits compute Boolean functions and refer, e.g., to [16] for exact definitions. We use $|c|$ to denote the number of gates in the Boolean circuit c, and $|\varphi|$ to denote the length of the PL formula φ. It is known that, unless PTIME = NC, there exists no polynomial p such that every Boolean circuit c can be converted into a PL formula φ that computes the same function as c_i and satisfies $|\varphi| \leq p(|c_i|)$, see e.g. Exercise 15.5.4 of [16].

We show that non-existence of such a polynomial p implies that uniform interpolants are not bounded polynomially in the size of the interpolated concept. Take a Boolean circuit c with k inputs. Then c can be translated into a propositional concept D_c by introducing concept names $I_1, \ldots, I_k$ for the inputs and, additionally, one auxiliary concept name for the output of every gate. Let $\mathcal{G}$ be the set of concept names introduced for gate outputs, and let $O \in \mathcal{G}$ be the concept name for the output of the gate computing the final output of c. It is not difficult to see that this translation can be done such that there exists a polynomial q such that, for all Boolean circuits c,

(i) $|D_c| \leq p(|c|)$ and

(ii) for all interpretations $\mathcal{I}$ and all $x \in D_c^{\mathcal{I}}$, $x \in O^{\mathcal{I}}$ iff c outputs "true" on input $b_1, \ldots, b_k$, where $b_j = 1$ if $x \in I_j^{\mathcal{I}}$ and $b_j = 0$ otherwise.

Now, set $\Gamma := \mathcal{G} \setminus \{O\}$. Then the uniform Γ-interpolant $\widehat{D}_c$ of D_c also satisfies (ii). Thus, $\widehat{D}_c$ is a (notational variant of a) propositional logic formula computing the same Boolean function as c. If the size of (smallest) Γ-interpolants of propositional concepts was bounded polynomially in the size of the interpolated concept, we thus had obtained a contradiction to our assumption on the non-existence of the polynomial p. Together with Lemma 3, we obtain the following theorem.

Theorem 4. *Unless* PTIME = NC*, there exists no polynomial p such that, for all propositional ABoxes $\mathcal{A}$ and propositional updates $\mathcal{U}$, there exists a propositional ABox $\mathcal{A}'$ such that*

- $\mathcal{A} * \mathcal{U} \equiv \mathcal{A}'$ *and*
- $|\mathcal{A}'| \leq p(|\mathcal{A}| + |\mathcal{U}|)$.

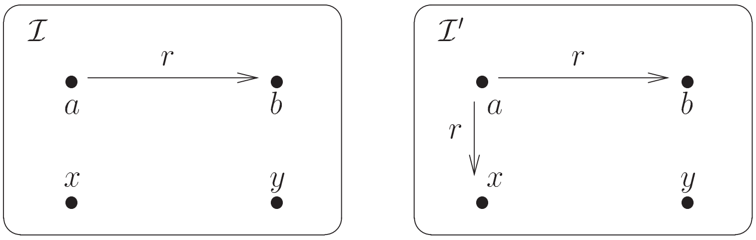

Figure 3: $\mathcal{I}$ and $\mathcal{I}'$

In the terminology of Cadoli et al. [7], this result states that the common update operators for propositional theories are not logically compactable even for updates with conjunctions of literals (unless PTIME = NC). Since the additional constructors do not add to Boolean expressivity, it is not difficult to prove that Theorem 4 carries over to all description logics considered in this paper.

3 Description Logics without Updates

We say that a description logic $\mathcal{L}$ *has ABox updates* iff, for every ABox $\mathcal{A}$ formulated in $\mathcal{L}$ and every update $\mathcal{U}$, there exists an ABox $\mathcal{A}'$ formulated in $\mathcal{L}$ such that $\mathcal{A}*\mathcal{U} \equiv \mathcal{A}'$. In this section, we show that a lot of basic DLs are lacking ABox updates.

3.1 Updates in $\mathcal{ALC}$

We analyze the basic description logic $\mathcal{ALC}$ and show that it does not have ABox updates. In particular, we consider the following combination of original ABox $\mathcal{A}$, update $\mathcal{U}$, and updated ABox $\mathcal{A}'$. Note that $\mathcal{A}$ is formulated in $\mathcal{ALC}$, and $\mathcal{A}'$ is formulated in $\mathcal{ALCO}$.

Lemma 5. *Let* $\mathcal{A} = \{\forall r.A(a)\}$, $\mathcal{U} := \{\neg A(b)\}$, *and*

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

Then $\mathcal{A} * \mathcal{U} \equiv \mathcal{A}'$.

Lemma 5 is readily checked by verifying that Conditions (U1) and (U2) of Definition 2 are satisfied.

To show that $\mathcal{ALC}$ does not have ABox updates, it suffices to prove that there is no $\mathcal{ALC}$-ABox equivalent to the $\mathcal{ALCO}$-ABox $\mathcal{A}'$. Consider the interpretations $\mathcal{I}$ and $\mathcal{I}'$ displayed in Figure 3. We assume that the individual names a and b are mapped to the individuals of the same name as shown in the figure. Moreover, all other individual names are mapped to the individual y, and every concept name is interpreted as the empty set. Clearly, $\mathcal{I} \models \mathcal{A}'$ and $\mathcal{I}' \not\models \mathcal{A}'$. To show that no $\mathcal{ALC}$-ABox is equivalent to $\mathcal{A}'$, it thus suffices to prove that $\mathcal{ALC}$-ABoxes cannot distinguish $\mathcal{I}$ and $\mathcal{I}'$: for every $\mathcal{ALC}$ ABox, we have $\mathcal{I} \models \mathcal{A}'$ iff $\mathcal{I}' \models \mathcal{A}'$. We first establish the following lemma.

Lemma 6. *For all $\mathcal{ALC}$-concepts C and all individual names α, we have $\mathcal{I} \models C(\alpha)$ iff $\mathcal{I}' \models C(\alpha)$.*

Proof. The truth of an assertion $C(\alpha)$, C an $\mathcal{ALC}$-concept, in a model $\mathcal{J}$ only depends on the set of points reachable from $\alpha^{\mathcal{J}}$ using paths along the relations $r^{\mathcal{J}}$, where r occurs in C. Therefore, the lemma is clear for $\alpha \neq a$. For $\alpha = a$, the lemma can be proved by observing that the submodel of $\mathcal{I}$ induced by $\{a^{\mathcal{I}}, b^{\mathcal{I}}\}$ is bisimilar to the submodel of $\mathcal{I}'$ induced by $\{a^{\mathcal{I}'}, b^{\mathcal{I}'}, x\}$, see [10] for a discussion of the notion of bisimulation for $\mathcal{ALC}$. Thus, for $\alpha = a$, the lemma is an immediate consequence of the fact that the extension $C^{\mathcal{I}}$ of $\mathcal{ALC}$ concepts C is preserved under bisimulations. ❑

Lemma 7. *There exists no $\mathcal{ALC}$-ABox that is equivalent to the $\mathcal{ALCO}$-ABox $\mathcal{A}' = \{\neg A(b), \forall r.(A \sqcup \{b\})(a)\}$.*

Proof. Assume to the contrary of what is to be shown that there exists an $\mathcal{ALC}$-ABox $\mathcal{B}$ that is equivalent to $\mathcal{A}'$. Then $\mathcal{I} \models \mathcal{B}$ and $\mathcal{I}' \not\models \mathcal{B}$. We show that, for all assertions $\varphi \in \mathcal{B}$, we have $\mathcal{I}' \models \varphi$, thus obtaining a contradiction to $\mathcal{I}' \not\models \mathcal{B}$. First, let φ be a (positive or negative) role assertion. Then $\mathcal{I}' \models \varphi$ is a consequence of $\mathcal{I} \models \varphi$ and the fact that $\mathcal{I}$ and $\mathcal{I}'$ satisfy exactly the same role assertions. Now, let φ be a concept assertion. Then, $\mathcal{I}' \models \varphi$ is a consequence of $\mathcal{I} \models \varphi$ and Lemma 6. ❑

We have thus established the following result:

Theorem 8. *$\mathcal{ALC}$ does not have ABox updates.*

Note that Theorem 8 even applies to the case where the update contains only concept assertions, but no role assertions. The fact that the updated ABox $\mathcal{A}'$ used in this section is actually an $\mathcal{ALCO}$-ABox may give rise to the conjecture that adding nominals to $\mathcal{ALC}$ recovers the existence of updates. Unfortunately, as shown in the following section, this is not the case.

3.2 Updates in $\mathcal{ALCO}$

We consider the DL $\mathcal{ALCO}$, which is obtained by extending $\mathcal{ALC}$ with nominals, and show that $\mathcal{ALCO}$ does not have ABox updates. More precisely, we proceed

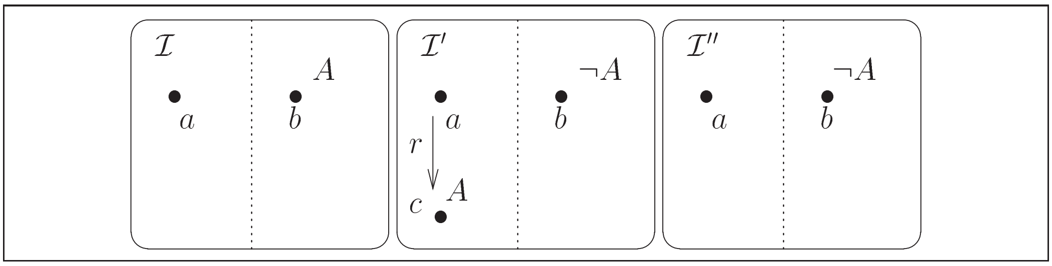

Figure 4: $\mathcal{I}$, $\mathcal{I}'$ and $\mathcal{I}''$

in two steps: we first give a relatively straightforward proof of the non-existence of updated ABoxes in $\mathcal{ALCO}$ that relies on the use of role assertions in updates. This proof raises the question whether the restriction of updates to only concept assertions recovers the existence of updates. In the second step, we answer this question to the negative by using a slightly more complex construction.

For presenting the counterexample to the existence of ABox updates in $\mathcal{ALCO}$, it is convenient to describe the updated ABox in $\mathcal{ALCO}^{@}$, the extension of $\mathcal{ALCO}$ with the @ constructor. Note that the original ABox is even formulated in $\mathcal{ALC}$.

Lemma 9. *Let* $\mathcal{A} = \{\exists r.A(a)\}$, $\mathcal{U} := \{\neg r(a, b)\}$, *and*

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

Then $A * \mathcal{U} \equiv \mathcal{A}'$.

It is not hard to see that $\mathcal{A}'$ satisfies Conditions (U1) and (U2) of Definition 2. We now show that there exists no $\mathcal{ALCO}$-ABox that is equivalent to the $\mathcal{ALCO}^{@}$-ABox $\mathcal{A}'$. As in the previous section, it follows that $\mathcal{ALCO}$ does not have ABox updates.

Consider the interpretations $\mathcal{I}$, $\mathcal{I}'$ and $\mathcal{I}''$ depicted in Figure 4. We assume that the individual names a, b, and c are mapped to the individuals of the same name, and that all other individual names are mapped to the individual c. Moreover, the concept name A is interpreted as shown in the figure and all other concept names are interpreted as the empty set in all three interpretations. It can easily be checked that $\mathcal{I} \models \mathcal{A}'$, $\mathcal{I}' \models \mathcal{A}'$ and $\mathcal{I}'' \not\models \mathcal{A}'$. We will show that, if an $\mathcal{ALCO}$-ABox $\mathcal{B}$ is equivalent to $\mathcal{A}'$, then $\mathcal{I}'' \models \mathcal{B}$, which is a contradiction. First, we prove the following lemma:

Lemma 10. *For all $\mathcal{ALCO}$-concepts C and all individual names $\alpha \neq b$, we have $\mathcal{I} \models C(\alpha)$ iff $\mathcal{I}'' \models C(\alpha)$, and $\mathcal{I}' \models C(b)$ iff $\mathcal{I}'' \models C(b)$.*

Proof. The truth of an assertion $C(\alpha)$, C an $\mathcal{ALCO}$-concept, in a model $\mathcal{J}$ only depends on the set of points reachable from $\alpha^{\mathcal{J}}$ by paths along relations $r^{\mathcal{J}}$, where r occurs in C. The lemma follows immediately from this observation. ❑

The next lemma shows that $\mathcal{A}'$ in Lemma 9 cannot be formulated in $\mathcal{ALCO}$.

