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

LTCS–Report

Description Logic Actions with general TBoxes: a Pragmatic Approach

Hongkai Liu, Carsten Lutz, Maja Miličić, Frank Wolter

LTCS-Report 06-03

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

Hans-Grundig-Str. 25
01062 Dresden
Germany

Description Logic Actions with general TBoxes: a Pragmatic Approach

Hongkai Liu[1], Carsten Lutz[1], Maja Miličić[1], Frank Wolter[2]

[1]Inst. für Theor. Inf. TU Dresden, Germany

[2]Dept. of CS Univ. of Liverpool, UK

Abstract

Action formalisms based on description logics (DLs) have recently been introduced as decidable fragments of well-established action theories such as the Situation Calculus and the Fluent Calculus. However, existing DL action formalisms fail to include general TBoxes, which are the standard tool for formalising ontologies in modern description logics. We define a DL action formalism that admits general TBoxes, propose an approach to addressing the ramification problem that is introduced in this way, and perform a detailed investigation of the decidability and computational complexity of reasoning in our formalism.

1 Introduction

Action theories such as the Situation Calculus (SitCalc) and the Fluent Calculus aim at describing actions in a semantically adequate way [12, 15]. They are usually formulated in first- or higher-order logic and do not admit decidable reasoning. For reasoning about actions in practical applications, such theories are thus not directly suited. There are two obvious ways around this problem: the first one is to accept undecidability and replace reasoning by programming. This route is taken by the inventors of action-oriented programming languages such as Golog [6] and Flux [16], whose semantics is based on the SitCalc and Fluent Calculus, respectively. The second one is to try to identify fragments of action theories such as SitCalc that are sufficiently expressive to be useful in applications, but nevertheless admit decidable reasoning. For example, a simple such fragment is obtained by allowing only propositional logic for describing the state of the world and pre- and post-conditions of actions. A much more expressive formalism was identified in our recent paper [2], where we define action formalisms that are based on description logics (DLs) [3]. More precisely, we use DL ABoxes to describe the state of the world and pre- and post-conditions of actions and prove that reasoning in the resulting formalism is decidable [2]. We also show in [2] that, in this way, we actually get a decidable fragment of SitCalc.

In description logic, TBoxes are used as an ontology formalism, i.e., to define concepts and describe relations between them. For example, a TBox may describe relevant concepts from the domain of universities such as lecturers, students, courses, and libraries. From the reasoning about actions perspective, TBoxes correspond to state constraints. For example, a TBox for the university domain could state that every student that is registered for a course has access to a university library. If we execute an action that registers the student Dirk for a computer science course, then after the action Dirk should also have access to a university library to comply with the state constraint imposed by the TBox. Thus, general TBoxes as state constraints induce a ramificiation problem which we henceforth call the *TBox ramification problem*.

Regarding TBoxes, the DL action formalism defined in [2] has two major limitations: first, we only admit *acyclic TBoxes* which are a much more lightweight ontology formalism than the *general TBoxes* that can be found in all state-of-the-art DL reasoners [**?**]. For example, the DL formulation of the above ontology statement regarding access to libraries requires a general concept inclusion (GCIs) as offered by general TBoxes. Second, we allow only concept names (but no complex concepts) in post-conditions and additionally stipulate that these concept names are *not* defined in the TBox. In the present paper, we present an approach to overcoming these limitations while retaining decidability of the most important reasoning tasks. In particular, we show how to incorporate general TBoxes into DL action formalisms. This implies dropping the second restriction as well since there is no clear notion of a concept name "being defined" in a general TBox.

The main reason for adopting the mentioned restrictions in [2] was that they disarm the TBox ramification problem. Attempts to *automatically* solve the TBox ramificiation problem, e.g. by adopting a Winslett-style PMA semantics [19], lead to semantic and computational problems: we show in [2] that counter-intuitive results and undecidability of reasoning are the consequence of adopting such a semantics. Since there appears to be no general automated solution to the TBox ramification problem other than resorting to very inexpressive DLs [**?**], we propose to leave it to the designer of an action description to fine-tune the ramifications of the action. This is similar to the approach taken in the SitCalc and the Fluent Calculus to address the ramification problem. There, the designer of an action description can control the ramifications of the action by specifying causal relationships between predicates [7, 14]. While causality appears to be a satisfactory approach for addressing the ramification problem that is induced by Boolean state constraints, it seems not powerful enough for

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	$(\leqslant n\ r\ C)$	$\{x \in \Delta^\mathcal{I} \mid \#\{y \in C^\mathcal{I} \mid (x,y) \in r^\mathcal{I}\} \leq n\}$

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

attacking the ramifications introduced by general TBoxes, which usually involve complex quantification patterns. We therefore advocate a different approach: when describing an action, the user can specify the predicates that can change through the execution of the action, as well as those that cannot change. To allow an adequate fine-tuning of ramifications, we admit complex statements about the change of predicates such as "the concept name A can change from positive to negative only at the individual a, and from negative to positive only where the complex concept C was satisfied before the action was executed".

The family of action formalisms introduced in this paper can be parameterised with any description logic. We show that, for many standard DLs, the reasoning problems *executability* and *projection* in the corresponding action formalism are decidable. We also pinpoint the exact computational complexity of these reasoning problems. As a rule of thumb, our results show that reasoning in the action formalism instantiated with a description logic $\mathcal{L}$ is of the same complexity as standard reasoning in $\mathcal{L}$ extended with nominals (which correspond to first-order constants [1]). For fine-tuning the ramifications, consistency of actions is an important property. We introduce two notions of consistency (weak and strong) and show that weak consistency is of the same complexity as deciding projection while strong consistency is undecidable even when the action formalism is instantiated with the basic DL $\mathcal{ALC}$. Details regarding the technical results can be found in the report [8].

2 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*, 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 remaining six 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, and $\mathcal{O}$ for nominals. This explains the name $\mathcal{ALCQIO}$ for our DL, and also allows us to refer to its sublanguages in a simple way.

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.

A *general concept inclusion axiom (GCI)* is an expression of the form $C \sqsubseteq D$, where C and D are concepts. An expression $C \doteq D$ is an abbreviation for two GCIs $C \sqsubseteq D$ and $D \sqsubseteq C$. A *(general) TBox* $\mathcal{T}$ is a finite set of GCIs. An 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). An interpretation $\mathcal{I}$ *satisfies* a GCI $C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, 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 a GCI $C \sqsubseteq D$ by an interpretation $\mathcal{I}$ with $\mathcal{I} \models C \sqsubseteq D$, and similar for ABox assertions. An interpretation $\mathcal{I}$ is a *model* of a TBox $\mathcal{T}$ (written $\mathcal{I} \models \mathcal{T}$) iff it satisfies all GCIs in $\mathcal{T}$. It is a *model* of an ABox $\mathcal{A}$ (written $\mathcal{I} \models \mathcal{A}$) iff it satisfies all assertions in $\mathcal{A}$.

A concept C is satisfiable w.r.t. a TBox $\mathcal{T}$ iff $C^{\mathcal{I}} \neq \emptyset$ for some model $\mathcal{I}$ of $\mathcal{T}$. An ABox $\mathcal{A}$ is consistent w.r.t. a TBox $\mathcal{T}$ iff $\mathcal{A}$ and $\mathcal{T}$ have a common model.

3 Describing Actions

The action formalism proposed in this paper is not restricted to a particular DL. However, for our complexity results we consider the DL $\mathcal{ALCQIO}$ and its sublogics.

The main syntactic ingredients of our approach to reasoning about actions are action descriptions, ABoxes for describing the current knowledge about the state of affairs in the application domain, and TBoxes for describing general knowledge about the application domain similar to state constraints in the SitCalc and Fluent Calculus. On the semantic side, interpretations are used to describe the state of affairs in the application domain. Thus, the knowledge described by an ABox is incomplete: ABoxes may admit more than a single model, and all the corresponding states are considered possible. Before we go deeper into the semantics, we introduce the syntax of action descriptions. We use $\mathcal{LO}$ to denote the extension of a description logic $\mathcal{L}$ with nominals. A *concept literal* is a concept name or the negation thereof, and a *role literal* is defined analogously.

Definition 1 (Action). Let $\mathcal{L}$ be a description logic. An *atomic $\mathcal{L}$-action* $\alpha = (\mathsf{pre}, \mathsf{occ}, \mathsf{post})$ consists of

- a finite set pre of $\mathcal{L}$ ABox assertions, the *pre-conditions*;
- the *occlusion pattern* occ which is a set of mappings $\{\mathsf{occ}_{\varphi_1}, \ldots, \mathsf{occ}_{\varphi_n}\}$ indexed by $\mathcal{L}$ ABox assertions $\varphi_1, \ldots, \varphi_n$ such that each occ_{φ_i} assigns
  - to every concept literal B an $\mathcal{LO}$-concept $\mathsf{occ}_{\varphi_i}(B)$,
  - to every role literal s a finite set $\mathsf{occ}_{\varphi_i}(s)$ of pairs of $\mathcal{LO}$-concepts.
- a finite set post of *conditional post-conditions* of the form φ/ψ, where φ and ψ are $\mathcal{L}$ ABox assertions.

A *composite action* is a finite sequence of atomic actions $\alpha_1, \ldots, \alpha_n$. △

Intuitively, the pre-conditions specify under which conditions the action is applicable. A post-condition φ/ψ says that, if φ is true before executing the action, then ψ should be true afterwards. The purpose of the occlusion patterns is to control ramifications: they provide a description of where concept and role names may change during the execution of an action. More precisely, suppose $\mathsf{occ} = \{\mathsf{occ}_{\varphi_1}, \ldots, \mathsf{occ}_{\varphi_n}\}$ and $\varphi_{i_1}, \ldots, \varphi_{i_m}$ are the assertions which are true before the action was executed. If A is a concept name, then instances of the concept

$$\mathsf{occ}_{\varphi_{i_1}}(A) \sqcup \cdots \sqcup \mathsf{occ}_{\varphi_{i_m}}(A)$$

may change from A to $\neg A$ during the execution of the action provided, but instances of $\neg(\mathsf{occ}_{\varphi_{i_1}}(A) \sqcup \cdots \sqcup \mathsf{occ}_{\varphi_{i_m}}(A))$ may not. Likewise, instances of

$$\mathsf{occ}_{\varphi_{i_1}}(\neg A) \sqcup \cdots \sqcup \mathsf{occ}_{\varphi_{i_m}}(\neg A)$$

may change from $\neg A$ to A. For role names, $(C, D) \in \mathsf{occ}_{\varphi_{i_k}}(r)$ means that pairs from $C^{\mathcal{I}} \times D^{\mathcal{I}}$ that have been connected by r before the action may lose this connection through the execution of the action, and similarly for the occlusion of negated role names. More details on how occlusions relate to ramifications will be given after we have introduced the semantics.

For defining the semantics in a succinct way, it is convenient to introduce the following abbreviation. For an action α with $\mathsf{occ} = \{\mathsf{occ}_{\varphi_1}, \ldots, \mathsf{occ}_{\varphi_n}\}$, an interpretation $\mathcal{I}$, a concept literal B (a concept name or the negation thereof), and a role literal s (a role name or the negation thereof), we set

$$\begin{array}{rcl} (\mathsf{occ}(B))^{\mathcal{I}} & := & \bigcup_{\mathcal{I} \models \varphi_i} (\mathsf{occ}_{\varphi_i}(B))^{\mathcal{I}} \\ (\mathsf{occ}(s))^{\mathcal{I}} & := & \bigcup_{(C,D) \in \mathsf{occ}_{\varphi_i}(s), \mathcal{I} \models \varphi_i} (C^{\mathcal{I}} \times D^{\mathcal{I}}) \end{array}$$

Thus, $\mathsf{occ}(X)^{\mathcal{I}}$ describes those elements of $\Delta^{\mathcal{I}}$ that may change from X to $\neg X$ when going to $\mathcal{I}'$, and similarly for $\mathsf{occ}(s)^{\mathcal{I}}$.

Definition 2 (Action semantics). Let $\alpha = (\mathsf{pre}, \mathsf{occ}, \mathsf{post})$ be an atomic action and $\mathcal{I}, \mathcal{I}'$ interpretations sharing the same domain and interpretation of all individual names. We say that α *may transform* $\mathcal{I}$ *to* $\mathcal{I}'$ w.r.t. a TBox $\mathcal{T}$ ($\mathcal{I} \Rightarrow_{\alpha}^{\mathcal{T}} \mathcal{I}'$) iff the following holds:

- $\mathcal{I}, \mathcal{I}'$ are models of $\mathcal{T}$;
- for all $\varphi/\psi \in \mathsf{post}$: $\mathcal{I} \models \varphi$ implies $\mathcal{I}' \models \psi$ (written $\mathcal{I}, \mathcal{I}' \models \mathsf{post}$);
- for each $A \in \mathsf{N_C}$ and $r \in \mathsf{N_R}$, we have

$$\begin{array}{rcll} A^{\mathcal{I}} \setminus A^{\mathcal{I}'} & \subseteq & (\mathsf{occ}(A))^{\mathcal{I}} & \neg A^{\mathcal{I}} \setminus \neg A^{\mathcal{I}'} \subseteq (\mathsf{occ}(\neg A))^{\mathcal{I}} \\ r^{\mathcal{I}} \setminus r^{\mathcal{I}'} & \subseteq & (\mathsf{occ}(r))^{\mathcal{I}} & \neg r^{\mathcal{I}} \setminus \neg r^{\mathcal{I}'} \subseteq (\mathsf{occ}(\neg r))^{\mathcal{I}} \end{array}$$

The composite action $\alpha_1, \ldots, \alpha_n$ *may transform* $\mathcal{I}$ to $\mathcal{I}'$ w.r.t. $\mathcal{T}$ ($\mathcal{I} \Rightarrow_{\alpha_1,\ldots,\alpha_n}^{\mathcal{T}} \mathcal{I}'$) iff there are models $\mathcal{I}_0, \ldots, \mathcal{I}_n$ of $\mathcal{T}$ with $\mathcal{I} = \mathcal{I}_0$, $\mathcal{I}' = \mathcal{I}_n$, and $\mathcal{I}_{i-1} \Rightarrow_{\alpha_i}^{\mathcal{T}} \mathcal{I}_i$ for $1 \leq i \leq n$. △

