

Projection in a Description Logic of Context with Actions^{*}

Satyadharna Tirtarasa and Benjamin Zarriess

Institute for Theoretical Computer Science
Technische Universität Dresden
`{satyadharna.tirtarasa,benjamin.zarriess}@tu-dresden.de`

Abstract. Projection is the problem of checking whether the execution of a given sequence of actions will achieve its goal starting from some initial state. In this paper, we study a setting where we combine a two-dimensional Description Logic of context (ConDL) with an action formalism. We choose a well-studied ConDL where both: the possible states of a dynamical system itself (object level) and also different context-dependent views on this system state (context level) are organised in relational structures and can be described using usual DL constructs. To represent how such a system and its views evolve we introduce a suitable action formalism. It allows one to describe change on both levels. Furthermore, the observable changes on the object level due to an action execution can also be context-dependent. We show that the formalism is well-behaved in the sense that projection has the same complexity as standard reasoning tasks in case *ALCO* is the underlying DL.

1 Introduction

The *role-based* paradigm of modelling languages has been introduced for the design of adaptive and context-sensitive software systems. The concept of roles has been used at different levels of abstraction, for example in data models [6], in a formal high-level modelling language [11] for dynamical systems, and as an extension of more low-level object-oriented programming languages [12]. Unlike in a classical object-oriented setting, where an object has a fixed number of methods attached to it, in a role-based setting an object adapts its behaviour dynamically according to the roles it can play in different contexts. For example, in the conference management system used for this workshop the concept of roles is quite prominent. In the context of this workshop a researcher might play the role of an author whereas in context of some other conference also the role of a program committee member can be played by the same researcher. Both the view on submissions and the abilities to change something are context-dependent and may vary over time in this scenario.

How to deal with explicit context extensions of modelling languages efficiently is a well-studied research topic in different areas (e.g. [10,12,7]). In [8], Böhme

^{*} Supported by DFG within the Research Training Group 1907 (RoSI) and the DFG Research Unit FOR 1513

and Lippmann studied a family of *contextualized Description Logics* (ConDLs). For this family, a reasoning tool has been implemented, and has been used for translating and checking consistency of models of a role-based modelling language for software systems [11,7].

However, ConDLs are only suitable for expressing *static* context-dependent knowledge. In this paper, we focus on an extension with *dynamic aspects* and introduce a *ConDL-based action formalism* for reasoning about change in context models. To talk about particular states we consider the ConDL $\mathcal{ALCCO}[\mathcal{ALCCO}]$ from [8]. It is a two-sorted logic with a *meta level* signature for describing contexts and an *object level* signature for the object domain. \mathcal{ALCCO} is the outer *meta level* logic and is used to describe sets of contexts and relations among them. Each context element of the meta level domain corresponds to a relational structure of the inner object level, which is represented using \mathcal{ALCCO} as well. Both levels are connected with a modality that allows one to access the object level from the meta level. In the example of the conference management system one could think of a model, where we talk about researcher accounts with their properties (for example, being PC member or author) and relations (like conflict of interest with someone else) on the meta level, and where each account corresponds to an individual view on concrete submissions and reviews on the object level. The action formalism we introduce makes it possible to describe changes on both levels. For example, the meta level can change if someone becomes a PC member or declares conflict of interest with someone. An object level action might represent the changes when a particular review is entered for a submission. The observable changes of this action from the perspective of a particular account depend on its meta level properties. As a reasoning task we consider the *projection problem*. Projection is the problem of checking whether the execution of a given sequence of actions will achieve its goal starting from some initial state. In our example, a typical projection query could ask whether, after a subreviewer has accepted to review some submission s due to an invitation by a PC member and after a review has been entered by some other PC member for this submission s , the subreviewer is able to see this review or not. To solve projection we reduce it polynomially to consistency in the underlying logic $\mathcal{ALCCO}[\mathcal{ALCCO}]$ by adapting techniques that have been used before for reasoning in DL-based action formalisms [5].