Lemma 11. *There is no $\mathcal{ALCO}$-ABox that is equivalent to the $\mathcal{ALCO}^{@}$-ABox $\mathcal{A}' = \{(\exists r.(A \sqcap \neg\{b\}) \sqcup @_b A)(a), \neg r(a,b)\}$.*

Proof. Assume there is an $\mathcal{ALCO}$-ABox $\mathcal{B}$ that is equivalent to $\mathcal{A}'$. Then $\mathcal{I} \models \mathcal{B}$, $\mathcal{I}' \models \mathcal{B}$, and $\mathcal{I}'' \not\models \mathcal{B}$. We show that, for all assertions $\varphi \in \mathcal{B}$, we have $\mathcal{I}'' \models \varphi$, thus obtaining a contradiction to $\mathcal{I}'' \not\models \mathcal{B}$. First, $\mathcal{B}$ does not contain any positive role assertion since $\mathcal{I} \models \mathcal{B}$ and $\mathcal{I}$ does not satisfy any positive role assertions. Second, if φ is a negative role assertion, then $\mathcal{I}'' \models \varphi$ since $\mathcal{I}''$ satisfies all negative role assertions. Finally, let φ be a concept assertion. Then, $\mathcal{I}'' \models \varphi$ is a consequence of $\mathcal{I} \models \varphi$, $\mathcal{I}' \models \varphi$, and Lemma 10. ❏

The proof also shows that $\mathcal{ALC}$ does not have ABox updates even if we restrict ourselves to updates containing only role assertions, thus complementing the result from Section 3.1 where $\mathcal{ALC}$ updates with only concept assertions are considered.

As stated initially, the above proof raises the question whether or not restricting updates to concept assertions regains the existence of updated ABoxes in $\mathcal{ALCO}$. We answer this question to the negative. The following counterexample is quite similar to the example for ABox updates given in Section 2.2:

Lemma 12. *Let $\mathcal{A} = \{\exists r.A(a), \exists r.A(a'), r(b,c), \neg A \sqcup \forall r.(\{c\} \rightarrow \neg A)(b)\}$, $\mathcal{U} := \{\neg A(b), \neg A(c)\}$, and $\mathcal{A}' = \{C'(a), r(b,c), \neg A(b), \neg A(c)\}$ with*

$$\begin{aligned} C' = & \big(@_a \exists r.(A \sqcup \{b\}) \sqcap @_{a'} \exists r.(A \sqcup \{b\})\big) \sqcup \\ & \big(@_a \exists r.(A \sqcup \{c\}) \sqcap @_{a'} \exists r.(A \sqcup \{c\})\big). \end{aligned}$$

Then $\mathcal{A} * \mathcal{U} \equiv \mathcal{A}'$.

By verifying Conditions (U1) and (U2) in Definition 2, one can check that $\mathcal{A}'$ is indeed the result of updating $\mathcal{A}$ with $\mathcal{U}$. Intuitively, the ABox assertions $r(b,c)$ and $\neg A \sqcup \forall r.(\{c\} \rightarrow \neg A)(b)$ in $\mathcal{A}$ enforce that, in every model $\mathcal{I}$ of $\mathcal{A}$, $b^{\mathcal{I}} \notin A^{\mathcal{I}}$ or $c^{\mathcal{I}} \notin A^{\mathcal{I}}$. The assertion $C'(a)$ represents the update of the assertions $\exists r.A(a)$ and $\exists r.A(a')$ in $\mathcal{A}$. The first disjunct captures the case where $b \in A^{\mathcal{I}}$ and $c \notin A^{\mathcal{I}}$, and the second disjunct captures the case where $b \notin A^{\mathcal{I}}$ and $c \in A^{\mathcal{I}}$. In the remaining case $b \notin A^{\mathcal{I}}$ and $c \notin A^{\mathcal{I}}$, the update of the mentioned assertions is $@_a \exists r.A \sqcap @_{a'} \exists r.A$. A corresponding disjunct is not needed since it would imply the first two disjuncts. The assertion $\forall r.(\{c\} \rightarrow \neg A)(b)$ can simply be dropped since all the information it provides is invalidated by the update.

In order to show that $\mathcal{ALCO}$ does not have ABox updates even if only concept assertions are allowed in updates, we prove that there is no $\mathcal{ALCO}$-ABox that is equivalent to $\mathcal{A}'$.

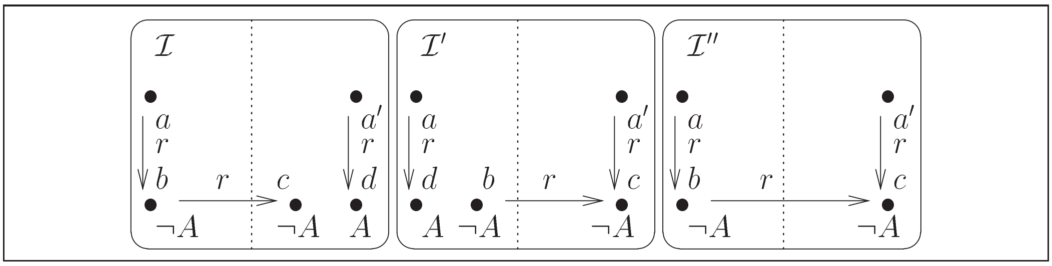

Figure 5: $\mathcal{I}$, $\mathcal{I}'$ and $\mathcal{I}''$

Consider the interpretations $\mathcal{I}$, $\mathcal{I}'$ and $\mathcal{I}''$ depicted in Figure 5. We assume that the individual names a, a', b, c, and d are mapped to the individuals of the same name, and that all other individual names are mapped to the individual d. Moreover, $A^{\mathcal{I}} = A^{\mathcal{I}'} = A^{\mathcal{I}''} = \{d\}$ and all other concept names are interpreted as the empty set. Clearly, $\mathcal{I} \models \mathcal{A}'$, $\mathcal{I}' \models \mathcal{A}'$, but $\mathcal{I}'' \not\models \mathcal{A}'$.

Lemma 13. *For all $\mathcal{ALCO}$-concepts C and all individuals $\alpha \neq a'$, we have $\mathcal{I} \models C(\alpha)$ iff $\mathcal{I}'' \models C(\alpha)$, and $\mathcal{I}' \models C(a')$ iff $\mathcal{I}'' \models C(a')$.*

Proof. Recall from the proof of Lemma 10 that the truth of an assertion $C(\alpha)$, C an $\mathcal{ALCO}$-concept, in a model $\mathcal{J}$ only depends on the set of points reachable from $\alpha^{\mathcal{J}}$ by paths along relations $r^{\mathcal{J}}$, where r occurs in C. Again, the lemma follows immediately from this observation. ❑

Thus, we are ready to prove that $\mathcal{A}'$ is not expressible in $\mathcal{ALCO}$.

Lemma 14. *There is no $\mathcal{ALCO}$-ABox that is equivalent to the $\mathcal{ALCO}^{@}$-ABox $\mathcal{A}'$ from Lemma 12.*

Proof. Assume there is some $\mathcal{ALCO}$-ABox $\mathcal{B}$ with $\mathcal{A}' \equiv \mathcal{B}$. Then $\mathcal{I} \models \mathcal{A}'$, $\mathcal{I}' \models \mathcal{A}'$ and $\mathcal{A}' \equiv \mathcal{B}$ implies that $\mathcal{I} \models \mathcal{B}$ and $\mathcal{I}' \models \mathcal{B}$. We show that $\mathcal{I}''$ satisfies every assertion in $\mathcal{B}$, contradicting the facts that $\mathcal{I}'' \not\models \mathcal{A}'$ and $\mathcal{A}' \equiv \mathcal{B}$. We make a case distinction according to the type of assertion:

- φ is a concept assertion. Since $\varphi \in \mathcal{B}$, we have $\mathcal{I} \models \varphi$ and $\mathcal{I}' \models \varphi$. Thus, Lemma 13 implies $\mathcal{I}'' \models \varphi$.
- φ is a positive role assertion. Then $\varphi = r(b, c)$ since, otherwise, we have $\mathcal{I} \not\models \varphi$ or $\mathcal{I}' \not\models \varphi$ contradicting $\mathcal{I} \models \mathcal{B}$ and $\mathcal{I}' \models \mathcal{B}$. Clearly, $\varphi = r(b, c)$ implies $\mathcal{I}'' \models \varphi$.
- φ is a negative role assertion. Since $\varphi \in \mathcal{B}$, we have $\mathcal{I} \models \varphi$ and $\mathcal{I}' \models \varphi$. Assume to the contrary of what is to be shown that $\mathcal{I}'' \not\models \varphi$. Then $\varphi \in \{\neg r(a, b), \neg r(a', c), \neg r(b, c)\}$. However, in each of the three cases we obtain a contradiction to $\mathcal{I} \models \varphi$ or $\mathcal{I}' \models \varphi$. Hence, $\mathcal{I}'' \models \varphi$.

❑

Summing up, we obtain the following result:

Theorem 15. *$\mathcal{ALCO}$ is lacking ABox updates, even if updates contain only concept assertions or only role assertions.*

3.3 Updates in $\mathcal{ALC}^{@}$ and Boolean ABoxes in $\mathcal{ALC}$

Due to the fact that, in the previous section, the ABoxes $\mathcal{A}'$ are expressed in $\mathcal{ALCO}^{@}$, one may conjecture that the existence of updated ABoxes in $\mathcal{ALCO}$ is recovered by adding the @ constructor. We will later see that this is indeed the case. However, one may even reckon that adding *only* the @ constructor to $\mathcal{ALC}$ does suffice to guarantee the existence of updated ABoxes. In this section, we show that this is not the case. Indeed, we even show a stronger result related to Boolean ABoxes.

A *Boolean ABox* is a finite set of *Boolean ABox assertions*, i.e., Boolean combinations of ABox assertions expressed in terms of the connectives $\wedge$ and $\vee$. We do not need to explicitly introduce negation since we admit negated role assertions and concept negation is contained in every DL considered in this paper. For example, the following is a Boolean ABox:

$$\{B(a), (A(a) \wedge r(a,b)) \vee \neg\exists s.A(b)\}.$$

An interpretation $\mathcal{I}$ satisfies a Boolean ABox $\mathcal{A}$ if every Boolean ABox assertion in $\mathcal{A}$ evaluates to true. There exists a rather close connection between the @ constructor and Boolean ABoxes:

Lemma 16.

1. *For every non-Boolean $\mathcal{ALC}^{@}$-ABox, there exists an equivalent Boolean $\mathcal{ALC}$-ABox;*

2. *for every Boolean $\mathcal{ALCO}$-ABox, there exists an equivalent non-Boolean $\mathcal{ALCO}^{@}$-ABox.*

Proof. Concerning Point 1, let $\mathcal{A}$ be a non-Boolean $\mathcal{ALC}^{@}$-ABox, and let $C(a)$ be an assertion from $\mathcal{A}$ such that $@_bD$ is a subconcept of C. Then the ABox $\mathcal{A}'$ is obtained from $\mathcal{A}$ by replacing the assertion $C(a)$ with $(D(b) \wedge C[\top/@_bD]) \vee (\neg D(b) \wedge C[\bot/@_bD])$, where $C[X/@_bD]$ denotes the concept obtained from C by replacing all occurrences of $@_bD$ with X. It is readily checked that $\mathcal{A}'$ is equivalent to $\mathcal{A}$. By repeating this replacement, we will eventually obtain a Boolean $\mathcal{ALC}$-ABox.

Concerning Point 2, define a mapping $\cdot^*$ from ABox assertions to $\mathcal{ALCO}^@$-concepts as follows:

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

Now every Boolean ABox assertion ϕ can be converted into an $\mathcal{ALCO}^@$-concept ϕ^* by replacing $\wedge$ with $\sqcap$, $\vee$ with $\sqcup$, and every assertion ψ with ψ^*. Now, let $\mathcal{A} = \{\phi_1, \ldots, \phi_n\}$ be a Boolean $\mathcal{ALCO}$-ABox. Define a non-Boolean $\mathcal{ALCO}^@$-ABox $\mathcal{A}' := \{(\phi_1^* \sqcap \cdots \sqcap \phi_n^*)(a)\}$, where a is an arbitrary individual name. It is readily checked that $\mathcal{A}'$ is equivalent to $\mathcal{A}$. ❑

Thus, non-Boolean $\mathcal{ALCO}^@$-ABoxes have exactly the same expressive power as Boolean $\mathcal{ALCO}$-ABoxes. Note that the same does not hold for $\mathcal{ALC}$: while every non-Boolean $\mathcal{ALC}^@$-ABox can be translated into an equivalent Boolean $\mathcal{ALC}$-ABox, there are Boolean $\mathcal{ALC}$-ABoxes for which no equivalent non-Boolean $\mathcal{ALC}^@$-ABox exists. For example, it is relatively easy to prove that the Boolean $\mathcal{ALC}$-ABox $\{A(a) \vee r(a,b)\}$ has this property.

Since, for $\mathcal{ALC}$, Boolean ABoxes are more expressive than the @ constructor, we prove that $\mathcal{ALC}$ does not have ABox updates, even if we allow Boolean ABoxes for the updated ABox.

Theorem 17. *There exists an $\mathcal{ALC}$-ABox $\mathcal{A}$ and an update $\mathcal{U}$ such that there exists no Boolean $\mathcal{ALC}$-ABox $\mathcal{A}'$ with $\mathcal{A} * \mathcal{U} \equiv \mathcal{A}'$.*

Proof. Consider the $\mathcal{ALC}$-ABox $\mathcal{A}$, the update $\mathcal{U}$, and the $\mathcal{ALCO}$-ABox $\mathcal{A}'$ given in Lemma 5. To prove Theorem 17, it is enough to show that there is no Boolean $\mathcal{ALC}$-ABox that is equivalent to $\mathcal{A}'$.