Let us reconsider the example from the introduction to explain how occlusions provide a way to control the ramifications induced by general TBoxes. The TBox $\mathcal{T}$ contains the following GCIs which say that everybody registered for a course has access to a university library, and that every university has a library:

$$\begin{array}{rcl} \exists \mathsf{registered\_for.Course} & \sqsubseteq & \exists \mathsf{access\_to.Library} \\ \mathsf{University} & \sqsubseteq & \exists \mathsf{has\_facility.Library} \end{array}$$

The upper GCI cannot be expressed in terms of an acyclic TBox and is thus outside the scope of the formalism in [2]. The ABox $\mathcal{A}$ which describes the current state of the world (in an incomplete way) says that computer science is a course held at TU Dresden, SLUB is the library of TU Dresden, and Dirk is neither registered for a course nor has access to a library:

$$\begin{array}{lll} \mathsf{Course}(\mathsf{cs}) & \mathsf{held\_at}(\mathsf{cs},\mathsf{tud}) & \neg\exists \mathsf{registered\_for}.\mathsf{Course}(\mathsf{dirk}) \\ \mathsf{University}(\mathsf{tud}) & \mathsf{has\_facility}(\mathsf{tud},\mathsf{slub}) & \neg\exists \mathsf{access\_to}.\mathsf{Library}(\mathsf{dirk}) \\ \mathsf{Library}(\mathsf{slub}) & & \end{array}$$

The action

$$\alpha := (\emptyset, \mathsf{occ}, \{\mathsf{taut}/\mathsf{registered\_for}(\mathsf{dirk},\mathsf{cs})\})$$

describes the registration of Dirk for the computer science course. For simplicity, the set of pre-conditions is empty and taut is some ABox assertion that is trivially satisfied, say $\top(\mathsf{cs})$. To obtain occ, we may start by strictly following the law of inertia, i.e., requiring that the only changes are those that are explicitly stated in the post-condition. Thus, occ consists of just one mapping $\mathsf{occ}_{\mathsf{taut}}$ such that

$$\mathsf{occ}_{\mathsf{taut}}(\neg\mathsf{registered\_for}) := \{(\{\mathsf{dirk}\}, \{\mathsf{cs}\})\}$$

and all concept and role literals except $\neg\mathsf{registered\_for}$ are mapped to $\bot$ and $\{(\bot,\bot)\}$, respectively. This achieves the desired effect that only the pair $(\mathsf{dirk},\mathsf{cs})$ can be added to "$\mathsf{registered\_for}$" and nothing else can be changed.

It is not hard to see that this attempt to specify occlusions for α is too strict. Intuitively, not allowing any changes is appropriate for Course, $\mathsf{Library}$, $\mathsf{University}$, $\mathsf{held\_at}$, $\mathsf{has\_facility}$ and their negations since the action should have no impact on these predicates. However, not letting $\neg\mathsf{access\_to}$ change leads to a problem with the ramifications induced by the TBox: as Dirk has no access to a library before the action and $\neg\mathsf{access\_to}$ is not allowed to change, he cannot have access to a library after execution of the action as required by the TBox. Thus, the action is inconsistent in the following sense: there is no model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$ and model $\mathcal{I}'$ of $\mathcal{T}$ such that $\mathcal{I} \Rightarrow_\alpha^\mathcal{T} \mathcal{I}'$. To take care of the TBox ramifications and regain consistency, we can modify occ. One option is to set

$$\mathsf{occ}_{\mathsf{taut}}(\neg\mathsf{access\_to}) := \{(\{\mathsf{dirk}\}, \mathsf{Library})\}$$

and thus allow Dirk to have access to a library after the action. Another option is to set

$$\mathsf{occ}_{\mathsf{taut}}(\neg\mathsf{access\_to}) := \{(\{\mathsf{dirk}\}, \mathsf{slub})\}$$

which allows Dirk to have access to SLUB after the action, but not to any other library.

Two remarks regarding this example are in order. First, the occlusion occ consists only of a single mapping $\mathsf{occ}_{\mathsf{taut}}$. The reason for this is that there is only a single post-condition in the action. If we have different post-conditions φ/ψ and φ'/ψ such that φ and φ' are not equivalent, there will usually be different occlusion mappings (indexed with φ and φ') to deal with the ramifications that the TBox induces for these post-conditions. Second, the example explains the need for extending $\mathcal{L}$ to $\mathcal{LO}$ when describing occlusions (c.f. Definition 1): without nominals, we would not have been able to properly formulate the occlusions although all other parts of the example are formulated without using nominals (as a concept-forming operator).

As illustrated by the example, it is important for the action designer to decide consistency of actions to detect ramification problems that are not properly addressed by the occlusions. In the following, we propose two notions of consistency.

Definition 3 (Consistency). Let $\alpha = (\mathsf{pre}, \mathsf{occ}, \mathsf{post})$ be an atomic action and $\mathcal{T}$ a TBox. We say that

- α is *weakly consistent with* $\mathcal{T}$ iff there are models $\mathcal{I}, \mathcal{I}'$ of $\mathcal{T}$ such that $\mathcal{I} \models \mathsf{pre}$ and $\mathcal{I} \Rightarrow^{\mathcal{T}}_{\alpha} \mathcal{I}'$.
- α is *strongly consistent with* $\mathcal{T}$ iff for all models $\mathcal{I}$ of $\mathcal{T}$ and pre, there is a model $\mathcal{I}'$ of $\mathcal{T}$ such that $\mathcal{I} \Rightarrow^{\mathcal{T}}_{\alpha} \mathcal{I}'$.

$\triangle$

Intuitively, strong consistency is the most desirable form of consistency: if the preconditions of an action are satisfied by an interpretation $\mathcal{I}$, then the action can transform $\mathcal{I}$ into a new interpretation $\mathcal{I}'$. Unfortunately, strong consistency will turn out to be undecidable. For this reason we introduce also weak consistency, which is still sufficient to detect serious ramification problems. In the example above, the first attempt to define the occlusions results in an action that is not even weakly consistent. After each of the two possible modifictions, the action is strongly consistent. We will see later that weak consistency is decidable while strong consistency is not.

To check whether an action can be applied in a given situation, the user wants to know whether it is executable, i.e., whether all pre-conditions are satisfied in the states of the world considered possible. If the action is executable, he wants to know whether applying it achieves the desired effect, i.e., whether an assertion that he wants to make true really holds after executing the action. These two problems are called executability and projection [12, 2].

Definition 4 (Executability and projection). Let $\alpha_1, \ldots, \alpha_n$ be a composite action with $\alpha_i = (\mathsf{pre}_i, \mathsf{occ}_i, \mathsf{post}_i)$ for $i = 1, \ldots, n$, let $\mathcal{T}$ be a TBox, and $\mathcal{A}$ an ABox.

- Executablity: $\alpha_1, \ldots, \alpha_n$ is *executable in $\mathcal{A}$ w.r.t. $\mathcal{T}$* iff the following conditions are true for all models $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$:
  - $\mathcal{I} \models \mathsf{pre}_1$

– for all i with $1 \leq i < n$ and all interpretations $\mathcal{I}'$ with $\mathcal{I} \Rightarrow^{\mathcal{T}}_{\alpha_1,\ldots,\alpha_i} \mathcal{I}'$, we have $\mathcal{I}' \models \mathsf{pre}_{i+1}$.

- Projection: The assertion φ is a *consequence of applying* $\alpha_1, \ldots, \alpha_n$ *in* $\mathcal{A}$ *w.r.t.* $\mathcal{T}$ iff for all models $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$ and for all $\mathcal{I}'$ with $\mathcal{I} \Rightarrow^{\mathcal{T}}_{\alpha_1,\ldots,\alpha_n} \mathcal{I}'$, we have $\mathcal{I}' \models \varphi$.

$\triangle$

To make sure that a composite action $\alpha = \alpha_1, \ldots, \alpha_n$ can be successfully executed, α has to be executable and the atomic actions $\alpha_1, \ldots, \alpha_n$ have to be strongly consistent: without strong consistency, it could be that although the action α is executable w.r.t. the ABox $\mathcal{A}$ describing the knowledge about the current state of the world, the actual state of the world $\mathcal{I}$ is such that there is no interpretation $\mathcal{I}'$ with $\mathcal{I} \Rightarrow^{\mathcal{T}}_{\alpha} \mathcal{I}'$. Even worth, such a situation may arise also after we have already executed some of the atomic actions in the sequence α.

It is not difficult to see that the action formalism just introduced is a generalisation of the one introduced in [2] when composite actions are disallowed, for details see Appendix A. Clearly, executability can be polynomially reduced to *ABox consequence* which is defined as follows: given an ABox $\mathcal{A}$ and an assertion φ, decide whether $\mathcal{I}$ satisfies φ in all models $\mathcal{I}$ of $\mathcal{A}$. The complexity of this problem is extensively discussed in [2]. For example, it is NExpTime-complete for $\mathcal{ALCQIO}$ and ExpTime-complete for $\mathcal{ALC}$ extended with at most two of $\mathcal{Q}$, $\mathcal{I}$, and $\mathcal{O}$.

It can also be seen that (i) an action $\alpha = (\mathsf{pre}, \mathsf{occ}, \mathsf{post})$ is weakly consistent with a TBox $\mathcal{T}$ iff $\bot(a)$ is not a consequence of applying α in pre w.r.t. $\mathcal{T}$; (ii) φ is a consequence of applying $\alpha = (\mathsf{pre}, \mathsf{occ}, \mathsf{post})$ in $\mathcal{A}$ w.r.t. $\mathcal{T}$ iff the action $(\mathcal{A} \cup \mathsf{pre}, \mathsf{occ}, \mathsf{post} \cup \{\top(a)/\neg\varphi\})$ is not weakly consistent with $\mathcal{T}$. Thus, weak consistency can be reduced to (non-)projection and vice versa and complexity results carry over from one to the other. In this paper, we will concentrate on projection.

4 Projection in ExpTime

We show that projection and weak consistency are ExpTime-complete for DL actions formulated in $\mathcal{ALC}$, $\mathcal{ALCO}$, $\mathcal{ALCI}$, $\mathcal{ALCIO}$. Thus, in these DLs reasoning about actions is not more difficult than the standard DL reasoning problems such as concept satisfiability and subsumption w.r.t. TBoxes. The complexity results established in this section are obtained by proving that projection in $\mathcal{ALCIO}$ is in ExpTime. We use a Pratt-style type elimination technique as first proposed in [10].

Let $\alpha_1, \ldots, \alpha_n$ be a composite action with $\alpha_i = (\mathsf{pre}_i, \mathsf{occ}_i, \mathsf{post}_i)$ for $1 \leq i \leq n$, $\mathcal{T}$ a TBox, $\mathcal{A}_0$ an ABox and φ_0 an assertion. We want to decide whether φ_0 is a consequence of applying $\alpha_1, \ldots, \alpha_n$ in $\mathcal{A}_0$ w.r.t. $\mathcal{T}$. In what follows, we call $\alpha_1, \ldots, \alpha_n$, $\mathcal{T}$, $\mathcal{A}_0$ and φ_0 the *input*. W.l.o.g., we make the following assumptions:

- concepts used in the input are built only from the constructors $\{a\}$, $\neg$, $\sqcap$, and $\exists r.C$;
- φ_0 is of the form $\varphi_0 = C_0(a_0)$, where C_0 is a (complex) concept;
- $\mathcal{A}_0$ and $\alpha_1, \ldots, \alpha_n$ contain only concept assertions.

The last two assumptions can be made because every assertion $r(a, b)$ can be replaced with $(\exists r.\{b\})(a)$, and every $\neg r(a, b)$ with $(\neg\exists r.\{b\})(a)$.

Before we can describe the algorithm, we introduce several notions and abbreviations. With Sub, we denote the set of subconcepts of the concepts which occur in the input. With Ind, we denote the set of individual names used in the input, and set $\mathsf{Nom} := \{\{a\} \mid a \in \mathsf{Ind}\}$. The algorithm that we give in the following checks for the existence of a countermodel witnessing that φ_0 is *not* a consequence of applying $\alpha_1, \ldots, \alpha_n$ in $\mathcal{A}_0$ w.r.t. $\mathcal{T}$. Such a countermodel consists of $n+1$ interpretations $\mathcal{I}_0, \ldots, \mathcal{I}_n$ such that $\mathcal{I}_0 \models \mathcal{A}_0$, $\mathcal{I}_0 \Rightarrow^{\mathcal{T}}_{\alpha_1} \mathcal{I}_1, \ldots, \mathcal{I}_{n-1} \Rightarrow^{\mathcal{T}}_{\alpha_n} \mathcal{I}_n$, and $\mathcal{I}_n \not\models \varphi_0$. To distinguish the extension of concept and role names in the different interpretations, we introduce concept names $A^{(i)}$ and role names $r^{(i)}$ for every concept name A and role name r used in the input, and every $i \leq n$. For a concept $C \in \mathsf{Sub}$ that is not a concept name, we use $C^{(i)}$, $i \leq n$, to denote the concept obtained by replacing all concept names A and role names r occurring in C by $A^{(i)}$ and $r^{(i)}$ respectively. We define the set of concepts Cl as:

$$\mathsf{Cl} = \bigcup_{i \leq n} \{C^{(i)}, \neg C^{(i)} \mid C \in \mathsf{Sub} \cup \mathsf{Nom}\}$$

The notion of a type plays a central role in the projection algorithm to be devised.

Definition 5. A set of concepts $t \subseteq \mathsf{Cl}$ is a *type* for Cl iff it satisfies the following conditions:

- for all $\neg D \in \mathsf{Cl}$: $\neg D \in t$ iff $D \notin t$;
- for all $D \sqcap E \in \mathsf{Cl}$: $D \sqcap E \in t$ iff $\{D, E\} \subseteq t$;
- for all $C \sqsubseteq D \in \mathcal{T}$ and $i \leq n$, $C^{(i)} \in t$ implies $D^{(i)} \in t$;

A type is *anonymous* if it does not contain a nominal. Let $\mathfrak{T}_{\mathsf{ano}}$ be the set of all anonymous types. △

Intuitively, a type describes the concept memberships of a domain element in all $n + 1$ interpretations. Our algorithm starts with a set containing (almost) all types, then repeatedly eliminates those types that cannot be realized in a countermodel witnessing that φ_0 is not a consequence of applying $\alpha_1, \ldots, \alpha_n$ in $\mathcal{A}_0$ w.r.t. $\mathcal{T}$, and finally checks whether the surviving types give rise to such a countermodel. The picture is slightly complicated by the presence of ABoxes and nominals. These are treated via core type sets to be introduced next.