Assume that there exists a Boolean $\mathcal{ALC}$-ABox $\mathcal{B}$ with $\mathcal{A}' \equiv \mathcal{B}$. We can assume w.l.o.g. that $\mathcal{B}$ is in disjunctive normal form, i.e., that

$$\mathcal{B} = \bigwedge \mathcal{B}_0 \vee \cdots \vee \bigwedge \mathcal{B}_{n-1},$$

where $\mathcal{B}_1, \ldots, \mathcal{B}_n$ are $\mathcal{ALC}$-ABoxes. Now take the interpretations $\mathcal{I}$ and $\mathcal{I}'$ displayed in Figure 3. Recall that $\mathcal{I} \models \mathcal{A}'$ and $\mathcal{I}' \not\models \mathcal{A}'$. Then, $\mathcal{A}' \equiv \mathcal{B}$ and $\mathcal{I} \models \mathcal{A}'$ imply that there is an $i < n$ such that $\mathcal{I} \models \mathcal{B}_i$. Since $\mathcal{I}' \not\models \mathcal{A}'$, we have $\mathcal{I}' \not\models \mathcal{B}_i$. We can proceed as in the proof of Lemma 7 to show that $\mathcal{I}' \models \varphi$ for every $\varphi \in \mathcal{B}_i$, thus obtaining a contradiction to $\mathcal{I}' \not\models \mathcal{B}_i$. ❑

By Lemma 16, we obtain the following corollary.

Corollary 18. *$\mathcal{ALC}^@$ does not have ABox updates.*

Observe that both Theorem 17 and Corollary 18 remain true if we restrict updates to only concept assertions.

4 Computing Updates in $\mathcal{ALCQIO}^@$

The results obtained in the previous section imply that, if an extension of $\mathcal{ALC}$ does not allow to express nominals and the @ constructor, then we cannot hope that it has ABox updates. In this section, we show that, for the common extensions of $\mathcal{ALC}$ introduced in Section 2.1, adding nominals and the @ constructor suffices to have ABox updates. More presicely, we prove that the expressive DL $\mathcal{ALCQIO}^@$ has ABox updates, and show that the proof is easily adapted to the fragments of $\mathcal{ALCQIO}^@$ obtained by dropping number restrictions, inverse roles, or both.

Our construction of updated ABoxes is an extension of the corresponding construction for propositional logic described in [26], and proceeds as follows. First, we consider *updates of concepts* on the level of interpretations. More precisely, we show how to convert a concept C and an update $\mathcal{U}$ into a concept $C^{\mathcal{U}}$ such that the following holds: for all interpretations $\mathcal{I}$ and $\mathcal{I}'$ such that $\mathcal{I}$ satisfies *no* assertion in $\mathcal{U}$ and $\mathcal{I} \Longrightarrow_{\mathcal{U}} \mathcal{I}'$, we have $C^{\mathcal{I}} = (C^{\mathcal{U}})^{\mathcal{I}'}$ $(*)$. The limitation that $C^{\mathcal{U}}$ satisfies $(*)$ only if $\mathcal{I}$ satisfies no assertion in $\mathcal{U}$ can be overcome by replacing $C^{\mathcal{U}}$ with $C^{\mathcal{U}'}$, where $\mathcal{U}'$ is the set of those assertions in $\mathcal{U}$ that are violated in $\mathcal{I}$. Obviously, the translation $C^{\mathcal{U}}$ will be used to update concept assertions in ABoxes (role assertions are very easy to deal with). However, we are confronted with the problem that ABoxes have many different models, and these models can violate *different* subsets of the update $\mathcal{U}$. Hence, there is no unique way of moving from $C^{\mathcal{U}}$ to $C^{\mathcal{U}'}$ as described above. The solution is to produce an updated ABox for each subset $\mathcal{U}' \subseteq \mathcal{U}$ separately, and then simply take the disjunction.

We first introduce a bit of notation. For an ABox $\mathcal{A}$, we use $\mathsf{Obj}(\mathcal{A})$ to denote the set of individual names in $\mathcal{A}$, and $\mathsf{sub}(\mathcal{A})$ to denote the set of subconcepts of the concepts occurring in $\mathcal{A}$. For an update $\mathcal{U}$, we use $\mathcal{I}^{\mathcal{U}}$ to denote the (unique) interpretation satisfying $\mathcal{I} \Longrightarrow_{\mathcal{U}} \mathcal{I}^{\mathcal{U}}$. We use $\mathcal{U}^{\neg}$ to denote $\{\neg\varphi \mid \varphi \in \mathcal{U}\}$. The inductive translation that takes a concept C and an update U to a concept C^U as explained above is given in Figure 6.

Lemma 19. *Let $\mathcal{U}$ be an update and C a $\mathcal{ALCQIO}^@$-concept. For every interpretation $\mathcal{I}$ with $\mathcal{I} \models \mathcal{U}^{\neg}$ and every individual name a, we have $\mathcal{I} \models C(a)$ iff $\mathcal{I}^{\mathcal{U}} \models C^{\mathcal{U}}(a)$*

Proof. The following is an immediate consequence of the definition of $\mathcal{I}^{\mathcal{U}}$:

Claim. If $\mathcal{I} \models \mathcal{U}^{\neg}$, then, for all $x, y \in \Delta^{\mathcal{I}}$ and role names r, we have $(x, y) \in r^{\mathcal{I}}$ iff one of the following holds:

$$
\begin{array}{rcl}
A^{\mathcal{U}} & = & \big(A \sqcup \bigsqcup_{\neg A(a)\in\mathcal{U}} \{a\}\big) \sqcap \neg(\bigsqcup_{A(a)\in\mathcal{U}} \{a\}) \\
\{a\}^{\mathcal{U}} & = & \{a\} \\
(@_a C)^{\mathcal{U}} & = & @_a C^{\mathcal{U}} \\
(\neg C)^{\mathcal{U}} & = & \neg C^{\mathcal{U}} \\
(C \sqcap D)^{\mathcal{U}} & = & C^{\mathcal{U}} \sqcap D^{\mathcal{U}} \\
(C \sqcup D)^{\mathcal{U}} & = & C^{\mathcal{U}} \sqcup D^{\mathcal{U}} \\
(\geqslant m\ r\ C)^{\mathcal{U}} & = & (\displaystyle\bigsqcap_{a\in\mathsf{Obj}(\mathcal{U})} \neg\{a\} \sqcap (\geq m\ r\ C^{\mathcal{U}})) \\
 & & \sqcup \displaystyle\bigsqcup_{a\in\mathsf{Obj}(\mathcal{U})} \Big(\{a\} \sqcap \bigsqcup_{m_1+m_2+m_3=m} \big((\geq m_1\ r \bigsqcap_{b\in\mathsf{Obj}(\mathcal{U})} \neg\{b\} \sqcap C^{\mathcal{U}}) \\
 & & \qquad \sqcap(\geq m_2\ r \displaystyle\bigsqcup_{b\in\mathsf{Obj}(\mathcal{U}),r(a,b)\notin\mathcal{U}} \{b\} \sqcap C^{\mathcal{U}}) \\
 & & \qquad \sqcap \displaystyle\bigsqcup_{S\subseteq\{b\mid\neg r(a,b)\in\mathcal{U}\},\#S=m_3} \bigsqcap_{b\in S} @_b C^{\mathcal{U}}\big)\Big) \\
(\leqslant m\ r\ C)^{\mathcal{U}} & = & (\displaystyle\bigsqcap_{a\in\mathsf{Obj}(\mathcal{U})} \neg\{a\} \sqcap (\leq m\ r\ C^{\mathcal{U}})) \\
 & & \sqcup \displaystyle\bigsqcup_{a\in\mathsf{Obj}(\mathcal{U})} \Big(\{a\} \sqcap \bigsqcup_{m_1+m_2+m_3=m} \big((\leq m_1\ r \bigsqcap_{b\in\mathsf{Obj}(\mathcal{U})} \neg\{b\} \sqcap C^{\mathcal{U}}) \\
 & & \qquad \sqcap(\leq m_2\ r \displaystyle\bigsqcup_{b\in\mathsf{Obj}(\mathcal{U}),r(a,b)\notin\mathcal{U}} \{b\} \sqcap C^{\mathcal{U}}) \\
 & & \qquad \sqcap \displaystyle\bigsqcap_{S\subseteq\{b\mid\neg r(a,b)\in\mathcal{U}\},\#S=m_3+1} \bigsqcup_{b\in S} \neg @_b C^{\mathcal{U}}\big)\Big)
\end{array}
$$

Figure 6: Constructing $C^{\mathcal{U}}$

- $x \neq a^{\mathcal{I}}$ for all $a \in \mathsf{Obj}(\mathcal{U})$ and $(x, y) \in r^{\mathcal{I}^{\mathcal{U}}}$;
- $x = a^{\mathcal{I}}$ for an $a \in \mathsf{Obj}(\mathcal{U})$ and
  - $y \neq b^{\mathcal{I}}$ for all $b \in \mathsf{Obj}(\mathcal{U})$ and $(x, y) \in r^{\mathcal{I}^{\mathcal{U}}}$,
  - or $y = b^{\mathcal{I}}$ for a $b \in \mathsf{Obj}(\mathcal{U})$ such that $r(a, b) \notin \mathcal{U}$ and $(x, y) \in r^{\mathcal{I}^{\mathcal{U}}}$,
  - or $y = b^{\mathcal{I}}$ for a $b \in \mathsf{Obj}(\mathcal{U})$ such that $\neg r(a, b) \in \mathcal{U}$.

Let $\mathcal{I}$ be an interpretation such that $\mathcal{I} \models \mathcal{U}^{\neg}$ and let $E \in \mathsf{sub}(\mathcal{A})$. By structural induction on E, we show that $(E^{\mathcal{U}})^{\mathcal{I}^{\mathcal{U}}} = E^{\mathcal{I}}$. As $\mathcal{I}$ and $\mathcal{I}^{\mathcal{U}}$ interpret individuals in the same way, this implies Lemma 19.

- $E = \{a\}$: this case is trivial since $\mathcal{I}$ and $\mathcal{I}^{\mathcal{U}}$ interpret individuals in the same way.

- $E = A$, for A a concept name: then

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

  since $A^{\mathcal{I}} \cap \bigcup_{A(a)\in\mathcal{U}} \{a^{\mathcal{I}}\} = \emptyset$ and $\bigcup_{\neg A(a)\in\mathcal{U}} \{a^{\mathcal{I}}\} \subseteq A^{\mathcal{I}}$ due to $\mathcal{I} \models \mathcal{U}^{\neg}$.

- $E = @_a C$: $((@_a C)^{\mathcal{U}})^{\mathcal{I}^{\mathcal{U}}} = (@_a C^{\mathcal{U}})^{\mathcal{I}^{\mathcal{U}}} = (@_a C)^{\mathcal{I}}$ since $(C^{\mathcal{U}})^{\mathcal{I}^{\mathcal{U}}} = C^{\mathcal{I}}$ and $\mathcal{I}$ and $\mathcal{I}^{\mathcal{U}}$ interpret individuals in the same way.

- The cases $E = \neg C$, $E = C \sqcup D$ and $E = C \sqcap D$ are straightforward and left to the reader.

- $E = (\geq m\ r\ C)$: we have $x \in ((\geq m\ r\ C)^{\mathcal{U}})^{\mathcal{I}^{\mathcal{U}}}$ iff one of the following holds:

  $x \in \big(\neg \bigsqcup_{a\in\mathsf{Obj}(\mathcal{U})} \{a\}\big)^{\mathcal{I}^{\mathcal{U}}}$ and $\#\{y \mid (x,y) \in r^{\mathcal{I}^{\mathcal{U}}} \wedge y \in (C^{\mathcal{U}})^{\mathcal{I}^{\mathcal{U}}}\} \geq m$

  or

  $x = a^{\mathcal{I}^{\mathcal{U}}}$, for an $a \in \mathsf{Obj}(\mathcal{U})$ and there are $m_1, m_2, m_3 \geq 0$ such that $m_1 + m_2 + m_3 = m$ and

  - $\#\{y \mid (x,y) \in r^{\mathcal{I}^{\mathcal{U}}} \wedge y \in \big(\neg \bigsqcup_{b\in\mathsf{Obj}(\mathcal{U})} \{b\}\big)^{\mathcal{I}^{\mathcal{U}}} \cap (C^{\mathcal{U}})^{\mathcal{I}^{\mathcal{U}}}\} \geq m_1$,
  - $\#\{y \mid (x,y) \in r^{\mathcal{I}^{\mathcal{U}}} \wedge y \in \bigcup_{b\in\mathsf{Obj}(\mathcal{U}), r(a,b)\notin\mathcal{U}} \{b\}^{\mathcal{I}^{\mathcal{U}}} \cap (C^{\mathcal{U}})^{\mathcal{I}^{\mathcal{U}}}\} \geq m_2$ and
  - $\#\{b \mid \neg r(a,b) \in \mathcal{U} \wedge b^{\mathcal{I}^{\mathcal{U}}} \in (C^{\mathcal{U}})^{\mathcal{I}^{\mathcal{U}}}\} \geq m_3$.

  By induction, we have that $(C^{\mathcal{U}})^{\mathcal{I}^{\mathcal{U}}} = C^{\mathcal{I}}$. Thus, using the claim above, we obtain that $x \in ((\geq m\ r\ C)^{\mathcal{U}})^{\mathcal{I}^{\mathcal{U}}}$ iff

  $x \in \big(\neg \bigsqcup_{a\in\mathsf{Obj}(\mathcal{U})} \{a\}\big)^{\mathcal{I}}$ and $\#\{y \mid (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \geq m$

  or

  $x = a^{\mathcal{I}}$, for an $a \in \mathsf{Obj}(\mathcal{U})$ and there are $m_1, m_2, m_3 \geq 0$ such that $m_1 + m_2 + m_3 = m$ and

– $\#\{y \mid (x,y) \in r^{\mathcal{I}} \wedge y \in \big(\neg \bigsqcup_{b \in \mathsf{Obj}(\mathcal{U})} \{b\}\big)^{\mathcal{I}} \cap C^{\mathcal{I}}\} \geq m_1$,

– $\#\{y \mid (x,y) \in r^{\mathcal{I}} \wedge y \in \bigcup_{b \in \mathsf{Obj}(\mathcal{U}), r(a,b) \notin \mathcal{U}} \{b\}^{\mathcal{I}} \cap C^{\mathcal{I}}\} \geq m_2$ and

– $\#\{y \mid (x,y) \in r^{\mathcal{I}} \wedge y \in \bigcup_{\neg r(a,b) \in \mathcal{U}} \{b\}^{\mathcal{I}} \cap C^{\mathcal{I}}\} \geq m_3$.

Further, by the claim above, this is equivalent to

$$x \in \big(\neg \bigsqcup_{a \in \mathsf{Obj}(\mathcal{U})} \{a\}\big)^{\mathcal{I}} \text{ and } \#\{y \mid (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \geq m$$

or

$$x \in \big(\bigsqcup_{a \in \mathsf{Obj}(\mathcal{U})} \{a\}\big)^{\mathcal{I}} \text{ and } \#\{y \mid (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \geq m.$$

But this is equivalent to $x \in (\geq m\ r\ C)^{\mathcal{I}}$.

- The case $E = (\leq m\ r\ C)$ is proved similarly to the previous case.

❏

We now extend the update of concepts to the update of ABoxes, while still remaining on the level of interpretations. Let $\mathcal{A}$ be an ABox and $\mathcal{U}$ an update. Then define the ABox $\mathcal{A}^{\mathcal{U}}$ by setting

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

Lemma 20. *Let $\mathcal{A}$ be an ABox and $\mathcal{U}$ an update. For every interpretation $\mathcal{I}$ with $\mathcal{I} \models \mathcal{U}^{\neg}$, 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 $\mathcal{I}^{\mathcal{U}}$ and $\mathcal{A}^{\mathcal{U}}$, $\mathcal{I}^{\mathcal{U}} \models \varphi$. If $\varphi = E^{\mathcal{U}}(a)$ for $E(a) \in \mathcal{A}$, it follows from Lemma 19 that $\mathcal{I}^{\mathcal{U}} \models E^{\mathcal{U}}(a)$.

"$\Leftarrow$" Let $\mathcal{I}^{\mathcal{U}} \models \mathcal{A}^{\mathcal{U}}$. We show that $\mathcal{I} \models \mathcal{A}$. Let $\varphi \in \mathcal{A}$. If $\varphi = r(a,b)$, there are two cases to consider:

1. $\neg r(a,b) \in \mathcal{U}$. Then $r(a,b) \in \mathcal{U}^{\neg}$, and since $\mathcal{I} \models \mathcal{U}^{\neg}$, we obtain that $\mathcal{I} \models r(a,b)$.

2. $\neg r(a,b) \notin \mathcal{U}$. Then $r(a,b) \in \mathcal{A}^{\mathcal{U}}$, and thus $\mathcal{I}^{\mathcal{U}} \models r(a,b)$. By definition of $\mathcal{I}^{\mathcal{U}}$ we obtain $\mathcal{I} \models r(a,b)$.

The case $\varphi = \neg r(a,b)$ is analogous to the previous one, and the case $\varphi = E(a)$ follows from Lemma 19. ❏

We are in the position now to lift updates from the level of interpretations to the level of ABoxes. Let $\mathcal{A}$ be an ABox and $\mathcal{U}$ an update. The set of *literals* over $\mathcal{U}$ is defined as $L_{\mathcal{U}} := \{\psi, \neg\psi \mid \psi \in \mathcal{U}\}$. A simple ABox $\mathcal{D}$ is called a *diagram for* $\mathcal{U}$ if it is a maximal consistent subset of $L_{\mathcal{U}}$. Intuitively, a diagram gives a complete description of the part of a model of $\mathcal{A}$ that is "relevant" for the update $\mathcal{U}$. Let $\mathfrak{D}$ be the set of all diagrams for $\mathcal{U}$ and let $\mathcal{D} \in \mathfrak{D}$. Then define the update $\mathcal{D}_{\mathcal{U}}$ as

$$\mathcal{D}_{\mathcal{U}} := \{\psi \mid \neg\psi \in \mathcal{D} \text{ and } \psi \in \mathcal{U}\}.$$

Considering $\mathcal{D}_{\mathcal{U}}$ means taking a subset of $\mathcal{U}$ as described at the beginning of this section: we retain only those parts of $\mathcal{U}$ that are violated by interpretations whose relevant part is described by $\mathcal{D}$. We now define the updated ABox $\mathcal{A}'$ as

$$\mathcal{A}' := \bigvee_{\mathcal{D} \in \mathfrak{D}} \bigwedge \mathcal{A}^{\mathcal{D}_{\mathcal{U}}} \cup \mathcal{D}_{\mathcal{U}} \cup \mathcal{D}^{\mathcal{D}_{\mathcal{U}}}. \tag{1}$$

Intuitively, the component $\mathcal{A}^{\mathcal{D}_{\mathcal{U}}}$ is the update of the original ABox, $\mathcal{D}_{\mathcal{U}}$ asserts that the changes effected by the update hold, and $\mathcal{D}^{\mathcal{D}_{\mathcal{U}}}$ denotes the result of changing the diagram $\mathcal{D}$ under consideration as described by $\mathcal{U}$. The Boolean ABox operators are used only as an abbreviation for the "@" constructor. This can be safely done since the translation from Boolean ABoxes to non-Boolean ones described in the proof of Lemma 16 is linear. To achieve a less redundant ABox, it is possible to drop from $\mathcal{A}'$ those disjuncts for which the diagram $\mathcal{D}$ is not consistent w.r.t. $\mathcal{A}$. This is, however, not strictly necessary since the ABox $\mathcal{D}^{\mathcal{D}_{\mathcal{U}}}$ ensures that these disjuncts are inconsistent.

Lemma 21. $\mathcal{A} * \mathcal{U} \equiv \mathcal{A}'$.

Proof. We have to prove that $\mathcal{A}'$ satisfies Points (U1) and (U2) from Definition 2.

(U1) Let $\mathcal{I}$ and $\mathcal{I}'$ be two interpretations such that $\mathcal{I} \models \mathcal{A}$ and $\mathcal{I} \Longrightarrow_{\mathcal{U}} \mathcal{I}'$. We have to show that $\mathcal{I} \models \mathcal{A}'$. We define $\mathcal{D} \in \mathfrak{D}$ as $\mathcal{D} = \{l \in L_{\mathcal{U}} \mid \mathcal{I} \models l\}$. Then,

$$\chi_{\mathcal{D}} = \bigwedge \mathcal{D}^{\mathcal{D}_{\mathcal{U}}} \cup \mathcal{A}^{\mathcal{D}_{\mathcal{U}}} \cup \mathcal{D}_{\mathcal{U}}$$

is a disjunct in $\mathcal{A}'$ and it suffices to show that $\mathcal{I}' \models \chi_{\mathcal{D}}$. Since $\mathcal{I} \models \mathcal{D}$, by the definition of $\mathcal{D}_{\mathcal{U}}$ and $\Longrightarrow_{\mathcal{U}}$ it easily follows that $\mathcal{I}' = \mathcal{I}^{\mathcal{D}_{\mathcal{U}}}$. Thus, $\mathcal{I}' \models \mathcal{D}_{\mathcal{U}}$. Since $\mathcal{I} \models \mathcal{A} \cup \mathcal{D}$, by Lemma 20 we obtain that

$$\mathcal{I}' \models \mathcal{D}^{\mathcal{D}_{\mathcal{U}}} \cup \mathcal{A}^{\mathcal{D}_{\mathcal{U}}} \cup \mathcal{D}_{\mathcal{U}}.$$

(U2) 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 $\mathcal{I} \Longrightarrow_{\mathcal{U}} \mathcal{I}'$. Since $\mathcal{I}' \models \mathcal{A}'$ and $\mathcal{A}' = \bigvee_{\mathcal{D} \in \mathfrak{D}} \bigwedge \mathcal{D}^{\mathcal{D}_{\mathcal{U}}} \cup \mathcal{A}^{\mathcal{D}_{\mathcal{U}}} \cup \mathcal{D}_{\mathcal{U}}$, there exists a $\mathcal{D} \in \mathfrak{D}$ such that $\mathcal{I}' \models \mathcal{D}^{\mathcal{D}_{\mathcal{U}}} \cup \mathcal{A}^{\mathcal{D}_{\mathcal{U}}} \cup \mathcal{D}_{\mathcal{U}}$.

$$
\begin{aligned}
(\exists r.C)^{\mathcal{B}} &= (\bigsqcap_{a \in \mathsf{Obj}(\mathcal{B})} \neg\{a\} \sqcap \exists r.C^{\mathcal{B}}) \sqcup \exists r.(\bigsqcap_{a \in \mathsf{Obj}(\mathcal{B})} \neg\{a\} \sqcap C^{\mathcal{B}}) \\
&\sqcup \bigsqcup_{a,b \in \mathsf{Obj}(\mathcal{B}), r(a,b) \notin \mathcal{B}} (\{a\} \sqcap \exists r.(\{b\} \sqcap C^{\mathcal{B}})) \sqcup \bigsqcup_{\neg r(a,b) \in \mathcal{B}} (\{a\} \sqcap @_b C^{\mathcal{B}}) \\
(\forall r.C)^{\mathcal{B}} &= (\bigsqcap_{a \in \mathsf{Obj}(\mathcal{B})} \neg\{a\} \rightarrow \forall r.C^{\mathcal{B}}) \sqcap \forall r.(\bigsqcap_{a \in \mathsf{Obj}(\mathcal{B})} \neg\{a\} \rightarrow C^{\mathcal{B}}) \\
&\sqcap \bigsqcap_{a,b \in \mathsf{Obj}(\mathcal{B}), r(a,b) \notin \mathcal{B}} (\{a\} \rightarrow \forall r.(\{b\} \rightarrow C^{\mathcal{B}})) \sqcap \bigsqcap_{\neg r(a,b) \in \mathcal{B}} (\{a\} \rightarrow @_b C^{\mathcal{B}})
\end{aligned}
$$

Figure 7: Constructing $C^{\mathcal{B}}$ for existential and universal restrictions

Let $\mathcal{I} = (\mathcal{I}')^{\mathcal{D}_{\mathcal{U}}^{-}}$. Then we have that $\mathcal{I}' = \mathcal{I}^{\mathcal{D}_{\mathcal{U}}}$. Thus, by Lemma 20 and since $\mathcal{I}' \models \mathcal{A}^{\mathcal{D}_{\mathcal{U}}}$, we obtain that $\mathcal{I} \models \mathcal{A}$. Similarly, we obtain that $\mathcal{I} \models \mathcal{D}$.
It remains to show that $\mathcal{I} \Longrightarrow_{\mathcal{U}} \mathcal{I}'$. Let A be a concept name. Since $\mathcal{I}' = \mathcal{I}^{\mathcal{D}_{\mathcal{U}}}$, we have that

$$A^{\mathcal{I}'} = A^{\mathcal{I}} \cup \{a^{\mathcal{I}} \mid A(a) \in \mathcal{D}_{\mathcal{U}}\} \setminus \{a^{\mathcal{I}} \mid \neg A(a) \in \mathcal{D}_{\mathcal{U}}\}.$$

Moreover, by the definition of $\mathcal{D}_{\mathcal{U}}$ and since $\mathcal{I} \models \mathcal{D}$, we obtain that

$$\{a^{\mathcal{I}} \mid A(a) \in \mathcal{D}_{\mathcal{U}}\} = \{a^{\mathcal{I}} \mid A(a) \in \mathcal{U}\} \setminus A^{\mathcal{I}}$$

and

$$\{a^{\mathcal{I}} \mid \neg A(a) \in \mathcal{D}_{\mathcal{U}}\} = \{a^{\mathcal{I}} \mid \neg A(a) \in \mathcal{U}\} \cap A^{\mathcal{I}}.$$

Having $A^{\mathcal{I}^{\mathcal{U}}} = A^{\mathcal{I}} \cup \{a^{\mathcal{I}} \mid A(a) \in \mathcal{U}\} \setminus \{a^{\mathcal{I}} \mid \neg A(a) \in \mathcal{U}\}$, and $\mathcal{U}$ being consistent, we obtain that $A^{\mathcal{I}'} = A^{\mathcal{I}^{\mathcal{U}}}$. Similarly, we obtain $r^{\mathcal{I}'} = r^{\mathcal{I}^{\mathcal{U}}}$ for each role name r. Thus, $\mathcal{I}' = \mathcal{I}^{\mathcal{U}}$ and $\mathcal{I} \Longrightarrow_{\mathcal{U}} \mathcal{I}'$.