Definition 6. $\mathfrak{T}_S$ is a *core type set* iff $\mathfrak{T}_S$ is a minimal set of types such that, for all $a \in \mathsf{Ind}$, there is a $t \in \mathfrak{T}_S$ with $\{a\} \in \mathfrak{T}_S$.

A core type set $\mathfrak{T}_S$ is called *proper* if the following conditions are satisfied:

1. for all $C(a) \in \mathcal{A}_0$, $\{a\} \in t \in \mathfrak{T}_S$ implies $C^{(0)} \in t$;
2. for all $C(a)/D(b) \in \mathsf{post}_i$, $1 \leq i \leq n$: if there is a $t \in \mathfrak{T}_S$ with $\{\{a\}, C^{(i-1)}\} \subseteq t$ then there is a $t' \in \mathfrak{T}_S$ with $\{\{b\}, D^{(i)}\} \subseteq t'$.

$\triangle$

Intuitively, a core type set carries information about the "named" part of the interpretations $\mathcal{I}_0, \ldots, \mathcal{I}_n$, where the named part of an interpretation consists of those domain elements that are identified by nominals. Let m be the size of the input. It is not difficult to check that the number of core type sets is exponential in m. Also, checking whether a core type set is proper can be done in linear time.

The following definition specifies the conditions under which a type is eliminated. We start with introducing some convenient abbreviations. Consider an action $\alpha_\ell = (\mathsf{pre}_\ell, \mathsf{occ}_\ell, \mathsf{post}_\ell)$. For a role name r and $\mathsf{occ}_\varphi \in \mathsf{occ}_\ell$, we set $\mathsf{occ}_\varphi(r^-) := \{(Y, X) \mid (X, Y) \in \mathsf{occ}_\varphi(r)\}$, and analogously for $\mathsf{occ}(\neg r^-)$. Let t, t' be types, $\mathfrak{T}$ a set of types, $C(a)$ an ABox assertion, B a concept literal, and s a role literal. We write

- $\mathfrak{T} \models C(a)$ if there exists a $t \in \mathfrak{T}$ with $\{\{a\}, C\} \subseteq t$;
- $t, \mathfrak{T} \models \mathsf{occ}_\ell(B)$ if there is an $\mathsf{occ}_{C(a)} \in \mathsf{occ}_\ell$ such that (i) $\mathfrak{T} \models C(a)$ and (ii) for the concept $D = \mathsf{occ}_{C(a)}(B)$, we have $D^{(\ell)} \in t$;
- $t, t', \mathfrak{T} \models \mathsf{occ}_\ell(s)$ if there is an $\mathsf{occ}_{C(a)} \in \mathsf{occ}_\ell$ and a pair $(D, E) \in \mathsf{occ}_{C(a)}(s)$ such that (i) $\mathfrak{T} \models C(a)$, (ii) $D^{(\ell)} \in t$, and (iii) $E^{(\ell)} \in t'$.

Intuitively, $t, \mathfrak{T} \models \mathsf{occ}_\ell(B)$ states that when the action α_ℓ is executed in a model that realizes only types from $\mathfrak{T}$, then instances of t may change from B to $\neg B$.

For a role r, we use $\mathsf{Inv}(r)$ to denote r^- if r is a role name and s if $r = s^-$.

Definition 7. Let $\mathfrak{T}$ be a set of types for Cl. Then a type $t \in \mathfrak{T}$ is *good in* $\mathfrak{T}$ iff

$\mathcal{ALCIO}\text{-elim}(\mathcal{A}_0, \mathcal{T}, \alpha_1, \ldots, \alpha_n, \varphi_0)$
  for all proper core type sets $\mathfrak{T}_S$ do
    $i := 0$;
    $\mathfrak{T}_0 := \mathfrak{T}_S \cup \mathfrak{T}_{\mathsf{ano}}$
    repeat
      $\mathfrak{T}_{i+1} := \{t \in \mathfrak{T}_i \mid t \text{ is good in } \mathfrak{T}_i\}$;
      $i := i + 1$;
    until $\mathfrak{T}_i = \mathfrak{T}_{i-1}$;
    if $\mathfrak{T}_S \subseteq \mathfrak{T}_i$ and there is a $t \in \mathfrak{T}_i$ with $\{\{a_0\}, \neg C_0^{(n)}\} \subseteq t$ then
      return false
    endif
  endfor
  return true

Figure 2: The type elimination algorithm.

- for all concept names A and $i < n$:

  (C1) if $\{A^{(i)}, \neg A^{(i+1)}\} \subseteq t$, then $t, \mathfrak{T} \models \mathsf{occ}_{i+1}(A)$;

  (C2) if $\{\neg A^{(i)}, A^{(i+1)}\} \subseteq t$, then $t, \mathfrak{T} \models \mathsf{occ}_{i+1}(\neg A)$.

- for all $(\exists r.C)^{(i)} \in t$, there exists a type $t' \in \mathfrak{T}$ and a set $\rho \subseteq \{0, \ldots, n\}$ such that for all $\ell \leq n$, the following conditions are satisfied:

  (R1) $C^{(i)} \in t'$ and $i \in \rho$;

  (R2) if $(\neg\exists r.D)^{(\ell)} \in t$ and $\ell \in \rho$, then $\neg D^{(\ell)} \in t'$;

  (R3) if $(\neg\exists \mathsf{Inv}(r).D)^{(\ell)} \in t'$ and $\ell \in \rho$, then $\neg D^{(\ell)} \in t$;

  (R4) if $n > \ell \in \rho$ and $\ell + 1 \notin \rho$ then $t, t', \mathfrak{T} \models \mathsf{occ}_{\ell+1}(r)$;

  (R5) if $n > \ell \notin \rho$ and $\ell + 1 \in \rho$ then $t, t', \mathfrak{T} \models \mathsf{occ}_{\ell+1}(\neg r)$.

$\triangle$

Intuitively, the above definition checks whether there can be any instances of t in an interpretation in which all domain elements have a type in $\mathfrak{T}$. More precisely, t' is the type that is needed to satisfy the existential restriction $(\exists r.C)^{(i)} \in t$. The set ρ determines the extension of the role r: if $\ell \in \rho$, then the instance of t' that we introduce as a witness for $(\exists r.C)^{(i)}$ is reachable via an r-edge from the instance of t in the interpretation $\mathcal{I}_\ell$.

The type elimination algorithm is given in a pseudo-code notation in Figure 2, where C_0 is the concept from the ABox assertion $\varphi_0 = C_0(a_0)$.

Lemma 8. *$\mathcal{ALCIO}\text{-elim}(\mathcal{A}_0, \mathcal{T}, \alpha_1, \ldots, \alpha_n, \varphi_0)$ returns true iff φ_0 is a consequence of applying $\alpha_1, \ldots, \alpha_n$ in $\mathcal{A}_0$ w.r.t. $\mathcal{T}$.*

Proof. "$\Rightarrow$". We prove this direction by contraposition. Assume that φ_0 is not a consequence of applying $\alpha_1, \ldots, \alpha_n$ in $\mathcal{A}_0$ w.r.t. $\mathcal{T}$. Then there are $\mathcal{I}_0, \ldots, \mathcal{I}_n$ such that $\mathcal{I}_i \Rightarrow^{\mathcal{T}}_{\alpha_{i+1}} \mathcal{I}_{i+1}$ for all $i < n$, $\mathcal{I}_0 \models \mathcal{A}_0$, and $\mathcal{I}_n \not\models C_0(a_0)$. We show that the algorithm returns false in this case.
Let $\Delta := \Delta^{\mathcal{I}_0}$ $(= \cdots = \Delta^{\mathcal{I}_n})$ and $x \in \Delta$. We define

$$t_{\mathsf{Cl}}(x) := \{C^{(i)} \in \mathsf{Cl} \mid x \in C^{\mathcal{I}_i} \text{ for some } i \leq n\}.$$

Claim 1. For all $x \in \Delta$, $t_{\mathsf{Cl}}(x)$ is a type for Cl.

Proof: For all $x \in \Delta$ and all $j \leq n$, the following holds:

- for all $\neg C^{(j)} \in \mathsf{Cl}$: $(\neg C)^{(j)} \in t_{\mathsf{Cl}}(x)$ iff $x \in (\neg C)^{\mathcal{I}_j}$ iff $x \notin C^{\mathcal{I}_j}$ iff $C^{(j)} \notin t_{\mathsf{Cl}}(x)$;
- for all $(C \sqcap D)^{(j)} \in \mathsf{Cl}$: $(C \sqcap D)^{(j)} \in t_{\mathsf{Cl}}(x)$ iff $x \in (C \sqcap D)^{\mathcal{I}_j}$ iff $x \in C^{\mathcal{I}_j}$ and $x \in D^{\mathcal{I}_j}$ iff $C^{(j)} \in t_{\mathsf{Cl}}(x)$ and $D^{(j)} \in t_{\mathsf{Cl}}(x)$ iff $\{C, D\} \subseteq t_{\mathsf{Cl}}(x)$;
- for all $C \sqsubseteq D \in \mathcal{T}$, $\mathcal{I}_j \models \mathcal{T}$ implies that if $x \in C^{\mathcal{I}_j}$, then $x \in D^{\mathcal{I}_j}$. Thus, $C^{(j)} \in t_{\mathsf{Cl}}(x)$ implies $D^{(j)} \in t_{\mathsf{Cl}}(x)$;

This finishes the proof of Claim 1. We set $\mathfrak{T}_S := \{t_{\mathsf{Cl}}(a^{\mathcal{I}_0}) \mid a \in \mathsf{Ind}\}$ and $\mathfrak{T} := \{t_{\mathsf{Cl}}(x) \mid x \in \Delta\}$. Then we have the following:

Claim 2: $\mathfrak{T}_S$ is a proper core type set.

Proof: By the definition of $\mathfrak{T}_S$, it is easy to see that it is a core type set. Moreover, it is proper:

1. for all $C(a) \in \mathcal{A}_0$, $\mathcal{I}_0 \models \mathcal{A}_0$ implies $\mathcal{I}_0 \models C(a)$. Thus, we know that $a^{\mathcal{I}_0} \in C^{\mathcal{I}_0}$. $\{a\} \in t \in \mathfrak{T}_S$ implies $C^{(0)} \in t (= t_{\mathsf{Cl}}(a^{\mathcal{I}_0}))$;
2. for all $C(a)/D(b) \in \mathsf{post}_i$, $1 \leq i \leq n$: if there is a $t \in \mathfrak{T}_S$ with $\{\{a\}, C^{(i-1)}\} \subseteq t$ then $a^{\mathcal{I}_{i-1}} \in C^{\mathcal{I}_{i-1}}$. Thus, $\mathcal{I}_{i-1} \models C(a)$. Since $\mathcal{I}_{i-1} \Rightarrow^{\mathcal{T}}_{\alpha_i} \mathcal{I}_i$, we know that $\mathcal{I}_i \models D(b)$. Hence, $b^{\mathcal{I}_i} \in D^{\mathcal{I}_i}$. Let $t' := t_{\mathsf{Cl}}(b^{\mathcal{I}_i})$. $b^{\mathcal{I}_i} \in D^{\mathcal{I}_i}$ implies $\{\{b\}, D^{(i)}\} \subseteq t'$.

Claim 3. For every $t \in \mathfrak{T}$, t is good in $\mathfrak{T}$.

Proof: Let $t = t_{\mathsf{Cl}}(x)$, for an $x \in \Delta$.

(i) for all concept names A and $i < n$, $\{A^{(i)}, \neg A^{(i+1)}\} \subseteq t_{\mathsf{Cl}}(x)$ holds iff $x \in A^{\mathcal{I}_i} \setminus A^{(i+1)}$, which is, by the semantics of actions, equivalent to $x \in (\mathsf{occ}_{i+1}(A))^{\mathcal{I}_i}$. It is not difficult to show that this holds iff $t, \mathfrak{T} \models \mathsf{occ}_{i+1}(A)$. The case $\{\neg A^{(i)}, A^{(i+1)}\} \subseteq t_{\mathsf{Cl}}(x)$ is similar.

(ii) Let r be a role and let $(\exists r.C)^{(i)} \in t$. Thus $x \in (\exists r.C)^{\mathcal{I}_i}$ and there is a $y \in \Delta$, such that $(x, y) \in r^{\mathcal{I}_i}$ and $y \in C^{\mathcal{I}_i}$. We define $t' := t_{\mathsf{Cl}}(y)$ and $\rho := \{i \mid (x, y) \in r^{\mathcal{I}_i}\}$. It is not difficult to check that t' and ρ satisfy Conditions (R1) to (R5) from Definition 7.

This finishes the proof of Claim 3. Let $\mathfrak{T}_0 := \mathfrak{T}_S \cup \mathfrak{T}_{\mathsf{ano}}$ and let $\mathfrak{T}'$ be the set of types which "survive" type elimination when it is started with $\mathfrak{T}_0$. By Claim 3 and since a type t being good in a type set $\mathfrak{T}$ implies that t is good in any set $\mathfrak{T}' \supseteq \mathfrak{T}$, we have that $\mathfrak{T}_S \subseteq \mathfrak{T} \subseteq \mathfrak{T}'$. Moreover, since $\mathcal{I}_n \models \neg C_0(a_0)$, we have that $\neg C_0^{(n)} \in t_{\mathsf{Cl}}(a_0^{\mathcal{I}_n})$, and thus $\{\{a_0\}, \neg C_0^{(n)}\} \subseteq t_{\mathsf{Cl}}(a_0^{\mathcal{I}_n}) \in \mathfrak{T}'$.

"$\Leftarrow$". For this direction, we also show the contrapositive. Assume that $\mathcal{ALCIO}$-$\mathsf{elim}(\mathcal{A}_0, \mathcal{T}, \alpha_1, \ldots, \alpha_n, \varphi_0)$ returns false. We show that φ_0 is not a consequence of applying $\alpha_1, \ldots, \alpha_n$ in $\mathcal{A}_0$ w.r.t. $\mathcal{T}$. To this end, we construct $\mathcal{I}_0, \ldots, \mathcal{I}_n$ such that $\mathcal{I}_0 \models \mathcal{A}_0$, $\mathcal{I}_i \Rightarrow^{\mathcal{T}}_{\alpha_{i+1}} \mathcal{I}_{i+1}$ for all $i \leq n$, and $\mathcal{I}_n \not\models C_0(a_0)$.