❑

It is easy to adapt the construction of updated ABoxes to the DLs $\mathcal{ALCO}^{@}$, $\mathcal{ALCIO}^{@}$, $\mathcal{ALCQO}^{@}$. 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 7. The lemmas proved above for $\mathcal{ALCQIO}^{@}$ are then easily easily adapted.

Theorem 22. *All of the following DLs have ABox updates: $\mathcal{ALCO}^{@}$, $\mathcal{ALCIO}^{@}$, $\mathcal{ALCQO}^{@}$, and $\mathcal{ALCQIO}^{@}$.*

A close inspection of the updated ABox $\mathcal{A}'$ computed above reveals that, first, the size the concepts $C^{\mathcal{U}}$ is exponential in the size of $\mathcal{A}$ and the update $\mathcal{U}$; and second, the number of disjuncts in $\mathcal{A}'$ is exponential in $\mathcal{U}$, but polynomial in $\mathcal{A}$. This is independent of whether the numbers inside number restrictions are coded in unary or in binary. Therefore, we obtain the following.

Theorem 23. *Let $\mathcal{L} \in \{\mathcal{ALCO}^{@}, \mathcal{ALCIO}^{@}, \mathcal{ALCQO}^{@}, \mathcal{ALCQIO}^{@}\}$. Then there exist polynomials p_1, p_2, and q such that, for every $\mathcal{L}$-ABox $\mathcal{A}$ and every update $\mathcal{U}$, there exists an $\mathcal{L}$-ABox $\mathcal{A}'$ such that the following hold:*

- $\mathcal{A} * \mathcal{U} \equiv \mathcal{A}'$;
- $|\mathcal{A}'| \leq 2^{p_1(|\mathcal{A}|)} \cdot 2^{p_2(|\mathcal{U}|)}$;
- *$\mathcal{A}'$ can be computed in time $q(|\mathcal{A}'|)$.*

By the arguments given in Section 2.3, an exponential blowup cannot be entirely avoided unless PTime = NC. However, we should pay attention to whether the blowup occurs in the size of the original ABox $\mathcal{A}$ or in the size of the update $\mathcal{U}$. As the update will usually be rather small compared to the original ABox, an exponential blowup in the size of $\mathcal{U}$ is much more acceptable than an exponential blowup in the size of $\mathcal{A}$. The algorithm given in this section produces an exponential in *both* $\mathcal{A}$ and $\mathcal{U}$. In the case of propositional logic, Winslett [26] gives an algorithm that blows up exponentially only in the size of $\mathcal{U}$, but not in the size of (the equivalent of) $\mathcal{A}$. We believe that, for the languages mentioned in Theorem 23, the exponential blowup in $|\mathcal{A}|$ can *not* be avoided. For example, consider the family of ABoxes $(\mathcal{A}_i)_{i \in \mathbb{N}}$ defined as follows:

$$\mathcal{A}_i := \{a : \exists r.(A_1 \sqcap \exists r.(A_2 \sqcap \cdots \exists r.(A_i \sqcap \exists r.A_{i+1}) \cdots))\}.$$

Clearly, for $\mathcal{U} = \{\neg r(b, b')\}$ the size of the ABox $\mathcal{A}'_i \equiv \mathcal{A}_i * \mathcal{U}$ when computing it using the algorithm above is exponential in the size of $\mathcal{A}_i$. We suspect that there exists no polynomial p such that, for all $i \geq 0$, there is an ABox $\mathcal{A}'_i$ such that $\mathcal{A}_i * \{\neg r(b, b')\} = \mathcal{A}'_i$ and $|\mathcal{A}'_i| \leq p(|\mathcal{A}_i|)$. While we leave a proof as an open problem, in Section 5 we exhibit several ways around an exponential blowup in the size $\mathcal{A}$. Before that, however, we take a look at several variations of our result.

Iterated Updates

There are applications in which the domain of interest evolves continuously. In such an environment, it is necessary to update an ABox over and over again. Then, it is clearly important that the exponential blowups of the individual updates do not add up. The following theorem shows that this is indeed not the case.

Theorem 24. *There exist polynomials p_1, p_2 such that the following holds: for all ABoxes $\mathcal{A}_0, \ldots, \mathcal{A}_n$ and updates $\mathcal{U}_1, \ldots, \mathcal{U}_n$, if $\mathcal{A}_i$ is the ABox computed by our algorithm when $\mathcal{A}_{i-1}$ is updated with $\mathcal{U}_i$, for $0 < i \leq n$, then*

$$|\mathcal{A}_n| \leq 2^{p_1(|\mathcal{A}_0|)} \cdot 2^{p_2(|\mathcal{U}_1|+\cdots+|\mathcal{U}_n|)}.$$

Proof. For a concept C, denote by n_C the maximal number occurring in a qualified number restriction in C. Furthermore, denote by $d(C)$ the maximal nesting depth of qualified number restrictions in C. We find polynomials q_1 and q_2 such that, for every concept C and every update $\mathcal{U}$,

$$|C^{\mathcal{U}}| \leq |C| \times (q_1(n_C) \times 2^{q_2(|\mathcal{U}|)})^{d(C)}.$$

The crucial observation now is that, for every concept C and update $\mathcal{U}$, $n_C = n_{C_{\mathcal{U}}}$ and $d(C) = d(C^{\mathcal{U}})$: neither the maximal number nor the maximal nesting depth of qualified numbers restrictions increases when forming $C^{\mathcal{U}}$. It follows that there exist polynomials q_1 and q_2 such that for every concept C and sequence of updates $\mathcal{U}_1, \ldots, \mathcal{U}_i$,

$$|(C^{\mathcal{U}_1})^{\mathcal{U}_2 \cdots \mathcal{U}_i}| \leq 2^{q_1(|C|) \times q_2(|\mathcal{U}_1|+\cdots+|\mathcal{U}_i|)}.$$

A close inspection of the construction of $\mathcal{A}_i$ from $\mathcal{A}_{i-1}$ using the concepts $(C^{\mathcal{U}_1})^{\mathcal{U}_2 \cdots \mathcal{U}_{i-1}}$ shows that there exists an additional polynomial q_2' such that, for all i,

$$|\mathcal{A}_i| \leq 2^{q_2'(|\mathcal{U}_1|+\cdots+|\mathcal{U}_i|)} \times \sum_{a:C \in \mathcal{A}_0} 2^{q_1(|C|) \times q_2(|\mathcal{U}_1|+\cdots+|\mathcal{U}_i|)}$$

The upper bound claimed in the theorem follows immediately. ❏

Conditional Updates

For the sake of simplicity, we have defined ABox updates to be unconditional: the assertions in the update $\mathcal{U}$ are true after the update, no matter to which interpretation $\mathcal{U}$ is applied. In some applications such as reasoning about actions with DLs [3], it is more useful to have *conditional updates*, where the initial interpretation determines the changes that are triggered.

A *conditional update* $\mathcal{U}^*$ is a finite set of expressions φ/ψ, where φ is an ABox assertion (possibly involving non-atomic concepts) and ψ is an assertion of the form

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

with A a concept name. Intuitively, an expression φ/ψ means that if φ holds in the initial interpretation, then ψ holds after the update. As in the case of uncondition 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.

The definition of interpretation updates can straightforwardly be adapted to the case of conditional updates: an interpretation $\mathcal{I}'$ is the *result of updating an interpretation $\mathcal{I}$ with a conditional update $\mathcal{U}^*$* if the following hold:

- for all concept names A,

  $$A^{\mathcal{I}'} \;=\; A^{\mathcal{I}} \cup \{a^{\mathcal{I}} \mid \varphi/A(a) \in \mathcal{U}^* \wedge \mathcal{I} \models \varphi\} \setminus \{a^{\mathcal{I}} \mid \varphi/\neg A(a) \in \mathcal{U}^* \wedge \mathcal{I} \models \varphi\}$$

- for all role names r,

  $$\begin{aligned} r^{\mathcal{I}'} \;=\; & (r^{\mathcal{I}} \cup \{(a^{\mathcal{I}}, b^{\mathcal{I}}) \mid \varphi/r(a,b) \in \mathcal{U}^* \wedge \mathcal{I} \models \varphi\}) \\ & \qquad \setminus \{(a^{\mathcal{I}}, b^{\mathcal{I}}) \mid \varphi/\neg r(a,b) \in \mathcal{U}^* \wedge \mathcal{I} \models \varphi\}. \end{aligned}$$

Conditions (U1) and (U2) are as in the case of unconditional updates. Clearly, conditional updates generalize unconditional once since assertions ψ of unconditional updates can be expressed as $\top(a)/\psi$, with a an arbitrary individual name.

We now show how to adapt our construction of updated ABoxes to conditional updates. For $\mathcal{U}^*$ a conditional update, we use $\mathsf{rhs}(\mathcal{U}^*)$ to denote $\{\psi \mid \varphi/\psi \in \mathcal{U}^*\}$, and $\mathsf{lhs}(\mathcal{U}^*)$ for $\{\varphi \mid \varphi/\psi \in \mathcal{U}^*\}$. In the original algorithm, the updated ABox $\mathcal{A}'$ is assembled by taking one disjunct for every diagram for $\mathcal{U}$. The intuition is that such a diagram $\mathcal{D}$ gives complete information about the assertions in $\mathcal{U}^*$ that actually cause a change when $\mathcal{U}$ is applied to models whose relevant part is described by $\mathcal{D}$ (assertions in $\mathcal{U}$ do not cause a change if they were already satisfied before the update). We generalize this idea to conditional updates by taking one disjunct for each pair $(\mathcal{D}, \mathcal{U}')$, where $\mathcal{D}$ is a diagram for $\mathsf{rhs}(\mathcal{U}^*)$, and $\mathcal{U}'$ is a subset of $\mathcal{U}^*$. Intuitively, $\mathcal{U}'$ determines the set of assertions from $\mathcal{U}$ whose preconditions are satisfied in the initial model, and $\mathcal{D}$ determines the post-conditions that can actually cause a change.

Let $\mathfrak{D}^*$ be the set of all diagrams for $\mathsf{rhs}(\mathcal{U}^*)$. Let $\mathcal{D} \in \mathfrak{D}^*$ and $\mathcal{U}' \subseteq \mathcal{U}^*$. As before, we define the simple ABox $\mathcal{D}_{\mathcal{U}'}$ as

$$\mathcal{D}_{\mathcal{U}'} := \{\psi \mid \neg\psi \in \mathcal{D} \text{ and } \varphi/\psi \in \mathcal{U}'\}.$$

Then we can assemble the updated ABox $\mathcal{A}^*$ as follows:

$$\begin{aligned} \mathcal{A}^* \;=\; \bigvee_{\mathcal{D} \in \mathfrak{D}^*} \bigvee_{\mathcal{U}' \subseteq \mathcal{U}^*} \bigwedge & \{\varphi \mid \varphi/\psi \in \mathcal{U}'\}^{\mathcal{D}_{\mathcal{U}'}} \cup \{\neg\varphi \mid \varphi/\psi \in \mathcal{U} \setminus \mathcal{U}'\}^{\mathcal{D}_{\mathcal{U}'}} \\ & \cup \mathcal{A}^{\mathcal{D}_{\mathcal{U}'}} \cup \mathcal{D}_{\mathcal{U}'} \cup \mathcal{D}^{\mathcal{D}_{\mathcal{U}'}} \end{aligned}$$

By slightly modifying the proof of Lemma 21, it is not difficult to show that $\mathcal{A}^*$ is indeed the result of updating $\mathcal{A}$ with the conditional update $\mathcal{U}^*$. The notion of a description logic $\mathcal{L}$ *having conditional ABox updates* is defined in the obvious way.

Theorem 25. *All of the following DLs have conditional ABox updates: $\mathcal{ALCO}^{@}$, $\mathcal{ALCIO}^{@}$, $\mathcal{ALCQO}^{@}$, and $\mathcal{ALCQIO}^{@}$.*

Concerning the size and computability of updated ABoxes, we obtain a result analogous to Theorem 23.

Boolean ABox Updates

In Section 3.3, Boolean ABoxes were introduced as a generalization of standard ABoxes, and a close connection between Boolean ABoxes and the @ constructor was established. In fact, using the arguments of Lemma 16, it is easy to see that the expressive power of Boolean $\mathcal{L}$-ABoxes is identical to the expressive power of non-Boolean $\mathcal{L}^{@}$-ABoxes, for $\mathcal{L}$ any of $\mathcal{ALCO}$, $\mathcal{ALCIO}$, $\mathcal{ALCQO}$, and $\mathcal{ALCQIO}$. Hence, Theorems 22 and 23 can also be understood in terms of Boolean ABoxes.

We say that a description logic $\mathcal{L}$ has *Boolean ABox updates* if, for every Boolean $\mathcal{L}$-ABox $\mathcal{A}$ and update $\mathcal{U}$, there exists a Boolean $\mathcal{L}$-ABox $\mathcal{A}'$ satisfying Conditions (U1) and (U2) of Definition 2. Due to the generalization of Lemma 16 to the relevant languages, the construction presented in this section can also be used to compute Boolean ABox updates: first convert the Boolean $\mathcal{L}$-ABox into a non-Boolean $\mathcal{L}^{@}$-ABox, apply the described construction, and then convert the resulting non-Boolean $\mathcal{L}^{@}$-ABox back into a Boolean $\mathcal{L}$-ABox.

Theorem 26. *All of the following DLs have Boolean ABox updates: $\mathcal{ALCO}$, $\mathcal{ALCIO}$, $\mathcal{ALCQO}$, $\mathcal{ALCQIO}$, and their extensions with the @ constructor.*

What is the size of updated Boolean ABoxes computed by the above approach? The main observation is that, while the translation of Boolean $\mathcal{L}$-ABoxes into non-Boolean $\mathcal{L}^{@}$-ABoxes is polynomial, the reverse translation induces an exponential blowup. More precisely, this blowup is exponential in the nesting depth of the @ constructor. Since our translation introduces nestings of the @ constructor whose depth is linear in the size of the original ABox, our algorithm now produces a double exponential blowup in the size of the original ABox.

Theorem 27. *Let $\mathcal{L} \in \{\mathcal{ALCO}, \mathcal{ALCIO}, \mathcal{ALCQO}, \mathcal{ALCQIO}\}$. Then there exist polynomials p_1, p_2, and q such that, for every $\mathcal{L}$-ABox $\mathcal{A}$ and every update $\mathcal{U}$, there exists an $\mathcal{L}$-ABox $\mathcal{A}'$ such that the following hold:*

- $\mathcal{A} * \mathcal{U} = \mathcal{A}'$;
- $|\mathcal{A}'| \leq 2^{2^{p_1(|\mathcal{A}|)}} \cdot 2^{p_2(|\mathcal{U}|)}$;
- *$\mathcal{A}'$ can be computed in time $q(|\mathcal{A}'|)$.*

Note that, for the languages $\mathcal{L}^{@}$, with $\mathcal{L}$ as in Theorem 27, we have Boolean updates whose size is as described in Theorem 23, i.e., only single exponential in the original ABox: the final conversion step of non-Boolean $\mathcal{L}^{@}$-ABoxes into Boolean $\mathcal{L}$-ABoxes can simply be omitted. We currently don't know whether the upper bounds given in Theorem 27 can be improved.

5 Small(er) Updated ABoxes

The size of the updated ABoxes computed in the previous sections is exponential in the size of the original ABox. When replacing the @-operator with Boolean ABoxes, it is even 2-exponential in the size of the original ABox. In this section, we explore two different ways to extend $\mathcal{ALCQIO}^{@}$ and its fragments such that it becomes possible to compute updated ABoxes that are only polynomial in the size of the original ABox.

A first, rather restrictive solution is to admit only concept assertions in updates. Then, in all DLs captured by Theorem 22, computing the concepts $C^{\mathcal{U}}$ becomes a lot simpler: just replace every concept name A in C with

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

If modified in this way, our original construction clearly yields updated ABoxes that are only polynomial in the size of the original ABox (but still exponential in $\mathcal{U}$). The bound is independent of the coding of numbers and also applies to iterated updates.

5.1 $\mathcal{ALCQIO}^{@}$ Updates with TBoxes

We show how to produce "small"a updated ABoxes by allowing the introduction of additional concept names via an acyclic TBox. In the propositional case, this corresponds to admitting additional variables for defining abbreviations. In the terminology of Cadoli et al. [7], we thus move from logical equivalence to query equivalence. It will turn out that, in this way, we obtain updates that are only polynomial in the size of the original ABox. It is interesting to note that, in the propositional case, the admission of additional variables does not lead to more succinct updated formulas: in the worst case, they are still exponential in the size of the update [7].

A *concept definition* is of the form $A \equiv C$, where A is a concept name and C is a concept. A *TBox* $\mathcal{T}$ is a finite set of concept definitions with unique left-hand sides. A TBox $\mathcal{T}$ is called *acyclic* if no concept is defined (directly or indirectly) in terms of itself. We call a concept name A *defined* in a TBox $\mathcal{T}$ and write $A \in \mathsf{def}(\mathcal{T})$ if A occurs on the left-hand side of a concept definition in $\mathcal{T}$.

Otherwise, we call A *primitive* and write $A \in \mathsf{prim}(\mathcal{T})$. A *knowledge base (KB)* is a pair $(\mathcal{T}, \mathcal{A})$ consisting of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$. An interpretation $\mathcal{I}$ *satisfies* a concept definition $A \equiv C$ if $A^{\mathcal{I}} = C^{\mathcal{I}}$. $\mathcal{I}$ is a *model* of a TBox $\mathcal{T}$, written $\mathcal{I} \models \mathcal{T}$, if $\mathcal{I}$ satisfies all concept definitions in $\mathcal{T}$; $\mathcal{I}$ is a *model* of a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, written $\mathcal{I} \models \mathcal{K}$, if $\mathcal{I}$ is a model of $\mathcal{T}$ and $\mathcal{A}$.

Let $\mathcal{T}$ be a TBox. An *update $\mathcal{U}$ for $\mathcal{T}$* is a simple and consistent ABox that does not use concept names from $\mathsf{def}(\mathcal{T})$. We do not allow defined concept names in updates because this is obviously equivalent to admitting updates with complex concepts and thus violates our policy of considering only updates on an atomic level.

Definition 28 (Interpretation update). Let $\mathcal{T}$ be an acyclic TBox, $\mathcal{U}$ an update for $\mathcal{T}$, and $\mathcal{I}, \mathcal{I}'$ models of $\mathcal{T}$ such that $\Delta^{\mathcal{I}} = \Delta^{\mathcal{I}'}$ and $\mathcal{I}$ and $\mathcal{I}'$ agree on the interpretation of individual names. Then $\mathcal{I}'$ is the *result of updating $\mathcal{I}$ with $\mathcal{U}$ relative to $\mathcal{T}$, written $\mathcal{I} \Longrightarrow_{\mathcal{U}}^{\mathcal{T}} \mathcal{I}'$*, if the following hold:

- for all concept names $A \in \mathsf{prim}(\mathcal{T})$:

$$A^{\mathcal{I}'} = (A^{\mathcal{I}} \cup \{a^{\mathcal{I}} \mid A(a) \in \mathcal{U}\}) \setminus \{a^{\mathcal{I}} \mid \neg A(a) \in \mathcal{U}\};$$

- for all role names r,

$$r^{\mathcal{I}'} = (r^{\mathcal{I}} \cup \{(a^{\mathcal{I}}, b^{\mathcal{I}}) \mid r(a,b) \in \mathcal{U}\}) \setminus \{(a^{\mathcal{I}}, b^{\mathcal{I}}) \mid \neg r(a,b) \in \mathcal{U}\}.$$

△

The difference between Definitions 1 and 28 is that the latter talks only about concept names that are primitive w.r.t. $\mathcal{T}$. Observe that the relation $\Longrightarrow_{\mathcal{U}}^{\mathcal{T}}$ is still deterministic: in models of acyclic TBoxes, the interpretation of primitive concept names and role names determines the interpretation of defined concept names in a unique way.

Definition 29 (Knowledge Base Update). Let $\mathcal{K}_1$ and $\mathcal{K}_2$ be knowledge bases, $\mathcal{K}_i = (\mathcal{T}_i, \mathcal{A}_i)$, such that $\mathsf{prim}(\mathcal{T}_1) = \mathsf{prim}(\mathcal{T}_2)$ and $\mathcal{T}_1 \subseteq \mathcal{T}_2$, and let $\mathcal{U}$ be an update for $\mathcal{T}_1$. Then *$\mathcal{K}_2$ is a result of updating $\mathcal{K}_1$ with $\mathcal{U}$* if the following conditions hold:

(U1′) $\forall \mathcal{I}, \mathcal{I}' : \big(\, (\, \mathcal{I} \models \mathcal{K}_1 \;\wedge\; \mathcal{I} \Longrightarrow_{\mathcal{U}}^{\mathcal{T}_1} \mathcal{I}' \wedge \mathcal{I}' \models \mathcal{T}_2 \,) \rightarrow \mathcal{I}' \models \mathcal{A}_2) \, \big);$

(U2′) $\forall \mathcal{I}' : \big(\, \mathcal{I}' \models \mathcal{K}_2 \;\rightarrow\; \exists \mathcal{I} \; (\mathcal{I} \models \mathcal{K}_1 \;\wedge\; \mathcal{I} \Longrightarrow_{\mathcal{U}}^{\mathcal{T}_1} \mathcal{I}') \, \big).$

In this case, we write $\mathcal{K}_1 * \mathcal{U} \equiv_{\mathsf{P}} \mathcal{K}_2$. △

In contrast to ABox updates, the result $\mathcal{K}_2$ of updating a knowledge base is not unique up to logical equivalence. This is due to the fact that we have more than one choice for introducing new concept definitions in $\mathcal{T}_2$. However, we have the following, weaker form of equivalence. A *primitive interpretation for a TBox* $\mathcal{T}$ is an interpretation that interprets only the primitive concept names in $\mathcal{T}$ and the role names, but not the defined concept names. A primitive interpretation is a *primitive model* of a knowledge base $\mathcal{K}$ if it can be extended to a model of $\mathcal{K}$ by additionally interpreting the defined concept names. Then, it is an easy consequence of Definition 29 that $\mathcal{K}_1 * \mathcal{U} \equiv_{\mathsf{P}} \mathcal{K}_2$ and $\mathcal{K}_1 * \mathcal{U} \equiv_{\mathsf{P}} \mathcal{K}_2'$ implies that $\mathcal{K}_2$ and $\mathcal{K}_2'$ have the same primitive models.

We now use the notion of *unfolding* to establish a relationship between updates of ABoxes and updates of knowledge bases. Let $\mathcal{T}$ be an acyclic TBox, and C a concept. The concept $C^{\mathcal{T}}$ obtained from C by exhaustively replacing defined concept names in C with their definitions from $\mathcal{T}$ is called the *unfolding of C w.r.t. $\mathcal{T}$*. Clearly, all concept names occurring in $C^{\mathcal{T}}$ are primitive w.r.t. $\mathcal{T}$. If $\mathcal{A}$ is an ABox, then the *unfolding of $\mathcal{A}$ w.r.t.* $\mathcal{T}$ is the ABox

$$\mathcal{A}^{\mathcal{T}} := (\mathcal{A} \setminus \{C(a) \mid C(a) \in \mathcal{A}\}) \cup \{C^{\mathcal{T}}(a) \mid C(a) \in \mathcal{A}\}.$$

I.e., we keep role assertions and replace concept assertions by their unfolded variants.

The following lemma shows that updated ABoxes for acyclic TBoxes encode updated ABoxes without acyclic TBoxes. In the following, we use $\mathsf{prim}_{\mathcal{T}}(\mathcal{I})$ to denote the (unique) primitive interpretation w.r.t. $\mathcal{T}$ that can be extended to the full interpretation $\mathcal{I}$.

Lemma 30. *Let $\mathcal{K}_1$ and $\mathcal{K}_2$ be knowledge bases, $\mathcal{K}_i = (\mathcal{T}_i, \mathcal{A}_i)$, such that $\mathcal{T}_1 \subseteq \mathcal{T}_2$ and $\mathsf{prim}(\mathcal{T}_1) = \mathsf{prim}(\mathcal{T}_2)$, and let $\mathcal{U}$ be an update for $\mathcal{T}_1$. Then*

$$\mathcal{K}_1 * \mathcal{U} \equiv_{\mathsf{P}} \mathcal{K}_2 \quad \textit{iff} \quad \mathcal{A}_1^{\mathcal{T}_1} * \mathcal{U} \equiv \mathcal{A}_2^{\mathcal{T}_2}.$$

Proof. "$\Leftarrow$" Let $\mathcal{A}_1^{\mathcal{T}_1} * \mathcal{U} \equiv \mathcal{A}_2^{\mathcal{T}_2}$. In order to prove that $\mathcal{K}_1 * \mathcal{U} \equiv_{\mathsf{P}} \mathcal{K}_2$, we need to show that (U1$'$) and (U2$'$) from Definition 29 are satisifed:

(U1$'$) Let $\mathcal{I}$ and $\mathcal{I}'$ be interpretations such that $\mathcal{I} \models \mathcal{K}_1$, $\mathcal{I} \Longrightarrow_{\mathcal{U}}^{\mathcal{T}_1} \mathcal{I}'$, and $\mathcal{I}' \models \mathcal{T}_2$. We need to show that then $\mathcal{I}' \models \mathcal{A}_2$. Since $\mathcal{I} \models \mathcal{A}_1, \mathcal{T}_1$, we have that $\mathsf{prim}_{\mathcal{T}_1}(\mathcal{I}) \models \mathcal{A}_1^{\mathcal{T}_1}$. Moreover, since $\mathsf{prim}_{\mathcal{T}_1}(\mathcal{I}) \Longrightarrow_{\mathcal{U}} \mathsf{prim}_{\mathcal{T}_1}(\mathcal{I}')$ and $\mathcal{A}_1^{\mathcal{T}_1} * \mathcal{U} \equiv \mathcal{A}_2^{\mathcal{T}_2}$, by (U1) of Definition 2 we obtain that $\mathsf{prim}_{\mathcal{T}_1}(\mathcal{I}') \models \mathcal{A}_2^{\mathcal{T}_2}$. Thus, having $\mathsf{prim}(\mathcal{T}_1) = \mathsf{prim}(\mathcal{T}_2)$ and $\mathcal{I}' \models \mathcal{T}_2$, we obtain that $\mathcal{I}' \models \mathcal{A}_2$.