Since the algorithm returns false, there exists a proper core type set $\mathfrak{T}_S$ and a type set $\mathfrak{T}$ such that:

- $\mathfrak{T}_S \subseteq \mathfrak{T}$ and $\mathfrak{T} \setminus \mathfrak{T}_S \subseteq \mathfrak{T}_{\mathsf{ano}}$
- for every $t \in \mathfrak{T}$, t is good in $\mathfrak{T}$
- there is a $t_0 \in \mathfrak{T}$ with $\{\{a_0\}, \neg C_0^{(n)}\} \subseteq t_0$

The types from $\mathfrak{T}$ will be the elements of the domains of $\mathcal{I}_i$. Let $t \in \mathfrak{T}$. Since t is good in $\mathfrak{T}$, for every $D = (\exists r.C)^{(i)} \in t$ we can choose a type t' and a set of indices ρ which satisfy Conditions (R1) to (R5) of Definition 7. Then, we call the chosen t' a (ρ, r)-successor of t. For $t, t' \in \mathfrak{T}$ and r a role, set

$$\mathcal{R}(r, t, t') := \bigcup \{\rho \mid t' \text{ is a } (\rho, r)\text{-successor of } t\}$$

Since $\mathfrak{T}_S$ is a core type set and $\mathfrak{T} \setminus \mathfrak{T}_S \subseteq \mathfrak{T}_{\mathsf{ano}}$, for every $a \in \mathsf{Ind}$ there is exactly one type $t \in \mathfrak{T}$ such that $\{a\} \in t$. For every $a \in \mathsf{Ind}$, we denote this type with t_a. Now we can define $\mathcal{I}_0, \ldots, \mathcal{I}_n$ as follows: for all $i \leq n$,

$$\begin{array}{rcl}
\Delta^{\mathcal{I}_i} & := & \mathfrak{T} \\
A^{\mathcal{I}_i} & := & \{t \in \mathfrak{T} \mid A^{(i)} \in t\} \\
r^{\mathcal{I}_i} & := & \{(t, t') \in \mathfrak{T} \times \mathfrak{T} \mid i \in \mathcal{R}(r, t, t') \cup \mathcal{R}(r^-, t', t)\} \\
a^{\mathcal{I}_i} & := & t_a
\end{array}$$

Claim 4: For all $t \in \mathfrak{T}$ and $C^{(i)} \in \mathsf{Cl}$, we have $t \in C^{\mathcal{I}_i}$ iff $C^{(i)} \in t$.

Proof. We prove the claim by induction on the structure of C.

- $C = A$ and $C = \{a\}$: trivial by definition of $\mathcal{I}_0, \ldots, \mathcal{I}_n$.

- $C = \neg D$ and $C = D \sqcap E$: easy by definition of type.

- $C = \exists s.D$, where s is a (possibly inverse) role:

  "only if". $t \in (\exists s.D)^{\mathcal{I}_i}$ implies that there exists a t' such that $(t, t') \in s^{\mathcal{I}_i}$ and $t' \in D^{\mathcal{I}_i}$. By induction, we have that $D^{(i)} \in t'$. Assume that $(\exists s.D)^{(i)} \notin t$. Then $(\neg\exists s.D)^{(i)} \in t$. By definition of $\mathcal{I}_i$, $(t, t') \in s^{\mathcal{I}_i}$ implies that one of the following cases applies:

  (i) $i \in \mathcal{R}(s, t, t')$. Then $(\neg\exists s.D)^{(i)} \in t$ and Condition (R2) of Definition 7, imply $\neg D^{(i)} \in t'$, which is a contradiction to t' being a type.

  (ii) $i \in \mathcal{R}(\mathsf{Inv}(s), t', t)$. Then $(\neg\exists s.D)^{(i)} \in t$ and Condition (R3) of Definition 7, imply $\neg D^{(i)} \in t'$, which is a contradiction to t' being a type.

  Thus $(\exists s.D)^{(i)} \in t$.

  "if". Let $(\exists s.D)^{(i)} \in t$. Since t is a good type in $\mathfrak{T}$, there exists a type $t' \in \mathfrak{T}$ and a set $\rho \ni i$, such that $\mathcal{R}(s, t, t') \supseteq \rho$ and $D^{(i)} \in t$. By definition of $s^{\mathcal{I}_i}$, we have that $(t, t') \in s^{\mathcal{I}_i}$. Moreover, since $D^{(i)} \in t'$, by induction we have that $t' \in D^{\mathcal{I}_i}$. Thus, it holds that $t \in (\exists s.D)^{\mathcal{I}_i}$.

Using Claim 4, the next claim is easily shown.

Claim 5. Let $t, t' \in \mathfrak{T}$. For all $0 \leq i < n$ the following holds:

(i) for all concept literals B from the input, $t, \mathfrak{T} \models \mathsf{occ}_{i+1}(B)$ implies $t \in (\mathsf{occ}_{i+1}(B))^{\mathcal{I}_i}$.

(ii) for all role literals s from the input, $t, t', \mathfrak{T} \models \mathsf{occ}_{i+1}(s)$ implies $(t, t') \in (\mathsf{occ}_{i+1}(s))^{\mathcal{I}_i}$.

It use Claims 4 and 5 to show that $\mathcal{I}_0 \models \mathcal{A}_0$, $\mathcal{I}_i \Rightarrow^{\mathcal{T}}_{\alpha_i} \mathcal{I}_{i+1}$ for all $i \leq n$, and that $\mathcal{I}_0 \models \neg C_0(a_0)$.

- $\mathcal{I}_0 \models \mathcal{A}_0$: Let $C(a) \in \mathcal{A}_0$. Since $\mathfrak{T}_S$ is proper, $C^{(0)} \in t_a$ and, by Claim 4, we have that $a^{\mathcal{I}_0} = t_a \in C^{\mathcal{I}_0}$.

- $\mathcal{I}_i \Rightarrow^{\mathcal{T}}_{\alpha_i} \mathcal{I}_{i+1}$ for all $i \leq n$:

  - $\mathcal{I}_i$ are models of $\mathcal{T}$: Let $C \sqsubseteq D \in \mathcal{T}$ and $t \in C^{\mathcal{I}_i}$. By Claim 4, we have $C^{(i)} \in t$, and since t is a type for Cl, $C^{(i)} \in t$ implies $D^{(i)} \in t$. Hence, $t \in D^{\mathcal{I}_i}$.

– Let $C(a)/D(b) \in \mathsf{post}_i$ and let $\mathcal{I}_i \models C(a)$. This means that $t_a = a^{\mathcal{I}_i} \in C^{\mathcal{I}_i}$, and thus $\{\{a\}, C^{(i)}\} \subseteq t_a$. Since $\mathfrak{T}_S$ is proper, there exists a $t \in \mathfrak{T}_S$, such that $\{\{b\}, D^{(i+1)}\} \subseteq t$. Obviously, $t = t_b$, and thus $t_b = b^{\mathcal{I}_i} \in D^{\mathcal{I}_{i+1}}$, i.e. $\mathcal{I}_{i+1} \models D(b)$.

– for each $A \in \mathsf{N_C}$ and $i < n$, we have: $A^{\mathcal{I}_i} \setminus A^{\mathcal{I}_{i+1}} \subseteq (\mathsf{occ}_{i+1}(A))^{\mathcal{I}_i}$ and $A^{\mathcal{I}_{i+1}} \setminus A^{\mathcal{I}_i} \subseteq (\mathsf{occ}_{i+1}(\neg A))^{\mathcal{I}_i}$. We show only the former since the latter can be shown in a similar way. Let $t \in A^{\mathcal{I}_i}$ and $t \notin A^{\mathcal{I}_{i+1}}$. By Claim 4, this implies $A^{(i)} \in t$ and $(\neg A)^{(i+1)} \in t$. Since t is good in $\mathfrak{T}$, we have that $t, \mathfrak{T} \models \mathsf{occ}_{i+1}(A)$, and by Claim 5 we obtain that $t \in (\mathsf{occ}_{i+1}(A))^{\mathcal{I}_i}$.

– for each $r \in \mathsf{N_R}$ and $i < n$, we have: $r^{\mathcal{I}_i} \setminus r^{\mathcal{I}_{i+1}} \subseteq (\mathsf{occ}_{i+1}(r))^{\mathcal{I}_i}$ and $r^{\mathcal{I}_{i+1}} \setminus r^{\mathcal{I}_i} \subseteq (\mathsf{occ}_{i+1}(\neg r))^{\mathcal{I}_i}$. Again,we show only the former as the latter can be shown in a similar way. Let $(t, t') \in r^{\mathcal{I}_i}$ and $(t, t') \notin r^{\mathcal{I}_{i+1}}$. By the definition of $r^{\mathcal{I}_i}$, $i \in \mathcal{R}(r, t, t') \cup \mathcal{R}(r^-, t', t)$ and $i + 1 \notin \mathcal{R}(r, t, t') \cup \mathcal{R}(r^-, t', t)$.

  (i) Let $i \in \mathcal{R}(r, t, t')$. By definition of $\mathcal{R}$, t' is a (ρ, r)-successor of t for a $\rho \ni i$. Since $i + 1 \notin \mathcal{R}(r, t, t')$, we obtain that $i + 1 \notin \rho$. Since t is good in $\mathfrak{T}$, by Condition (R4) of Definition 7, we have that $t, t', \mathfrak{T} \models \mathsf{occ}_{i+1}(r)$. By Claim 5, we obtain that $(t, t') \in (\mathsf{occ}_{i+1}(r))^{\mathcal{I}_i}$.

  (ii) Let $i \in \mathcal{R}(r^-, t', t)$. Since $i + 1 \notin \mathcal{R}(r^-, t', t)$ and t' is good in $\mathfrak{T}$, similarly as in (i), by Condition (R4) of Definition 7, we have that $t', t, \mathfrak{T} \models \mathsf{occ}_{i+1}(r^-)$, which is equivalent to $t, t', \mathfrak{T} \models \mathsf{occ}_{i+1}(r)$. By Claim 5, we obtain that $(t, t') \in (\mathsf{occ}_{i+1}(r))^{\mathcal{I}_i}$.

- $\mathcal{I}_n \models \neg C_0(a_0)$: Since $(C_0)^{(n)} \in t_{a_0}$, by Claim 4, it holds that $t_{a_0} = a_0^{\mathcal{I}_n} \in (C_0)^{\mathcal{I}_n}$

❑

The algorithm runs in exponential time: first, we have already argued that there are only exponentially many core type sets. Second, the number of elimination rounds is bounded by the number of types, of which there are only exponentially many. And third, it is easily seen that it can be checked in exponential time whether a type is good in a given type set. Since concept satisfiability w.r.t. TBoxes is ExpTime-hard in $\mathcal{ALC}$ [3] and concept satisfiability can be reduced to (non-)projection [2], we obtain the following result.

Theorem 9. *Projection, executability and weak consistency are* ExpTime-*complete in $\mathcal{ALC}$, $\mathcal{ALCO}$, $\mathcal{ALCI}$, and $\mathcal{ALCIO}$.*

It is not too difficult to adapt the algorithm given in this section to the DL $\mathcal{ALCQO}$. Therefore, the reasoning problems from Theorem 9 are also ExpTime-complete for $\mathcal{ALCQ}$ and $\mathcal{ALCQO}$.

5 $\mathcal{ALCQI}$ and $\mathcal{ALCQIO}$: Beyond ExpTime

In the previous section, we have identified a number of DLs for which both reasoning about actions and standard DL reasoning are ExpTime-complete. Another candidate for a DL with such a behaviour is $\mathcal{ALCQI}$, in which satisfiability and subsumption are ExpTime-complete as well [18]. However, it follows from results in [2] that projection in $\mathcal{ALCQI}$ is co-NExpTime-hard. In the following, we show that it is in fact co-NExpTime-complete, and that the same holds for the DL $\mathcal{ALCQIO}$. Note that, for the latter DL, also concept subsumption is co-NExpTime-complete.

Since the action formalism defined in this paper is a generalization of the one from [2] (see Appendix A), Lemma 8 of [2] implies the following.

Theorem 10. *Projection and executability (weak consistency) in $\mathcal{ALCQI}$ are co-*NExpTime*-hard (*NExpTime*-hard) even if occlusions for roles are disallowed and only nominals are allowed in the occlusions of concept names.*

In the following, we establish a matching co-NExpTime upper bound for projection in $\mathcal{ALCQIO}$ (and thus also $\mathcal{ALCQI}$). The proof proceeds by reducing projection in $\mathcal{ALCQIO}$ to ABox (in)consistency in $\mathcal{ALCQIO}^{\neg,\cap,\cup}$, i.e., the extension of $\mathcal{ALCQIO}$ with the Boolean role constructors $\neg$, $\cap$, and $\cup$ with the following semantics:

$$
\begin{aligned}
(\neg r)^{\mathcal{I}} &:= (\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}) \setminus r^{\mathcal{I}} \\
(r \cap s)^{\mathcal{I}} &:= r^{\mathcal{I}} \cap s^{\mathcal{I}} \\
(r \cup s)^{\mathcal{I}} &:= r^{\mathcal{I}} \cup s^{\mathcal{I}}
\end{aligned}
$$

Let $\alpha_1, \ldots, \alpha_n$ be a composite action with $\alpha_i = (\mathsf{pre}_i, \mathsf{occ}_i, \mathsf{post}_i)$ for $i = 1, \ldots, n$, and let $\mathcal{T}$ be a TBox, $\mathcal{A}_0$ an ABox and φ_0 an assertion. We are interested in deciding whether φ_0 is a consequence of applying $\alpha_1, \ldots, \alpha_n$ in $\mathcal{A}_0$ w.r.t. $\mathcal{T}$. In what follows, we call $\alpha_1, \ldots, \alpha_n$, $\mathcal{T}$, $\mathcal{A}_0$ and φ_0 the *input*. W.l.o.g., we make the following assumptions:

- φ_0 is of the form $\varphi_0 = C_0(a_0)$, where C_0 is a (possibly complex) concept.

  As in the previous section, this assumption can be made because an assertion $r(a, b)$ can be replaced with $(\exists r.\{b\})(a)$, and $\neg r(a, b)$ with $(\neg\exists r.\{b\})(a)$.

- Each occlusion pattern occ_i contains exactly one occlusion pattern that is unconditional (i.e., indexed by taut) and formulated in $\mathcal{ALCQIO}^{\neg,\cap,\cup}$.

  An occlusion pattern $\{\mathsf{occ}_{\varphi_1}, \ldots, \mathsf{occ}_{\varphi_n}\}$ can be converted into an occlusion pattern $\{\mathsf{occ}_{\mathsf{taut}}\}$ formulated in $\mathcal{ALCQIO}^{\neg,\cap,\cup}$ as follows. First, we may assume w.l.o.g. that φ_i is of the form $C_i(a_i)$ for $1 \leq i \leq n$ (see previous point). For $1 \leq i \leq n$, let P_i denote the concept $\forall U.(\{a_i\} \rightarrow C_i)$, where

U denotes the universal role, i.e. $r \cup \neg r$ for some $r \in \mathsf{N_R}$. Then, define for each concept literal A

$$\mathsf{occ_{taut}}(A) := \bigsqcup_{1 \le i \le n} (P_i \sqcap \mathsf{occ}_{\varphi_i}(A))$$

Likewise, for each role literal r, define

$$\mathsf{occ_{taut}}(r) := \{(P_i \sqcap C, P_i \sqcap D) \mid (C, D) \in \mathsf{occ}_{\varphi_i}\}.$$

Having the occlusion pattern formulated in $\mathcal{ALCQIO}^{\neg,\cap,\cup}$ is unproblematic since our reduction is to $\mathcal{ALCQIO}^{\neg,\cap,\cup}$ anyway. In the following, we slightly abuse notation and confuse the singleton set occ_i with the (unconditional) occlusion mapping contained in it.

The idea of the reduction is to define an ABox $\mathcal{A}_{\mathsf{red}}$ and a TBox $\mathcal{T}_{\mathsf{red}}$ such that a single model of them encodes a sequence of interpretations $\mathcal{I}_0, \ldots, \mathcal{I}_n$ such that $\mathcal{I}_0 \models \mathcal{A}_0, \mathcal{T}$ and $\mathcal{I}_{i-1} \Rightarrow^{\mathcal{T}}_{\alpha_i} \mathcal{I}_i$ for $i = 1, \ldots, n$. As in the previous section, we use Sub to denote the set of subconcepts of the concepts which occur in the input and introduce concept names $A^{(i)}$ and role names $r^{(i)}$ for every concept name A and every role name r used in the input, for all $i \le n$. For a complex concept $C \in \mathsf{Sub}$, we use $C^{(i)}$, for $i \le n$, to denote the concept obtained by replacing all concept names A and role names r occurring in C by $A^{(i)}$ and $r^{(i)}$ respectively.

We start by assembling the reduction ABox $\mathcal{A}_{\mathsf{red}}$. First, define a "copy" $\mathcal{A}_{\mathsf{ini}}$ of the input ABox $\mathcal{A}_0$ as:

$$\begin{aligned}\mathcal{A}_{\mathsf{ini}} := \; & \{C^{(0)}(a) \mid C(a) \in \mathcal{A}_0\} \cup \\ & \{r^{(0)}(a,b) \mid r(a,b) \in \mathcal{A}_0\} \cup \{\neg r^{(0)}(a,b) \mid \neg r(a,b) \in \mathcal{A}_0\}\end{aligned}$$

Then, introduce abbreviations, for $i \le n$:

$$\begin{aligned}\mathsf{p}_i(C(a)) &:= \forall U.(\{a\} \to C^{(i)}),\\ \mathsf{p}_i(r(a,b)) &:= \forall U.(\{a\} \to \exists r^{(i)}.\{b\}),\\ \mathsf{p}_i(\neg r(a,b)) &:= \forall U.(\{a\} \to .\forall r^{(i)}.\neg\{b\}),\end{aligned}$$

Now we can define the components of $\mathcal{A}_{\mathsf{red}}$ that take care of post-condition satisfaction. For $1 \le i \le n$, we define:

$$\mathcal{A}^{(i)}_{\mathsf{post}} := \{\big(\mathsf{p}_{i-1}(\varphi) \to \mathsf{p}_i(\psi)\big)(a_0) \mid \varphi/\psi \in \mathsf{post}_i\}$$

We assemble $\mathcal{A}_{\mathsf{red}}$ as

$$\mathcal{A}_{\mathsf{red}} := \mathcal{A}_{\mathsf{ini}} \cup \bigcup_{1 \le i \le n} \mathcal{A}^{(i)}_{\mathsf{post}}.$$

Next, we define the components of the TBox $\mathcal{T}_{\mathsf{red}}$. Since all interpretations $\mathcal{I}_0, \ldots, \mathcal{I}_n$ have to be models of the input TBox $\mathcal{T}$, we define for each $i \le n$, a copy $\mathcal{T}^{(i)}$ of $\mathcal{T}$ in the obvious way:

$$\mathcal{T}^{(i)} = \{C^{(i)} \sqsubseteq D^{(i)} \mid C \sqsubseteq D \in \mathcal{T}\}.$$

To deal with occlusions, we introduce auxiliary role names $r^{(i)}_{\mathsf{Dom}(C)}$ and $r^{(i)}_{\mathsf{Ran}(D)}$ for $0 \leq i < n$ and all concepts C, D such that $(C, D) \in \mathsf{occ}_i(s)$ for some role literal s. The following TBox $\mathcal{T}^{(i)}_{\mathsf{aux}}$ ensures that $r^{(i)}_{\mathsf{Dom}(C)}$ and $r^{(i)}_{\mathsf{Ran}(D)}$ are interpreted as $C^{(i)} \times \top$ and $\top \times D^{(i)}$, respectively. It contains the following axioms, for all concepts C, D as above:

$$
\begin{array}{rclcrcl}
C^{(i)} & \sqsubseteq & \forall \neg r^{(i)}_{\mathsf{Dom}(C)}.\bot & \qquad & \top & \sqsubseteq & \forall r^{(i)}_{\mathsf{Ran}(D)}.D^{(i)} \\
\neg C^{(i)} & \sqsubseteq & \forall r^{(i)}_{\mathsf{Dom}(C)}.\bot & \qquad & \top & \sqsubseteq & \forall \neg r^{(i)}_{\mathsf{Ran}(D)}.\neg D^{(i)}
\end{array}
$$

The following TBox $\mathcal{T}^{(i)}_{\mathsf{fix}}$ ensures that concept and role names do not change unless this is allowed by the occlusion pattern:

- for every concept name A in the input,

$$
\begin{array}{rcl}
A^{(i)} \sqcap \neg A^{(i+1)} & \sqsubseteq & (\mathsf{occ}_{i+1}(A))^{(i)} \\
\neg A^{(i)} \sqcap A^{(i+1)} & \sqsubseteq & (\mathsf{occ}_{i+1}(\neg A))^{(i)}
\end{array}
$$

- for every role name r in the input,

$$
\begin{array}{rcl}
\top & \sqsubseteq & \forall \neg \Big(\bigcup\limits_{(C,D) \in \mathsf{occ}_{i+1}(r)} (r^{(i)}_{\mathsf{Dom}(C)} \cap r^{(i)}_{\mathsf{Ran}(D)}) \Big) \cap (r^{(i)} \cap \neg r^{(i+1)}).\bot \\
\top & \sqsubseteq & \forall \neg \Big(\bigcup\limits_{(C,D) \in \mathsf{occ}_{i+1}(\neg r)} (r^{(i)}_{\mathsf{Dom}(C)} \cap r^{(i)}_{\mathsf{Ran}(D)}) \Big) \cap (\neg r^{(i)} \cap r^{(i+1)}).\bot
\end{array}
$$

Finally, we can construct $\mathcal{T}_{\mathsf{red}}$ as

$$
\mathcal{T}_{\mathsf{red}} := \bigcup_{0 \leq i \leq n} \mathcal{T}^{(i)} \cup \bigcup_{0 \leq i < n} \mathcal{T}^{(i)}_{\mathsf{aux}} \cup \bigcup_{0 \leq i < n} \mathcal{T}^{(i)}_{\mathsf{fix}}.
$$

Then the following lemma holds:

Lemma 11. *$C_0(a_0)$ is a consequence of applying $\alpha_1, \ldots, \alpha_n$ in $\mathcal{A}_0$ w.r.t. $\mathcal{T}$ iff $\mathcal{A}_{\mathsf{red}} \cup \{\neg C^{(n)}_0(a_0)\}$ is inconsistent w.r.t. $\mathcal{T}_{\mathsf{red}}$.*

Proof. "$\Rightarrow$": We prove this direction by contraposition. Assume that $\mathcal{A}_{\mathsf{red}} \cup \{\neg C^{(n)}_0(a_0)\}$ is not inconsistent w.r.t. $\mathcal{T}_{\mathsf{red}}$. Thus, there exists a $\mathcal{J}$ such that $\mathcal{J} \models \mathcal{T}_{\mathsf{red}}$, $\mathcal{J} \models \mathcal{A}_{\mathsf{red}}$, and $\mathcal{J} \models \neg C^{(n)}_0(a_0)$. In order to prove that $C_0(a_0)$ is not a consequence of applying $\alpha_1, \ldots, \alpha_n$ in $\mathcal{A}_0$ w.r.t. $\mathcal{T}$ we show that there are $\mathcal{I}_0, \ldots, \mathcal{I}_n$ such that $\mathcal{I}_{i-1} \Rightarrow^{\mathcal{T}}_{\alpha_i} \mathcal{I}_i$ for all $1 \leq i \leq n$, $\mathcal{I}_0 \models \mathcal{A}_0$, and $\mathcal{I}_n \not\models C_0(a_0)$.

We construct $\mathcal{I}_i$ for $i \leq n$ as following:

- $\Delta^{\mathcal{I}_i} := \Delta^{\mathcal{J}}$;

- $A^{\mathcal{I}_i} := (A^{(i)})^{\mathcal{J}}$ for every concept name A;
- $r^{\mathcal{I}_i} := (r^{(i)})^{\mathcal{J}}$ for every role name r;
- $a^{\mathcal{I}_i} := a^{\mathcal{J}}$ for every individual name a.

By definition of $C^{(i)}$, it is obvious that for all $x \in \Delta^{\mathcal{J}}$, $C \in \mathsf{Sub}$ and $i \leq n$:

$$x \in C^{\mathcal{I}_i} \text{ iff } x \in (C^{(i)})^{\mathcal{J}} \tag{$*$}$$

We have that $\mathcal{I}_{i-1} \Rightarrow^{\mathcal{T}}_{\alpha_i} \mathcal{I}_i$ for all $1 \leq i \leq n$ since

- $\mathcal{I}_i \models \mathcal{T}$: for all $C \sqsubseteq D \in \mathcal{T}$, we have $C^{(i)} \sqsubseteq D^{(i)} \in \mathcal{T}_{\mathsf{red}}$. $\mathcal{J} \models \mathcal{T}_{\mathsf{red}}$ implies $\mathcal{J} \models C^{(i)} \sqsubseteq D^{(i)}$. For all $x \in \Delta^{\mathcal{I}_i} = \Delta^{\mathcal{J}}$, we have $x \in (C^{(i)})^{\mathcal{J}}$ implies $x \in (D^{(i)})^{\mathcal{J}}$. By ($*$), this yields that $x \in C^{\mathcal{I}_i}$ implies $x \in D^{\mathcal{I}_i}$.

- $\mathcal{I}_{i-1}, \mathcal{I}_i \models \mathsf{post}_i$: It follows from the definition of p_i that for all ABox assertions φ and for all $i \leq n$, we have $(\mathsf{p}_i(\varphi))^{\mathcal{J}} = \Delta^{\mathcal{J}}$ if $\mathcal{I}_i \models \varphi$ and $(\mathsf{p}_i(\varphi))^{\mathcal{J}} = \emptyset$ otherwise. For all $\varphi/\psi \in \mathsf{post}_i$, we have $(\mathsf{p}_{i-1}(\varphi) \to \mathsf{p}_i(\psi))(a_0) \in \mathcal{A}_{\mathsf{red}}$. Assume $\mathcal{I}_{i-1} \models \varphi$. Then $(\mathsf{p}_{i-1}(\varphi))^{\mathcal{J}} = \Delta^{\mathcal{I}_{i-1}}$ by ($*$). Thus, $\mathcal{J} \models \mathcal{A}_{\mathsf{red}}$ yields $(\mathsf{p}_{i-1}(\psi))^{\mathcal{J}} = \Delta^{\mathcal{J}}$, which implies $\mathcal{I}_i \models \psi$ by ($*$).

- $A^{\mathcal{I}_{i-1}} \setminus A^{\mathcal{I}_i} \subseteq (\mathsf{occ}_i(A))^{\mathcal{I}_{i-1}}$ and $A^{\mathcal{I}_i} \setminus A^{\mathcal{I}_{i-1}} \subseteq (\mathsf{occ}_i(\neg A))^{\mathcal{I}_{i-1}}$ follow from $\mathcal{J} \models \mathcal{T}^{(i)}_{\mathsf{fix}}$.

- $r^{\mathcal{I}_{i-1}} \setminus r^{\mathcal{I}_i} \subseteq (\mathsf{occ}_i(r))^{\mathcal{I}_{i-1}}$: Let $(x, y) \in r^{\mathcal{I}_{i-1}} \setminus r^{\mathcal{I}_i}$. By the construction of $\mathcal{I}_{i-1}$ and $\mathcal{I}_i$, we have $(x, y) \in (r^{(i-1)} \cap \neg r^{(i)})^{\mathcal{J}}$. Then $\mathcal{J} \models \mathcal{T}^{(i)}_{\mathsf{fix}}$ implies

  $$(x, y) \in \Big(\bigcup_{(C,D) \in \mathsf{occ}_i(r)} (r^{(i-1)}_{\mathsf{Dom}(C)} \cap r^{(i-1)}_{\mathsf{Ran}(D)}) \Big)^{\mathcal{J}}.$$

  Hence, there exists a pair $(C, D) \in \mathsf{occ}_i(r)$ such that $(x, y) \in (r^{(i-1)}_{\mathsf{Dom}(C)} \cap r^{(i-1)}_{\mathsf{Ran}(D)})^{\mathcal{J}}$. Moreover, $\mathcal{J} \models \mathcal{T}^{(i)}_{\mathsf{aux}}$ implies $x \in (C^{(i-1)})^{\mathcal{J}}$ and $y \in (D^{(i-1)})^{\mathcal{J}}$.