(U2$'$) Let $\mathcal{I}'$ be an interpretation such that $\mathcal{I}' \models \mathcal{K}_2$. We need to show that there is an $\mathcal{I}$ such that $\mathcal{I} \Longrightarrow_{\mathcal{U}}^{\mathcal{T}_1} \mathcal{I}'$ and $\mathcal{I} \models \mathcal{K}_1$. Since $\mathcal{I}' \models \mathcal{A}_2, \mathcal{T}_2$ and $\mathsf{prim}(\mathcal{T}_1) = \mathsf{prim}(\mathcal{T}_2)$, we have that $\mathsf{prim}_{\mathcal{T}_1}(\mathcal{I}') \models \mathcal{A}_2^{\mathcal{T}_2}$, and by (U2) of Definition 2, there exists an $\hat{\mathcal{I}}$ such that $\hat{\mathcal{I}} \Longrightarrow_{\mathcal{U}} \mathsf{prim}_{\mathcal{T}_1}(\mathcal{I}')$ and $\hat{\mathcal{I}} \models \mathcal{A}_1^{\mathcal{T}_1}$. Take an $\mathcal{I}$ such that $\mathsf{prim}_{\mathcal{T}_1}(\mathcal{I}) = \hat{\mathcal{I}}$ and $\mathcal{I} \models \mathcal{T}_1$. Then, by definition of unfolding we have that $\mathcal{I} \models \mathcal{A}_1$. Thus $\mathcal{I} \models \mathcal{K}_1$. Finally, since $\mathcal{T}_2 \supset \mathcal{T}_1$, it also holds that $\mathcal{I} \Longrightarrow_{\mathcal{U}}^{\mathcal{T}_1} \mathcal{I}'$.

"$\Rightarrow$" Let $\mathcal{K}_1 * \mathcal{U} \equiv_{\mathsf{P}} \mathcal{K}_2$. In order to prove that $\mathcal{A}_1^{\mathcal{T}_1} * \mathcal{U} \equiv \mathcal{A}_2^{\mathcal{T}_2}$, we need to show that (U1) and (U2) from Definition 2 are satisfied:

(U1) Let $\mathcal{I}$ and $\mathcal{I}'$ be interpretations such that $\mathcal{I} \models \mathcal{A}_1^{\mathcal{T}_1}$ and $\mathcal{I} \Longrightarrow_{\mathcal{U}} \mathcal{I}'$. Take an $\hat{\mathcal{I}}$ such that $\mathsf{prim}_{\mathcal{T}_1}(\hat{\mathcal{I}}) = \mathcal{I}$ and $\hat{\mathcal{I}} \models \mathcal{T}_1$. By definition of unfolding, we have that $\hat{\mathcal{I}} \models \mathcal{A}_1$. Now take an $\hat{\mathcal{I}}'$ such that $\mathsf{prim}_{\mathcal{T}_2}(\hat{\mathcal{I}}') = \mathcal{I}'$. and $\hat{\mathcal{I}}' \models \mathcal{T}_2$. Since $\mathsf{prim}(\mathcal{T}_1) = \mathsf{prim}(\mathcal{T}_2)$ and $\mathcal{T}_2 \supset \mathcal{T}_1$, we easily obtain that $\hat{\mathcal{I}} \Longrightarrow_{\mathcal{U}}^{\mathcal{T}_1} \hat{\mathcal{I}}'$, and by (U1$'$) of Definition 29 that $\hat{\mathcal{I}}' \models \mathcal{A}_2$. But then we have that $\mathcal{I}' \models \mathcal{A}_2^{\mathcal{T}_2}$.

(U2) Let $\mathcal{I}'$ be an interpretation such that $\mathcal{I}' \models \mathcal{A}_2^{\mathcal{T}_2}$. We need to show that there is an $\mathcal{I}$ such that $\mathcal{I} \Longrightarrow_{\mathcal{U}} \mathcal{I}'$ and $\mathcal{I} \models \mathcal{A}_1^{\mathcal{T}_1}$. Take an $\hat{\mathcal{I}}'$ such that $\mathsf{prim}_{\mathcal{T}_2}(\hat{\mathcal{I}}') = \mathcal{I}'$ and $\hat{\mathcal{I}}' \models \mathcal{T}_2$. By definition of unfolding we have that $\hat{\mathcal{I}}' \models \mathcal{A}_2$. Thus $\hat{\mathcal{I}}' \models \mathcal{K}_2$ and, by (U2$'$) of Definition 29, there is an interpretation $\hat{\mathcal{I}}$, such that $\hat{\mathcal{I}} \Longrightarrow_{\mathcal{U}}^{\mathcal{T}_1} \hat{\mathcal{I}}'$ and $\hat{\mathcal{I}} \models \mathcal{K}_1$. Take $\mathcal{I} = \mathsf{prim}_{\mathcal{T}_1}(\hat{\mathcal{I}})$. Then $\mathcal{I} \models \mathcal{A}_1^{\mathcal{T}_1}$. Finally, by definition of $\Longrightarrow_{\mathcal{U}}^{\mathcal{T}_1}$ and since $\mathsf{prim}(\mathcal{T}_2) = \mathsf{prim}(\mathcal{T}_1)$, we have that $\mathcal{I} \Longrightarrow_{\mathcal{U}} \mathcal{I}'$.

❑

We now show how to construct updated knowledge bases in $\mathcal{ALCQIO}^{@}$ and its fragments. Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a knowledge base, and let $\mathcal{U}$ be an update for $\mathcal{T}$. As in Section 4, we use $\mathfrak{D}$ to denote the set of all diagrams for $\mathcal{U}$ and set, for every $\mathcal{D} \in \mathfrak{D}$,

$$\mathcal{D}_{\mathcal{U}} := \{\psi \mid \psi \in \mathcal{U} \text{ and } \neg\psi \in \mathcal{D}\}.$$

Additionally, we use $\mathsf{sub}(\mathcal{K})$ to denote the set of all subconcepts of concepts occurring in $\mathcal{K}$. To construct the result of updating $\mathcal{K}$ with $\mathcal{U}$, we introduce a new concept name $A_C^{\mathcal{D}}$ for every diagram $\mathcal{D} \in \mathfrak{D}$ and every $C \in \mathsf{sub}(\mathcal{K})$. For a concept E, let $\mathsf{trans}(E, \mathcal{D})$ denote the concept on the right-hand side of the clause for $E^{\mathcal{D}_{\mathcal{U}}}$ in Figure 6 *without* inductively expanding the occurring subconcepts $C^{\mathcal{D}_{\mathcal{U}}}$, but with each such concept $C^{\mathcal{D}_{\mathcal{U}}}$ replaced with the concept name $A_C^{\mathcal{D}}$. For example, $\mathsf{trans}(C \sqcap D, \mathcal{D}) = A_C^{\mathcal{D}} \sqcap A_D^{\mathcal{D}}$ also if C and D are complex. For each diagram $\mathcal{D} \in \mathfrak{D}$, define a TBox

$$\mathcal{T}_{\mathsf{sub}}^{\mathcal{D}} := \{A_C^{\mathcal{D}} \equiv \mathsf{trans}(C, \mathcal{D}) \mid C \in \mathsf{sub}(\mathcal{K}) \setminus \mathsf{def}(\mathcal{T})\}.$$

Then, we define the TBox updated TBox as the union of the original TBox $\mathcal{T}$ and, for each diagram $\mathcal{D}$, the TBox $\mathcal{T}_{\mathsf{sub}}^{\mathcal{D}}$ and a version of $\mathcal{T}$ adapted to $\mathcal{D}$:

$$\mathcal{T}' := \mathcal{T} \cup \bigcup_{\mathcal{D} \in \mathfrak{D}} (\mathcal{T}_{\mathsf{sub}}^{\mathcal{D}} \cup \{A_A^{\mathcal{D}} \equiv A_C^{\mathcal{D}} \mid A \equiv C \in \mathcal{T}\}).$$

For every $\mathcal{D} \in \mathfrak{D}$, we define

$$\begin{aligned}
\mathcal{A}_{\mathcal{D}_{\mathcal{U}}} \;:=\; & \{A_C^{\mathcal{D}}(a) \mid C(a) \in \mathcal{A}\} \cup \\
& \{r(a,b) \mid r(a,b) \in \mathcal{A} \wedge \neg r(a,b) \notin \mathcal{D}_{\mathcal{U}}\} \cup \\
& \{\neg r(a,b) \mid \neg r(a,b) \in \mathcal{A} \wedge r(a,b) \notin \mathcal{D}_{\mathcal{U}}\}
\end{aligned}$$

Now we can define the updated ABox $\mathcal{A}'$ by setting

$$\mathcal{A}' = \bigvee_{\mathcal{D} \in \mathfrak{D}} \bigwedge \mathcal{A}_{\mathcal{D}_\mathcal{U}} \cup \mathcal{D}_\mathcal{U} \cup (\mathcal{D} \setminus \mathcal{D}_\mathcal{U}^\neg)$$

and finally assemble the updated knowledge base by setting $\mathcal{K}' := (\mathcal{T}', \mathcal{A}')$. Note that the concept definitions from $\mathcal{T}$ appear in $\mathcal{T}'$ without being referred to by $\mathcal{A}'$. Intuitively, this is hardly surprising: the definitions in $\mathcal{T}$ were used to describe the previous state of the world. Since this state has changed, the definitions in $\mathcal{T}$ are not appropriate any longer. We nevertheless keep $\mathcal{T}$ in the updated knowledge base since concept definitions are usually not only (technical) abbreviations, but rather reflect the terminology of the application domain. Therefore, they should not simply be discarded. One may even consider producing an updated knowledge base that reuses as many concept definitions from $\mathcal{T}$ as possible. This is outside the scope of the current paper.

Lemma 31. $\mathcal{K} * \mathcal{U} \equiv_\mathsf{P} \mathcal{K}'$

Proof. By the construction of $\mathcal{T}'$, it is obviously the case that $\mathcal{T}' \supseteq \mathcal{T}$ and $\mathsf{prim}(\mathcal{T}) = \mathsf{prim}(\mathcal{T}')$. Then, by Lemma 30 it suffices to show that $(\mathcal{A}')^{\mathcal{T}'}$ is the result of updating $\mathcal{A}^\mathcal{T}$ with $\mathcal{U}$. Let us use $(\mathcal{A}^\mathcal{T})'$ to refer to the update of $\mathcal{A}^\mathcal{T}$ with $\mathcal{U}$, as constructed in (1) in Section 4. Since the ABox update without TBoxes is unique up to equivalence, we just need to show that $(\mathcal{A}')^{\mathcal{T}'} \equiv (\mathcal{A}^\mathcal{T})'$. Since the updates, and thus also their diagrams contain no concept names from $\mathsf{def}(\mathcal{T})$, we have that

$$(\mathcal{A}')^{\mathcal{T}'} = \bigvee_{\mathcal{D} \in \mathfrak{D}} \bigwedge (\mathcal{A}_{\mathcal{D}_\mathcal{U}})^{\mathcal{T}'} \cup \mathcal{D}_\mathcal{U} \cup (\mathcal{D} \setminus \mathcal{D}_\mathcal{U}^\neg)$$

and

$$(\mathcal{A}^\mathcal{T})' = \bigvee_{\mathcal{D} \in \mathfrak{D}} \bigwedge (\mathcal{A}^\mathcal{T})^{\mathcal{D}_\mathcal{U}} \cup \mathcal{D}_\mathcal{U} \cup \mathcal{D}^{\mathcal{D}_\mathcal{U}}.$$

Thus, since it is easy to see that $\mathcal{D}_{\mathcal{D}_\mathcal{U}} \equiv \mathcal{D} \setminus \mathcal{D}_\mathcal{U}^\neg$ for all $\mathcal{D} \in \mathfrak{D}$, it remains to show that $(\mathcal{A}_{\mathcal{D}_\mathcal{U}})^{\mathcal{T}'} \equiv (\mathcal{A}^\mathcal{T})^{\mathcal{D}_\mathcal{U}}$ for all $\mathcal{D} \in \mathfrak{D}$. But this is true due to

$$(A_C^\mathcal{D})^{\mathcal{T}'} \equiv (C^\mathcal{T})^{\mathcal{D}_\mathcal{U}} \quad \text{for all } C \in \mathsf{sub}(\mathcal{K}),$$

which is a consequence of the definition of $\mathcal{T}'$ and can easily be shown by structural induction on C. ❑

We formulate the main result on updates with acyclic TBoxes. In constrast to updates without TBoxes, updated knowledge bases are now polynomial in the size of the original KB. Thus, Lemma 30 implies that we can use acyclic TBoxes to obtain a more succinct presentation of updated ABoxes. In the following, the size $|\mathcal{T}|$ of a TBox $\mathcal{T}$ is $\sum_{A \equiv C \in \mathcal{T}} |C|$, and the size $|\mathcal{K}|$ of a knowledge base $\mathcal{K} = (\mathcal{A}, \mathcal{T})$ is the sum of $|\mathcal{T}|$ and $|\mathcal{A}|$.