  Thus, by ($*$) we have $x \in C^{\mathcal{I}_{i-1}}$ and $y \in D^{\mathcal{I}_{i-1}}$ which implies $(x, y) \in (\mathsf{occ}_i(r))^{\mathcal{I}_{i-1}}$. Analogously, it can be shown that $r^{\mathcal{I}_i} \setminus r^{\mathcal{I}_{i-1}} \subseteq (\mathsf{occ}_i(\neg r))^{\mathcal{I}_{i-1}}$ holds.

$\mathcal{I}_0 \models \mathcal{A}_0$: for all concept assertions $C(a) \in \mathcal{A}$, we have $C^{(0)}(a) \in \mathcal{A}_{\mathsf{red}}$. $\mathcal{J} \models \mathcal{A}_{\mathsf{red}}$ implies $a^{\mathcal{J}} \in (C^{(0)})^{\mathcal{J}}$. Then, by ($*$) we know $a^{\mathcal{I}_0} \in C^{\mathcal{I}_0}$. We can prove the same result for all role assertions in $\mathcal{A}$ from the definition of $r^{\mathcal{I}_0}$ and $\mathcal{J} \models \mathcal{A}_{\mathsf{red}}$.

$\mathcal{I}_n \not\models C_0(a_0)$: $\mathcal{J} \models \neg C^{(n)}_0(a_0)$ implies $a_0^{\mathcal{J}} \in (\neg C^{(n)}_0)^{\mathcal{J}}$. Thus, by ($*$) we know $a_0^{\mathcal{J}} = a_0^{\mathcal{I}_n} \in (\neg C_0)^{\mathcal{I}_n}$.

"$\Leftarrow$": This direction can also be proved by contraposition. Assume that $C_0(a_0)$ is not a consequence of applying $\alpha_1, \dots, \alpha_n$ in $\mathcal{A}_0$ w.r.t. $\mathcal{T}$. Thus, there are $\mathcal{I}_0, \dots, \mathcal{I}_n$ such that $\mathcal{I}_{i-1} \Rightarrow^{\mathcal{T}}_{\alpha_i} \mathcal{I}_i$ for all $1 \leq i \leq n$, $\mathcal{I}_0 \models \mathcal{A}_0$, and $\mathcal{I}_n \not\models C_0(a_0)$. We define an interpretation $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ as follows:

- $\Delta^{\mathcal{J}} := \Delta^{\mathcal{I}_0}(= \Delta^{\mathcal{I}_1} = \cdots = \Delta^{\mathcal{I}_n});$
- $(A^{(i)})^{\mathcal{J}} := A^{\mathcal{I}_i}$ for all concept names A and for all $i \leq n$;
- $(r^{(i)})^{\mathcal{J}} := r^{\mathcal{I}_i}$ for all role names r and for all $i \leq n$;
- $a^{\mathcal{J}} := a^{\mathcal{I}_0}(= a^{\mathcal{I}_1} = \cdots = a^{\mathcal{I}_n})$ for all individual names a;
- $(r^{(i)}_{\mathsf{Dom}(C)})^{\mathcal{J}} := \{C^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_i}\}$ and $(r^{(i)}_{\mathsf{Ran}(D)})^{\mathcal{J}} := \{\Delta^{\mathcal{I}_i} \times D^{\mathcal{I}_i}\}$ for all $i < n$.

By definition of $C^{(i)}$, it obvious that for all $x \in \Delta^{\mathcal{J}}$, $C \in \mathsf{Sub}$ and $i \leq n$:

$$x \in C^{\mathcal{I}_i} \text{ iff } x \in (C^{(i)})^{\mathcal{J}}$$

Using this observation and the semantics of actions, it is not difficult to show that indeed $\mathcal{J} \models \mathcal{A}_{\mathsf{red}}$, $\mathcal{J} \models \mathcal{T}_{\mathsf{red}}$, and $\mathcal{J} \models \neg C_0^{(n)}(a_0)$. Thus, $\mathcal{A}_{\mathsf{red}} \cup \{\neg C_0^{(n)}(a_0)\}$ is consistent w.r.t. $\mathcal{T}_{\mathsf{red}}$.

❏

Since $\mathcal{ALCQIO}^{\neg,\cap,\cup}$ is a fragment of $\mathcal{C}^2$ (the 2-variable fragment of first-order logic with counting), ABox inconsistency in $\mathcal{ALCQIO}^{\neg,\cap,\cup}$ is in co-NExpTime even if numbers are coded in binary [11]. Since $\mathcal{A}_{\mathsf{red}}$ and $\mathcal{T}_{\mathsf{red}}$ are polynomial in the size of the input, Lemma 11 gives us a co-NExpTime upper bound for projection in $\mathcal{ALCQIO}$ and $\mathcal{ALCQI}$.

Theorem 12. *Projection and executability are co-*NExpTime*-complete, while weak consistency is* NExpTime*-complete in $\mathcal{ALCQIO}$ and $\mathcal{ALCQI}$.*

6 Reduction to implemented DLs

Together with a reasoner that is capable of deciding ABox consistency in the description logic $\mathcal{ALCQIO}^{\neg,\cap,\cup}$, the reduction developed in the previous section can be used for practical reasoning with $\mathcal{ALCQIO}$ actions. Unfortunately, to the best of our knowledge there exists no available DL reasoner that supports the Boolean operators on roles. In this section, we identify a restricted version of our action formalism in which the reduction of projection to ABox consistency does not need the Boolean role constructors. More precisely, we show that for this restricted fragment, projection in $\mathcal{L}$ can be reduced to ABox (in)consistency in $\mathcal{LO}$, the extension of $\mathcal{L}$ with nominals. The reduction works for $\mathcal{ALC}$ extended with any combination of inverses, nominals, and number restrictions.

Definition 13. A *restriced $\mathcal{L}$-action* α is a triple $(\mathsf{pre}, \mathsf{occ}, \mathsf{post})$ where pre and post are as in Definition 1 and occ assigns

1. to every concept literal B an $\mathcal{LO}$-concept $\mathsf{occ}(B)$;
2. to every role literal r a finite subset of

$$\{(\{a\}, \{b\}) \mid a, b \in \mathsf{N_I}\} \cup \{(\top, \top), (\bot, \bot)\}$$

3. for every role name r, $\mathsf{occ}(r) = \mathsf{occ}(\neg r)$;

$\triangle$

For assigning a semantics of restricted actions, we simply view the single occlusion mapping occ as an occlusion pattern $\{\mathsf{occ}_{\mathsf{taut}}\}$, where taut is a valid assertion such as $\top(a)$.

We now present the reduction of projection for restricted $\mathcal{L}$-actions to ABox consistency in $\mathcal{LO}$. Let $\mathcal{A}_0$ be an ABox, $\alpha_1, \ldots, \alpha_n$ a composite action with all $\alpha_i = (\mathsf{pre}_i, \mathsf{occ}_i, \mathsf{post}_i)$ restricted, $\mathcal{T}$ a TBox, and φ_0 an assertion. We are interested in deciding whether φ_0 is a consequence of applying $\alpha_1, \ldots, \alpha_n$ in $\mathcal{A}_0$ w.r.t. $\mathcal{T}$. Since nominals are needed for the reduction anyway, we may assume w.l.o.g. that φ_0 is of the form $A_0(a_0)$ with A_0 a concept name: (i) as in the previous sections, every role assertion can be replaced by a concept assertion using nominals; (ii) if $\varphi = C(a)$ with C not a concept name, we add a concept definition $A_0 \doteq C$ to the TBox $\mathcal{T}$, and then consider $\varphi = A_0(a)$.

In the following, we call $\mathcal{A}_0$, $\mathcal{T}$, $\alpha_1, \ldots, \alpha_n$, and φ_0 *the input*. As in the previous section, the main idea of the reduction is to define $\mathcal{A}_{\mathsf{red}}$ and $\mathcal{T}_{\mathsf{red}}$ such that each *single* model of them encodes a *sequence* of interpretations $\mathcal{I}_0, \ldots, \mathcal{I}_n$ obtained by applying $\alpha_1, \ldots, \alpha_n$ in $\mathcal{A}_0$ (and *all* such sequences are encoded by reduction models). To define $\mathcal{A}_{\mathsf{red}}$ and $\mathcal{T}_{\mathsf{red}}$ without using the Boolean role operators, we use a reduction technique similar to the one in [**?**].

To understand the reduction, it is important to distinguish two kinds of elements in interpretations: we call an element $d \in \Delta^{\mathcal{I}}$ *named* if $a^{\mathcal{I}} = d$ for some individual a used in the input, and *unnamed* otherwise. Intuitively, the ABox $\mathcal{A}_{\mathsf{red}}$ will ensure that role names do not change on named elements unless allowed by the occlusion pattern while the minimization of changes on unnamed elements is achieved through a suitable encoding of $\mathcal{T}$ in $\mathcal{T}_{\mathsf{red}}$.

In the reduction, we use the following concept names, role names, and individual names:

- We use Sub to denote the set of all subconcepts of concepts appearing in the input. For every $C \in \mathsf{Sub}$ and every $i \leq n$, we introduce a concept name $T_C^{(i)}$. It will be ensured by the TBox $\mathcal{T}_{\mathsf{red}}$ that the concept name $T_C^{(i)}$ stands for the interpretation of C in the i-th interpretation.

- We use a concept name $A^{(i)}$ for every concept name A used in the input and every $i \leq n$. Intuitively, $A^{(i)}$ represents the interpretation of the concept name A in the i-th interpretation.
- We use a role name $r^{(i)}$ for every role name r used in the input and every $i \leq n$. Intuitively, the extension of r in the i-th interpretation can be assembled from the extensions of $r^{(0)}, \ldots, r^{(n)}$ as follows:
  - regarding pairs (x, y) with both x, y named, we consider $r^{(i)}$;
  - regarding pairs (x, y) with one of x, y not named, we consider $r^{(j)}$, where $j \in \{1, \ldots, i\}$ is maximal such that $\mathsf{occ}_j(r)$ contains the global occlusion $(\top, \top)$ (and $j = 0$ if there is no such j).

  Thus, to check the membership in $r^{\mathcal{I}_i}$ of pairs (x, y) with x or y unnamed, we "go back" to the last interpretation before which (x, y) was occluded.
- We use a concept name N that will be used to denote the named elements of interpretations.
- We use Ind to denote the set of individual names in the input. For every $a \in \mathsf{Ind}$, we introduce an auxiliary role name r_a.

The reduction TBox $\mathcal{T}_{\mathsf{red}}$ consists of several components. The first component simply states that N denotes exactly the named domain elements:

$$\mathcal{T}_N := \Big\{ N \doteq \bigsqcup_{a \in \mathsf{Ind}} \{a\} \Big\}.$$

The sequence of components $\mathcal{T}_{\mathsf{fix}}^{(i)}$, $1 \leq i \leq n$, ensures that concept names do not change unless this is allowed by the occlusion pattern: for every concept name A in the input, $\mathcal{T}_{\mathsf{fix}}^{(i)}$ contains:

$$\begin{array}{rcl} A^{(i)} \sqcap \neg A^{(i+1)} & \sqsubseteq & (\mathsf{occ}_{i+1}(A))^{(i)} \\ \neg A^{(i)} \sqcap A^{(i+1)} & \sqsubseteq & (\mathsf{occ}_{i+1}(\neg A))^{(i)} \end{array}$$

In order to ensure the same for role names, we will let the ABox $\mathcal{A}_{\mathsf{red}}$ (see the component $\mathcal{A}_{\mathsf{fix}}^{(i)}$ below) tackle occlusions on the named part (i.e. $N \times N$). For the remaining part of the domain, we look back in order to determine when r was last occluded. We set $\mathsf{occ}_\ell(r^-) := \{(Y, X) \mid (X, Y) \in \mathsf{occ}_\ell(r)\}$. For every role r from the input, and $i \leq n$, we define

$$\mathsf{lo}_r^i \;\; := \;\; \max\{\, \ell \;\mid\; 1 \leq \ell \leq i \;\wedge\; (\top, \top) \in \mathsf{occ}_\ell(r) \,\}$$

where $\max(\emptyset) := 0$. The next component $\mathcal{T}_{\mathsf{sub}}$ contains one concept definition for every $i \leq n$ and every concept $C \in \mathsf{Sub}$ that is not a defined concept name in $\mathcal{T}$. These concept definitions ensure that $T_C^{(i)}$ stands for the interpretation of C in the i-th interpretation as desired:

$$
\begin{aligned}
T_A^{(i)} &\doteq A^{(i)} \qquad\qquad \text{if } A \text{ is a concept name}\\
T_{\neg C}^{(i)} &\doteq \neg T_C^{(i)}\\
T_{C \sqcap D}^{(i)} &\doteq T_C^{(i)} \sqcap T_D^{(i)}\\
T_{C \sqcup D}^{(i)} &\doteq T_C^{(i)} \sqcup T_D^{(i)}\\
T_{(\geqslant m\ r\ C)}^{(i)} &\doteq \Big(N \sqcap \bigsqcup_{0 \leq j \leq m} \big((\geqslant j\ r^{(i)}\ (N \sqcap T_C^{(i)})) \sqcap (\geqslant (m-j)\ r^{(\mathsf{ol}_r^i)}\ (\neg N \sqcap T_C^{(i)}))\big)\Big) \sqcup\\
&\quad \big(\neg N \sqcap (\geqslant m\ r^{(\mathsf{ol}_r^i)}\ T_C^{(i)})\big)\\
T_{(\leqslant m\ r\ C)}^{(i)} &\doteq \Big(N \sqcap \bigsqcup_{0 \leq j \leq m} \big((\leqslant j\ r^{(i)}\ (N \sqcap T_C^{(i)})) \sqcap (\leqslant (m-j)\ r^{(\mathsf{ol}_r^i)}\ (\neg N \sqcap T_C^{(i)}))\big)\Big) \sqcup\\
&\quad \big(\neg N \sqcap (\leqslant j\ r^{(\mathsf{lo}_r^i)}\ T_C^{(i)})\big)
\end{aligned}
$$

where $r^{-(i)}$ denotes $(r^{(i)})^-$ in the concept definitions for number restrictions. Now we can assemble the reduction TBox $\mathcal{T}_{\mathsf{red}}$:

$$\mathcal{T}_{\mathsf{red}} := \mathcal{T}_{\mathsf{sub}} \cup \mathcal{T}_N \ \cup \bigcup_{1 \leq i \leq n} \mathcal{T}_{\mathsf{fix}}^{(i)} \ \cup \{T_C^{(i)} \sqsubseteq T_D^{(i)} \mid C \sqsubseteq D \in \mathcal{T},\ i \leq n\}$$

The last summand of $\mathcal{T}_{\mathsf{red}}$ ensures that all GCIs from the input TBox $\mathcal{T}$ are satisfied by all interpretations $\mathcal{I}_0, \ldots, \mathcal{I}_n$.