Theorem 32. *Let $\mathcal{L} \in \{\mathcal{ALCO}^@, \mathcal{ALCIO}^@, \mathcal{ALCQO}^@, \mathcal{ALCQIO}^@\}$. Then there exist polynomials p_1, p_2, and q such that, for every $\mathcal{L}$-knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and every update $\mathcal{U}$ for $\mathcal{T}$, there exists an $\mathcal{L}$-knowledge base $\mathcal{K}'$ such that*

- $\mathcal{K} * \mathcal{U} \equiv_{\mathsf{P}} \mathcal{K}'$;
- $|\mathcal{K}'| \leq p_1(|\mathcal{K}|) \cdot 2^{p_2(|\mathcal{U}|)}$;
- $\mathcal{K}'$ *can be computed in time* $q(|\mathcal{K}'|)$.

Iterated Updates

As in Section 4, we show that iterated updates do not produce a blowup of the size of updated ABoxes that is worse than the blowup produced by a single update.

Theorem 33. *There exist polynomials p_1, p_2 such that the following holds: for all knowledge bases $\mathcal{K}_0, \ldots, \mathcal{K}_n$ and updates $\mathcal{U}_1, \ldots, \mathcal{U}_n$, if $\mathcal{K}_i$ is the ABox computed by our algorithm when $\mathcal{K}_{i-1}$ is updated with $\mathcal{U}_i$, for $0 < i \leq n$, then*

$$|\mathcal{K}_n| \leq p_1(|\mathcal{K}_0|) \cdot 2^{p_2(|\mathcal{U}_1|+\cdots+|\mathcal{U}_n|)}.$$

Proof. Let $\mathcal{K}_n = (\mathcal{T}_n, \mathcal{A}_n)$ and $|\mathcal{K}_0| = m$. We analyze the sizes of $\mathcal{T}_n$ and $\mathcal{A}_n$ separately:

(a) It is easily seen that $|\mathcal{A}_i| \leq (2 \times |\mathcal{U}_i| + |\mathcal{A}_{i-1}|) \times 2^{|\mathcal{U}_i|} \leq |\mathcal{A}_{i-1}| \times 2^{3|\mathcal{U}_i|}$. Since $|\mathcal{A}_0|$ is bounded by m, it follows that $|\mathcal{A}_i| \leq m \cdot 2^{3(|\mathcal{U}_1|+\cdots+|\mathcal{U}_i|)}$.

(b) For a TBox $\mathcal{T}$, let $||\mathcal{T}||$ denote the number of concept definitions in $\mathcal{T}$. Moreover, let $\mathfrak{D}_i$ be the set of diagrams for $\mathcal{U}_i$. It is not difficult to check that we have

$$||\mathcal{T}_1|| = ||\mathcal{T}_0|| + (||\mathcal{T}_0|| + m) \times |\mathfrak{D}_1|.$$

and, for $i > 1$,

$$||\mathcal{T}_i|| = ||\mathcal{T}_{i-1}|| + (||\mathcal{T}_{i-1}|| + |\mathcal{T}_{i-1}|) \times |\mathfrak{D}_i|.$$

This equation uses $|\mathcal{T}_{i-1}|$ instead if $|\mathcal{K}_{i-1}|$ since $\mathcal{A}_{i-1}$ contains only defined concept names and no complex concepts. Therefore, the cardinality of the $\mathcal{T}^{\mathcal{D}}_{\mathsf{sub}}$ component of $\mathcal{T}_i$ is bounded by $|\mathcal{T}_{i-1}|$.

Since $||\mathcal{T}_{i-1}||$ is bounded by $|\mathcal{T}_{i-1}|$ and the size of each concept equation in $\mathcal{T}_i$, $i > 0$, is bounded by a constant, there is a constant c such that

$$\begin{aligned} |\mathcal{T}_1| &\leq 3cm \times 2^{|\mathcal{U}_1|} \leq m \times 2^{3c|\mathcal{U}_1|} \\ |\mathcal{T}_i| &\leq 3c \times |\mathcal{T}_{i-1}| \times 2^{|\mathcal{U}_i|} \leq |\mathcal{T}_{i-1}| \times 2^{3c|\mathcal{U}_i|} \text{ for } i > 1. \end{aligned}$$

It follows that $|\mathcal{T}_i| \leq m \times 2^{3c(|\mathcal{U}_1|+\cdots+|\mathcal{U}_i|)}$.

The polynomials p_1 and p_2 are now easily derived. ❑

5.2 Updates in $\mathcal{ALCQIO}^{\cup,\setminus,\mathcal{O}}$

As argued at the beginning of Section 5, updated $\mathcal{ALCQIO}^{@}$ ABoxes are only polynomial in the size of the original ABox if the update contains no role assertions. Intuitively, updates with only concept assertions do not lead to an exponential blowup because we have available the Boolean operators on concepts, nominals, and the @-operator. In standard DLs, none of these operators 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 nominals. In this section, we explore the possibility of constructing updated ABoxes in a language in which such constructors are available. The language we consider is closely related to those introduced and investigated in [6, 12, 13], and is of almost the same expressive power as C^2, the two-variable fragment of first-order logic with counting quantifiers [9].

Denote by $\mathcal{ALCQIO}^{+}$ the description logic extending $\mathcal{ALCQIO}^{@}$ by means of the role constructors $\cap$ (role intersection), $-$ (set-theoretic difference of roles), and $\{(a,b)\}$ (nominal roles). In this language, complex roles are constructed starting from role names and nominal roles, and then applying $\cap$, $-$, and the inverse role operator $\cdot^{-}$. The interpretation of complex roles is as expected:

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

We note that reasoning in $\mathcal{ALCQIO}^{+}$ is decidable: this DL can easily be embedded into C^2 and, therefore, ABox consistency is decidable in NExpTime even if the numbers inside number restrictions are coded in binary [9, 15, 17]. This bound is tight as, already in $\mathcal{ALCQIO}$, reasoning is NExpTime-hard [24]. We now formulate the main result of this section:

Theorem 34.
There exist polynomials p_1, p_2, and q such that, for every $\mathcal{ALCQIO}^{+}$-ABox $\mathcal{A}$ and every update $\mathcal{U}$, there is an $\mathcal{ALCQIO}^{+}$-ABox $\mathcal{A}'$ such that

- $\mathcal{A} * \mathcal{U} \equiv \mathcal{A}'$;
- $|\mathcal{A}'| \leq p_1(|\mathcal{A}|) \cdot 2^{p_2(|\mathcal{U}|)}$;
- *$\mathcal{A}'$ can be computed in time $q(|\mathcal{A}'|)$.*

Proof. We modify the proof of Theorem 23. For $\mathcal{ALCQIO}^+$, the construction of the concepts $C^\mathcal{U}$ is much simpler: it suffices to replace every concept name A in C with

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

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 therefore of size polynomial in the size of C and $\mathcal{U}$. The ABox $\mathcal{A}'$ can then be constructed in the same way as in the proof of Theorem 23 and is polynomial in the size of $\mathcal{A}$, but exponential in the size of the update $\mathcal{U}$. ❏

Clearly, Theorem 34 is independent of the coding of numbers, and, also with iterated updates, updated ABoxes remain polynomial in the size of the original ABox. An alternative to working with a description logic such as $\mathcal{ALCQIO}^+$, is to work directly in the two-variable fragment with counting C^2. Then, a result analogous to Theorem 34 is easily obtained.

6 Conclusion

We have analyzed ABox updates in several common description logics. The main outcome of our analysis is as follows: first, in the case of the DLs under consideration, a description logic has updates if and only if it is able to express nominals and the @ constructor (or, equivalently, admits Boolean ABoxes). Second, an exponential blowup cannot by avoided unless NC = PTime. And third, an exponential blowup in the size of the original ABox can be avoided if (i) we allow the introduction of new concept definitions in acyclic TBoxes or (ii) move to DLs that include Boolean operators on roles and a certain nominal constructor for roles, thus eliminating the syntactic disbalance between concepts and roles observed in most DLs. We have also shown that, in the case of repeated updates, there are no repeated exponential blowups.

There are two obvious directions for future work. The first direction is to alleviate the syntactic restriction posed on concepts appearing in updates. This can be done either fully or in a controlled way. In the first case, it is very likely that updated ABoxes cannot be computed even if they exist. However, this has not been proved for some basic DLs such as $\mathcal{ALCO}^@$, and not for all available types of semantics. In the second case, one may for example admit Boolean combinations of concept names in updates. It seems likely that this generalization does not destroy computability of updates.

The second direction for future work is to incorporate cyclic TBoxes or GCIs into our framework. As discussed in [3], it is not at all straightforward to find a semantics for this case that addresses the frame problem (posed by the principle of inertia) in a convincing way. One possible way around this problem is to provide the user with expressive means that allow her to state, in the formulation of the update, the facts that change and the facts that don't. Note that this cannot be done with the updates used in the current paper since they can only talk about domain elements that are assigned a name by some individual name.

Acknowledgements

We are grateful to Franz Baader and Michael Thielscher for stimulating discussions.

References

[1] C. Areces, P. Blackburn, and M. Marx. A road-map on complexity for hybrid logics. In J. Flum and M. Rodríguez-Artalejo, editors, *Computer Science Logic*, number 1683 in Lecture Notes in Computer Science, pages 307–321. Springer-Verlag, 1999.

[2] C. Areces and M. de Rijke. From description logics to hybrid logics, and back. In F. Wolter, H. Wansing, M. de Rijke, and M. Zakharyaschev, editors, *Advances in Modal Logics Volume 3*. CSLI Publications, Stanford, CA, USA, 2001.

[3] F. Baader, C. Lutz, M. Milicic, U. Sattler, and F. Wolter. Integrating description logics and action formalisms: First results. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI-05)*, Pittsburgh, PA, USA, 2005.

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

[5] A. Borgida. Language features for flexible handling of exceptions in information systems. *ACM Transactions on Database Systems*, 10(4):565–603, 1985.

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

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

[8] K. D. Forbus. Introducing actions into qualitative simulations. In *International Joint Conference on Artificial Intelligence (IJCAI-89)*, pages 1273–1279. Morgan Kaufman, 1989.

[9] E. Grädel, M. Otto, and E. Rosen. Two-Variable Logic with Counting is Decidable. In *Proceedings of Twelfth IEEE Symposium on Logic in Computer Science (LICS'97)*, 1997.

[10] N. Kurtonina and M. de Rijke. Expressiveness of concept expressions in first-order description logics. *Artificial Intelligence*, 107(2):303–333, 1999.

[11] F. Lin. Embracing causality in specifying the indeterminate effects of actions. In B. Clancey and D. Weld, editors, *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI-96)*, pages 670–676, Portland, OR, Aug. 1996. MIT Press.

[12] C. Lutz and U. Sattler. The complexity of reasoning with boolean modal logics. In F. Wolter, H. Wansing, M. de Rijke, and M. Zakharyaschev, editors, *Advances in Modal Logics Volume 3*. CSLI Publications, Stanford, CA, USA, 2001.

[13] C. Lutz, U. Sattler, and F. Wolter. Modal logics and the two-variable fragment. In *Annual Conference of the European Association for Computer Science Logic CSL'01*, LNCS, Paris, France, 2001. Springer Verlag.

[14] M.P.Shanahan. *Solving the Frame Problem.* MIT Press, 1997.

[15] L. Pacholski, W. Szwast, and L. Tendera. Complexity results for first-order two-variable logic with counting. *SIAM Journal on Computing*, 29(4):1083–1117, Aug. 2000.

[16] C. H. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[17] I. Pratt-Hartmann. Complexity of the two-variable fragment with counting quantifiers. *Journal of Logic, Language, and Information*, 14(3):369–395, 2005.

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

[19] K. Satoh. Nonmonotonic reasoning by minimal belief revision. In *Proceedings of the International Conference on Fifth Generation Computer Systems. Volume 2*, pages 455–462. Springer Verlag, 1988.

[20] R. Scherl and H. Levesque. Knowledge, action, and the frame problem. *AIJ*, 144(1):1–39, 2003.

[21] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.

[22] M. Thielscher. Nondeterministic actions in the fluent calculus: Disjunctive state update axioms. In S. Hölldobler, editor, *Intellectics and Computational Logic*, pages 327–345. Kluwer Academic, 2000.

[23] M. Thielscher. Representing the knowledge of a robot. In A. Cohn, F. Giunchiglia, and B. Selman, editors, *KR*, pages 109–120, Breckenridge, CO, Apr. 2000. Morgan Kaufmann.

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

[25] M. Winslett. Reasoning about action using a possible models approach. In *Proceedings of the 7th National Conference on Artificial Intelligence (AAAI-88)*, pages 89–93, 1988.

[26] M. Winslett. *Updating Logical Databases*. Cambridge University Press, Cambridge, England, 1990.