The reduction ABox $\mathcal{A}_{\mathsf{red}}$ also consists of several components. The first component ensures that, for each individual name a occurring in the input, the auxiliary role r_a connects each named individual with a, and only with a. This construction will simplify the definition of the other components of $\mathcal{A}_{\mathsf{red}}$:

$$\mathcal{A}_{\mathsf{aux}} := \big\{a : \big(\exists r_b.\{b\} \sqcap \forall r_b.\{b\}\big) | a, b \in \mathsf{Ind}\big\}.$$

To continue, we first introduce the following abbreviations, for $i \leq n$:

$$
\begin{aligned}
\mathsf{p}_i(C(a)) &:= \forall r_a . T_C^{(i)}\\
\mathsf{p}_i(r(a,b)) &:= \forall r_a . \exists r^{(i)}.\{b\}\\
\mathsf{p}_i(\neg r(a,b)) &:= \forall r_a . \forall r^{(i)}. \neg\{b\}.
\end{aligned}
$$

The next component of $\mathcal{A}_{\mathsf{red}}$ formalizes satisfaction of the post-conditions. Note that its formulation relies on $\mathcal{A}_{\mathsf{aux}}$. For $1 \leq i \leq n$, we define

$$\mathcal{A}_{\mathsf{post}}^{(i)} := \big\{(\big(\mathsf{p}_{i-1}(\varphi) \rightarrow \mathsf{p}_i(\psi)\big)(a_0) \mid \varphi/\psi \in \mathsf{post}_i\big\}.$$

The following component formalizes occlusions of role names on named elements. For $1 \leq i \leq n$, role names r, and $a, b \in \mathsf{Ind}$ with $\{(\{a\}, \{b\}), (\top, \top)\} \cap \mathsf{occ}_i(r) = \emptyset$, the ABox $\mathcal{A}_{\mathsf{fix}}^{(i)}$ contains the following assertions:

$$\begin{array}{rcl}\big(\exists r^{(i-1)}.\{b\} & \rightarrow & \exists r^{(i)}.\{b\}\big)(a)\\ \big(\forall r^{(i-1)}.\neg\{b\} & \rightarrow & \forall r^{(i)}.\neg\{b\}\big)(a).\end{array}$$

The ABox $\mathcal{A}_{\mathsf{ini}}$ ensures that the first interpretation of the encoded sequence is a model of the input ABox $\mathcal{A}_0$:

$$\begin{array}{rcl}\mathcal{A}_{\mathsf{ini}} & := & \{T_C^{(0)}(a) \mid C(a) \in \mathcal{A}_0\} \cup \\ & & \{r^{(0)}(a,b) \mid r(a,b) \in \mathcal{A}_0\} \cup \\ & & \{\neg r^{(0)}(a,b) \mid \neg r(a,b) \in \mathcal{A}_0\}.\end{array}$$

We can now assemble $\mathcal{A}_{\mathsf{red}}$:

$$\mathcal{A}_{\mathsf{red}} := \mathcal{A}_{\mathsf{ini}} \cup \mathcal{A}_{\mathsf{aux}} \ \cup \bigcup_{1\leq i\leq n} \mathcal{A}_{\mathsf{post}}^{(i)} \ \cup \bigcup_{1\leq i\leq n} \mathcal{A}_{\mathsf{fix}}^{(i)}$$

Then we have the following lemma. The proof is similar to the proof of Lemma 15 in [**?**]. Details are left to the reader.

Lemma 14. *$A_0(a_0)$ is a consequence of applying $\alpha_1, \ldots, \alpha_n$ in $\mathcal{A}_0$ w.r.t. $\mathcal{T}$ iff $\mathcal{A}_{\mathsf{red}} \cup \{A_0^{(n)}(a_0)\}$ is inconsistent w.r.t. $\mathcal{T}_{\mathsf{red}}$.*

7 Undecidability of Strong Consistency

We show that strong consistency is undecidable already in $\mathcal{ALC}$. The proof consists of a reduction of the undecidable *semantic consequence problem* from modal logic. Before formulating the DL version of this problem, we need some preliminaries. We use $\mathcal{ALC}$ concepts with only one fixed role name r, which we call $\mathcal{ALC}_r$-concepts. Accordingly, we also assume that interpretations interpret only concept names and the role name r. A *frame* is a structure $\mathcal{F} = (\Delta^{\mathcal{F}}, r^{\mathcal{F}})$ where $\Delta^{\mathcal{F}}$ is a non-empty set and $r^{\mathcal{F}} \subseteq \Delta^{\mathcal{F}} \times \Delta^{\mathcal{F}}$. An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is *based* on a frame $\mathcal{F}$ iff $\Delta^{\mathcal{I}} = \Delta^{\mathcal{F}}$ and $r^{\mathcal{I}} = r^{\mathcal{F}}$. We say that a concept C is *valid* on $\mathcal{F}$ (written $\mathcal{F} \models C$) iff $C^{\mathcal{I}} = \Delta^{\mathcal{I}}$ for every interpretation $\mathcal{I}$ based on $\mathcal{F}$.

Definition 15 (Semantic consequence problem). Let D and E be $\mathcal{ALC}_r$-concepts. We say that E is a *semantic consequence* of D iff for every frame $\mathcal{F} = (\Delta^{\mathcal{F}}, r^{\mathcal{F}})$ such that $\mathcal{F} \models D$, it holds that $\mathcal{F} \models E$. △

In [17], it is proved that for $\mathcal{ALC}_r$-concepts D and E, the problem "Is E a semantic consequence of D?" is undecidable. We now show that the semantic consequence problem can be reduced to strong consistency. For $\mathcal{ALC}_r$-concepts D and E, we define the action $\alpha_D = (\mathsf{pre}, \{\mathsf{occ}_{\mathsf{taut}}\}, \mathsf{post})$ where $\mathsf{pre} := \{\neg E(a)\}$, $\mathsf{post} := \{\top(a)/(\exists u.\neg D)(a)\}$ (u a role name), and $\mathsf{occ}_{\mathsf{taut}}$ maps r and $\neg r$ to $\{(\bot, \bot)\}$, all other role literals to $\{(\top, \top)\}$, and all concept literals to $\top$. Then the following holds.

Lemma 16. *The action α_D is strongly consistent with the empty TBox iff E is a semantic consequence of D.*

Proof. "$\Rightarrow$" We show the contraposition. Assume that E is not a semantic consequence of D. Then there exists a frame $\mathcal{F} = (\Delta^{\mathcal{F}}, r^{\mathcal{F}})$ such that $\mathcal{F} \models D$ and there is an interpretation $\mathcal{I}$ based on $\mathcal{F}$ such that $E^{\mathcal{I}} \neq \Delta^{\mathcal{I}}$. We take $\mathcal{I}$ based on $\mathcal{F}$ such that $a^{\mathcal{I}} \notin E^{\mathcal{I}}$, thus $\mathcal{I} \models \mathsf{pre}$. But every $\mathcal{I}'$ such that $\mathcal{I} \Rightarrow^{\emptyset}_{\alpha_D} \mathcal{I}'$ must be based on $\mathcal{F}$ (since $r^{\mathcal{I}'} = r^{\mathcal{I}} = r^{\mathcal{F}}$) and must satisfy $D^{\mathcal{I}'} \neq \Delta^{\mathcal{I}'}$ (by the post-condition of α). Since $\mathcal{F} \models D$, there is no such $\mathcal{I}'$. Thus, α_D is not strongly consistent with the empty TBox.

"$\Leftarrow$" Assume that E is a semantic consequence of D. Let $\mathcal{I} \models \mathsf{pre}$. By definition of pre, we have that $a^{\mathcal{I}} \notin E^{\mathcal{I}}$, and thus $\mathcal{I}$ is not based on a frame $\mathcal{F} = (\Delta^{\mathcal{F}}, r^{\mathcal{F}})$ validating E. Since E is a semantic consequence of D, $\mathcal{F}$ is not validating D either, and there is an interpretation $\mathcal{I}'$ based on $\mathcal{F}$ such that $D^{\mathcal{I}'} \neq \Delta^{\mathcal{I}'}$. Take $y \in \Delta^{\mathcal{I}'}$ such that $y \notin D^{\mathcal{I}'}$. Since D is an $\mathcal{ALC}_r$- concept, we may assume that $u^{\mathcal{I}'} = \{(a^{\mathcal{I}'}, y)\}$. Obviously, we have that $\mathcal{I} \Rightarrow^{\emptyset}_{\alpha_D} \mathcal{I}'$, and, consequently, α_D is strongly consistent with the empty TBox. ❑

As an immediate consequence, we obtain the following theorem.

Theorem 17. *Strong consistency of $\mathcal{ALC}$-actions is undecidable, even with the empty TBox.*

8 Discussion

We have introduced an action formalism based on description logics that admits general TBoxes and complex post-conditions. To deal with ramifications induced by general TBoxes, the formalism includes powerful occlusion patterns that can be used to fine-tune the ramifications. Most important reasoning tasks in our formalism turn out to be decidable.

Our only negative result concerns the undecidability of strong consistency. To discuss the impact of this result, let us briefly review the relevance of strong consistency for the action designer and for the user of the action (the person who applies the action).

For the action *designer*, an algorithm for checking strong consistency would be useful for fine-tuning the ramifications of his action. However, it is worth noting that deciding strong consistency could not replace manual inspection of the ramifications. For example, occluding all concept names with $\top$ and all role names with $\{(\top, \top)\}$ usually ensures strong consistency but does not lead to an intuitive behaviour of the action. With weak consistency, we offer at least some automatic support to the action designer for detecting ramification problems.

For the *user* of the action, strong consistency is required to ensure that the execution of an action whose preconditions are satisfied will not fail. If the action is such that failure cannot be tolerated (because executing the action is expensive, dangerous, etc), strong consistency is thus indispensible and should already be guaranteed by the action designer. Also when working with composite actions, strong consistency has to be required: if an action execution fails after previous actions in the sequence have been successfully executed, then we have already changed the state of affairs and it may not be possible to revert these changes to use a different composite action for reaching the desired goal. However, in the case of atomic actions it is conceivable that an execution failure does not have any negative effects. If this is the case, the action user only needs to check that the action is executable, and strong consistency is not strictly required.

Future work will include developing practical decision procedures. A first step is carried out in [8], where we show that in the following special (but natural) case, projection can be reduced to standard reasoning problems in DLs that are implemented in DL reasoners such as RACER and FaCT++: (i) role occlusions in actions are given by $\mathsf{occ}_{\mathsf{taut}}$; (ii) $\mathsf{occ}_{\mathsf{taut}}(r) = \mathsf{occ}_{\mathsf{taut}}(\neg r)$; and (iii) concepts used in $\mathsf{occ}_{\mathsf{taut}}(r)$ are Boolean combinations of nominals,

Acknowledgements. We would like to thank Giuseppe De Giacomo for ideas and discussions. The third author is supported by the DFG Graduiertenkolleg 334. The fourth author is partially supported by UK EPSRC grant no. GR/S63182/01.

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] 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.

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

[4] Volker Haarslev and Ralf Möller. RACER system description. In Rajeev Goré, Alexander Leitsch, and Tobias Nipkow, editors, *Proceedings*

of the First International Joint Conference on Automated Reasoning (IJ-CAR'01), number 2083 in Lecture Notes in Artifical Intelligence, pages 701–705. Springer-Verlag, 2001.

[5] Ian Horrocks. Using an expressive description logic: Fact or fiction? In *Proceedings of the Sixth International Conference on the Principles of Knowledge Representation and Reasoning (KR98)*, pages 636–647, 1998.

[6] Hector J. Levesque, Raymond Reiter, Yves Lespérance, Fangzhen Lin, and Richard B. Scherl. GOLOG: A logic programming language for dynamic domains. *J. Log. Program.*, 31(1-3):59–83, 1997.

[7] Fangzhen Lin. Embracing causality in specifying the indirect effects of actions. In C. S. Mellish, editor, *Proc. of IJCAI-95*, pages 1985–1991, Montreal, Canada, August 1995. Morgan Kaufmann.

[8] H. Liu, C. Lutz, M. Milicic, and F. Wolter. Description logic actions with general tboxes: a pragmatic approach. LTCS-Report 06-03, TU Dresden, Germany, 2006. See http://lat.inf.tu-dresden.de/research/reports.html.

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

[10] V. R. Pratt. Models of program logics. In *Proceedings of the Twentieth Annual Symposium on Foundations of Computer Science*, San Juan, Puerto Rico, 1979.

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

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

[13] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical owl-dl reasoner. Submitted for publication to Journal of Web Semantics.

[14] Michael Thielscher. Ramification and causality. *Artificial Intelligence Journal*, 89(1–2):317–364, 1997.

[15] Michael Thielscher. Introduction to the Fluent Calculus. *Electronic Transactions on Artificial Intelligence*, 2(3–4):179–192, 1998.

[16] Michael Thielscher. FLUX: A logic programming method for reasoning agents. *TPLP*, 5(4-5):533–565, 2005.

[17] S. K. Thomason. The logical consequence relation of propositional tense logic. *Z. Math. Logik Grundl. Math.*, 21:29–40, 1975.

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

[19] Marianne Winslett. Reasoning about action using a possible models approach. In *AAAI*, pages 89–93, Saint Paul, MN, 1988.

A Relating DL Action Formalisms

We show that the action formalism introduced in this paper is a generalization of the one from [2]. To begin with, we recall a couple of relevant definitions. First, the semantics of actions in [2] is defined only w.r.t. acyclic TBoxes.

Definition 18 (Acyclic TBox). A *concept definition* is an identity of the form $A \doteq C$, where A is a concept name and C a concept. An *acyclic TBox* $\mathcal{T}$ is a finite set of concept definitions with unique left-hand sides such that there are no cyclic dependencies between the definitions. Concept names occurring on the left-hand side of a definition of $\mathcal{T}$ are called *defined in* $\mathcal{T}$ whereas the others are called *primitive in* $\mathcal{T}$. $\triangle$

Obviously, general TBoxes generalize acyclic TBoxes. Actions in [2] are more restricted than the actions introduced in this paper. For example, the former do not allow post-conditions $\varphi/C(a)$ with C a complex concept or defined concept name. Thus, in post-conditions of this form, C must be a primitive concept name. For this reason, we call the actions from [2] (where p stands for primitive). The actions from Definition 1 are called u-actions (where u stands for unrestricted).

Definition 19 (Syntax and semantics of p-actions). Let $\mathcal{T}$ be an acyclic TBox. An *(atomic)* p*-action* $\alpha = (\mathsf{pre}, \mathsf{occ}, \mathsf{post})$ for $\mathcal{T}$ consists of

- a finite set pre of ABox assertions, the *pre-conditions*;
- a finite set occ of *occlusions* of the form $A(a)$ or $r(a,b)$, with A a primitive concept in $\mathcal{T}$, r a role name, and $a, b \in \mathsf{N_I}$;
- a finite set of post of *conditional post-conditions* of the form φ/ψ, where φ is an ABox assertion and ψ is a *primitive literal for* $\mathcal{T}$, i.e., an ABox assertion $A(a)$, $\neg A(a)$, $r(a,b)$, $\neg r(a,b)$ with A a concept name primitive in $\mathcal{T}$ and r a role name.

For an interpretation $\mathcal{I}$, a concept name A, and a role name r, we introduce the following abbreviations:

$$
\begin{array}{rcl}
A^+ & := & \{b^\mathcal{I} \mid \varphi/A(b) \in \mathsf{post} \wedge \mathcal{I} \models \varphi\} \\
A^- & := & \{b^\mathcal{I} \mid \varphi/\neg A(b) \in \mathsf{post} \wedge \mathcal{I} \models \varphi\} \\
I_A & := & (\Delta^\mathcal{I} \setminus \{b^\mathcal{I} \mid A(b) \in \mathsf{occ}\} \cup (A^+ \cup A^-)) \\
r^+ & := & \{(a^\mathcal{I}, b^\mathcal{I}) \mid \varphi/r(a,b) \in \mathsf{post} \wedge \mathcal{I} \models \varphi\} \\
r^- & := & \{(a^\mathcal{I}, b^\mathcal{I}) \mid \varphi/\neg r(a,b) \in \mathsf{post} \wedge \mathcal{I} \models \varphi\} \\
I_r & := & (\Delta^\mathcal{I} \setminus \{(a^\mathcal{I}, b^\mathcal{I}) \mid r(b) \in \mathsf{occ}\} \cup (r^+ \cup r^-))
\end{array}
$$

Let $\mathcal{I}$, $\mathcal{I}'$ be models of $\mathcal{T}$ sharing the same domain and interpretation of all individual names. We say that α *may transform* $\mathcal{I}$ *to* $\mathcal{I}'$ *w.r.t.* $\mathcal{T}$ ($\mathcal{I} \Rightarrow_{\alpha}^{\mathcal{T}} \mathcal{I}'$) iff, for each primitive concept A and role name r, we have

$$
\begin{array}{rcl}
A^+ \cap A^- = \emptyset & \text{and} & r^+ \cap r^- = \emptyset \\
A^{\mathcal{I}'} \cap I_A & = & ((A^{\mathcal{I}} \cup A^+) \setminus A^-) \cap I_A \\
r^{\mathcal{I}'} \cap I_r & = & ((r^{\mathcal{I}} \cup r^+) \setminus r^-) \cap I_r
\end{array}
$$

△

Let $\mathcal{T}$ be an acyclic TBox and $\alpha = (\mathsf{pre}, \mathsf{occ}, \mathsf{post})$ a p-action. We use P to denote the set $\{\varphi \mid \exists\psi : \varphi/\psi \in \mathsf{post}\}$. We now construct a u-action α' that is equivalent to α in the sense that for all models $\mathcal{I}, \mathcal{I}'$ of $\mathcal{T}$, $\mathcal{I} \Rightarrow_{\alpha}^{\mathcal{T}} \mathcal{I}'$ iff $\mathcal{I} \Rightarrow_{\alpha'}^{\mathcal{T}} \mathcal{I}'$. Set $\alpha' := (\mathsf{pre}, \mathsf{occ}', \mathsf{post})$, where the occlusion pattern is defined as

$$\mathsf{occ}' := \{\mathsf{occ}_{\mathsf{taut}}\} \cup \{\mathsf{occ}_{\varphi} \mid \varphi/\psi \in \mathsf{post}\},$$

with the components $\mathsf{occ}_{\mathsf{taut}}$ and occ_{φ}, $\varphi/\psi \in \mathsf{post}$ are as follows:

- for every primitive concept name A and $\varphi \in P$,
  - $\mathsf{occ}_{\mathsf{taut}}(A) := \mathsf{occ}_{\mathsf{taut}}(\neg A) := \bigsqcup_{A(a) \in \mathsf{occ}} \{a\}$;
  - $\mathsf{occ}_{\varphi}(A) := \bigsqcup_{\varphi/\neg A(a) \in \mathsf{post}} \{a\}$;
  - $\mathsf{occ}_{\varphi}(\neg A) := \bigsqcup_{\varphi/A(a) \in \mathsf{post}} \{a\}$;
- for every role name r and $\varphi \in P$,
  - $\mathsf{occ}_{\mathsf{taut}}(r) := \mathsf{occ}_{\mathsf{taut}}(\neg r) := \bigcup_{r(a,b) \in \mathsf{occ}} \{(\{a\}, \{b\})\}$;
  - $\mathsf{occ}_{\varphi}(r) := \bigcup_{\varphi/\neg r(a,b) \in \mathsf{post}} \{(\{a\}, \{b\})\}$;
  - $\mathsf{occ}_{\varphi}(\neg r) := \bigcup_{\varphi/r(a,b) \in \mathsf{post}} \{(\{a\}, \{b\})\}$;
- for every defined concept name C and $\varphi \in P$,

$$\mathsf{occ}_{\mathsf{taut}}(C) = \mathsf{occ}_{\mathsf{taut}}(\neg C) = \mathsf{occ}_{\varphi}(C) = \mathsf{occ}_{\varphi}(\neg C) := \top.$$

The following lemma shows that α and α' are indeed equivalent.

Lemma 20. *For all models $\mathcal{I}$ and $\mathcal{I}'$ of $\mathcal{T}$, $\mathcal{I} \Rightarrow_{\alpha}^{\mathcal{T}} \mathcal{I}'$ iff $\mathcal{I} \Rightarrow_{\alpha'}^{\mathcal{T}} \mathcal{I}'$.*

Proof. "$\Rightarrow$". Let $\mathcal{I} \Rightarrow_{\alpha}^{\mathcal{T}} \mathcal{I}'$. We show that this implies $\mathcal{I} \Rightarrow_{\alpha'}^{\mathcal{T}} \mathcal{I}'$. This amounts to verifying the conditions from Definition 2.

- Let $\varphi/\psi \in \mathsf{post}$ and $\mathcal{I} \models \varphi$. Then
  - if ψ is of the form $A(a)$, then $a^{\mathcal{I}} \in A^{+}$. Thus, $a^{\mathcal{I}} \notin A^{-}$ and $a^{\mathcal{I}} \in I_A$. Then, $a^{\mathcal{I}} \in A^{\mathcal{I}'}$ since $A^{\mathcal{I}'} \cap I_A = ((A^{\mathcal{I}} \cup A^{+}) \setminus A^{-}) \cap I_A$. Hence, $\mathcal{I}' \models \psi$ since $a^{\mathcal{I}} = a^{\mathcal{I}'}$.
  - for the cases that ψ is of the form $\neg A(a)$, $r(a,b)$, or $\neg r(a,b)$, we can prove $\mathcal{I}' \models \psi$ similarly.
- Let A be a concept name. Then
  - if A is a primitive concept name,
    - for every $x \in A^{\mathcal{I}} \setminus A^{\mathcal{I}'}$, we have either $x \in A^{-}$ or $x \notin I_A$ since $A^{\mathcal{I}'} \cap I_A = ((A^{\mathcal{I}} \cup A^{+}) \setminus A^{-}) \cap I_A$. From either of them, we can get $x \in (\mathsf{occ}(A))^{\mathcal{I}}$.
    - for every $x \in (\neg A)^{\mathcal{I}} \setminus (\neg A)^{\mathcal{I}'}$, i.e., $x \notin A^{\mathcal{I}}$ and $x \in A^{\mathcal{I}'}$, we have either $x \in A^{+}$ or $x \notin I_A$ since $A^{\mathcal{I}'} \cap I_A = ((A^{\mathcal{I}} \cup A^{+}) \setminus A^{-}) \cap I_A$. From either of them, we can get $x \in (\mathsf{occ}(A))^{\mathcal{I}}$.
  - if A is a defined concept name, we have $(\mathsf{occ}(A))^{\mathcal{I}} = (\mathsf{occ}(\neg A))^{\mathcal{I}} = \Delta_{\mathcal{I}}$. Thus, $A^{\mathcal{I}} \setminus A^{\mathcal{I}'} \subseteq (\mathsf{occ}(A))^{\mathcal{I}}$ and $A^{\mathcal{I}'} \setminus A^{\mathcal{I}} \subseteq (\mathsf{occ}(\neg A))^{\mathcal{I}}$
- Let r be a role name. Then
  - for every $(x,y) \in r^{\mathcal{I}} \setminus r^{\mathcal{I}'}$, we have either $(x,y) \in r^{-}$ or $x \notin I_r$ since $r^{\mathcal{I}'} \cap I_r = ((r^{\mathcal{I}} \cup r^{+}) \setminus r^{-}) \cap I_r$. From either of them, we can get $(x,y) \in (\mathsf{occ}(r))^{\mathcal{I}}$.
  - for every $(x,y) \in (\neg r)^{\mathcal{I}} \setminus (\neg r)^{\mathcal{I}'}$, i.e., $(x,y) \notin r^{\mathcal{I}}$ and $(x,y) \in r^{\mathcal{I}'}$, we have either $(x,y) \in r^{+}$ or $(x,y) \notin I_r$ since $r^{\mathcal{I}'} \cap I_r = ((r^{\mathcal{I}} \cup r^{+}) \setminus r^{-}) \cap I_r$. From either of them, we can get $(x,y) \in (\mathsf{occ}(r))^{\mathcal{I}}$.

"$\Leftarrow$". Assume $\mathcal{I} \Rightarrow_{\alpha'}^{\mathcal{T}} \mathcal{I}'$. We show that then, $\mathcal{I} \Rightarrow_{\alpha}^{\mathcal{T}} \mathcal{I}'$. This amounts to checking the conditions from Definition 19.

- $A^{+} \cap A^{-} = \emptyset$: Assume the opposite. Then there are $\varphi_1/A(a), \varphi_2/\neg A(a) \in \mathsf{post}$ such that $\mathcal{I} \models \varphi_1$ and $\mathcal{I} \models \varphi_2$. But then $\mathcal{I}' \models A(a)$, $\mathcal{I}' \models \neg A(a)$, which is impossible. Analogously, we have $r^{+} \cap r^{-} = \emptyset$.
- for every primitive concept name A, $A^{\mathcal{I}'} \cap I_A = ((A^{\mathcal{I}} \cup A^{+}) \setminus A^{-}) \cap I_A$:
  - $\supseteq$: Consider an arbitrary x with $x \in ((A^{\mathcal{I}} \cup A^{+}) \setminus A^{-}) \cap I_A$. If $x \in (A^{\mathcal{I}} \cup A^{+}) \setminus A^{-}$, then either $x \in A^{\mathcal{I}} \setminus A^{-}$ or $x \in A^{+} \setminus A^{-}$.

* Let x be in $A^{\mathcal{I}} \setminus A^-$. Assume that x is not in $A^{\mathcal{I}'}$. Thus, $x \in A^{\mathcal{I}} \setminus A^{\mathcal{I}'}$ implies $x \in (\mathsf{occ}(A))^{\mathcal{I}}$ since $\mathcal{I} \Rightarrow^{\mathcal{T}}_{\alpha'} \mathcal{I}'$. Hence, we have either there is a $\varphi/\neg A(a) \in \mathsf{post}$ with $a^{\mathcal{I}} = x$ and $\mathcal{I} \models \varphi$ and thus this implies $x \in A^{\mathcal{I}'}$ since $\mathcal{I} \Rightarrow^{\mathcal{T}}_{\alpha'} \mathcal{I}'$, or there exists $A(a) \in \mathsf{occ}$ with $a^{\mathcal{I}} = x$ which contradicts $x \in I_A$.
* Let x be in $A^+ \setminus A^-$. $x \in A^+$ implies there is a $\varphi/A(a) \in \mathsf{post}$ with $a^{\mathcal{I}} = x$ and $\mathcal{I} \models \varphi$. Since $\mathcal{I} \Rightarrow^{\mathcal{T}}_{\alpha'} \mathcal{I}'$, $\mathcal{I}' \models A(a)$ and this yields $x \in A^{\mathcal{I}'}$.

– $\subseteq$: Consider an arbitrary x with $x \in A^{\mathcal{I}'} \cap I_A$. $x \in A^{\mathcal{I}'}$ implies $x \notin A^-$ since $\mathcal{I} \Rightarrow^{\mathcal{T}}_{\alpha'} \mathcal{I}'$. It is enough to show $x \in A^{\mathcal{I}} \cup A^+$. Equivalently, we show that $x \notin A^{\mathcal{I}}$ implies $x \in A^+$. From $x \notin A^{\mathcal{I}}$ and $x \in A^{\mathcal{I}'}$ we get $x \in (\neg A)^{\mathcal{I}} \setminus (\neg A)^{\mathcal{I}'}$. Thus, $x \in (\mathsf{occ}(\neg A))^{\mathcal{I}}$ since $\mathcal{I} \Rightarrow^{\mathcal{T}}_{\alpha'} \mathcal{I}'$. $x \in I_A$ implies there exists $\varphi/A(a) \in \mathsf{post}$ with $a^{\mathcal{I}} = x$ and $\mathcal{I} \models \varphi$. Hence, $x \in A^+$.

- $r^{\mathcal{I}'} \cap I_r = ((r^{\mathcal{I}} \cup r^+) \setminus r^-) \cap I_r$ for every role name r can be obtained analogously.

❏