The remainder of this paper is structured as follows. In the next section we recall the definitions of $\mathcal{ALCCO}[\mathcal{ALCCO}]$. Section 3 introduces our action formalism and defines the projection problem. In Section 4, we present our reduction method for deciding projection. We finish with a conclusion in Section 5.

2 The Description Logic of Context $\mathcal{ALCCO}[\mathcal{ALCCO}]$

For representing context-dependent knowledge we choose $\mathcal{ALCCO}[\mathcal{ALCCO}]$, a simple member of the family of ConDLs studied in [8]. To keep this part as simple as possible we omit rigidity constraints and focus only on \mathcal{ALCCO} on both levels.

The full Boolean concept constructors and nominals are essential in our setting because the actions we define later act on concrete individuals.

The logic is *two-sorted* with a *meta level signature* $M = (M_C, M_R, M_I)$ and an *object level signature* $O = (O_C, O_R, O_I)$. We call M_C , M_R and M_I the set of *meta concept names*, *role names*, and *individual names* respectively. Analogously, O_C , O_R , O_I is called the set of *object concept names*, *role names*, and *individual names* respectively. All these sets are assumed to be pairwise disjoint.

We assume the standard definition of the syntax of \mathcal{ALCO} -concepts, *general concept inclusions (GCIs)* and KBs (see [2,3] for details) and the corresponding semantics in terms of interpretations over some signature (either M or O in our case) with the unique name assumption for individual names.

Definition 1 (Syntax). *Let φ be an \mathcal{ALCO} -KB over the object level signature O and $A \in M_C$, $r \in M_R$ and $a \in M_I$ meta level names. An $\mathcal{ALCO}[\mathcal{ALCO}]$ -meta level concept description C over M and O (m-concept for short) is built according to the following syntax rule*

$$C ::= A \mid \{a\} \mid \llbracket \varphi \rrbracket \mid C \sqcap C \mid \neg C \mid \exists r.C.$$

Further constructors are defined as abbreviations: $\top := \neg(A \sqcap \neg A)$ and $\perp := (A \sqcap \neg A)$ (for some $A \in M_C$), $C \sqcup D := \neg(C \sqcap \neg D)$ and $\forall r.C := \neg \exists r.\neg C$.

Let C and D be m-concepts. An $\mathcal{ALCO}[\mathcal{ALCO}]$ -Boolean meta level knowledge base ψ over M and O (m-KB for short) is built according to the following syntax rule

$$\psi ::= C \sqsubseteq D \mid \psi \wedge \psi \mid \neg \psi.$$

Notation for concept assertions and role assertions is used as abbreviations: $(a : C) := \{a\} \sqsubseteq C$ and $((a, b) : r) := \{a\} \sqsubseteq \exists r.\{b\}$. Further Boolean connectives like \vee and \rightarrow are defined as usual.

The only non-standard expression is the meta concept constructor $\llbracket \varphi \rrbracket$ that refers to a standard \mathcal{ALCO} -KB φ . It gives access to the object level from the meta level and describes the set of meta level domain elements (contexts) in which φ is true.

The semantics of $\mathcal{ALCO}[\mathcal{ALCO}]$ is defined in terms of *nested interpretations*. The structure consists of a single meta level interpretation over M where each domain element is associated with an object level interpretation over O .

Definition 2 (Nested Interpretation). *A nested interpretation \mathfrak{J} (over M and O) is a tuple of the form $\mathfrak{J} := (\mathbb{C}, \cdot^{\mathfrak{J}}, \Delta, \{\mathcal{I}_c\}_{c \in \mathbb{C}})$, where*

- $(\mathbb{C}, \cdot^{\mathfrak{J}})$ is an M -interpretation, and
- $\mathcal{I}_c := (\Delta, \cdot^{\mathcal{I}_c})$ is an O -interpretation for each $c \in \mathbb{C}$.

Definition 3 (Semantics). *Let $\mathfrak{J} = (\mathbb{C}, \cdot^{\mathfrak{J}}, \Delta, \{\mathcal{I}_c\}_{c \in \mathbb{C}})$ be a nested interpretation. The extension of the mapping $\cdot^{\mathfrak{J}}$ to complex m-concepts is defined by*

induction on the structure of m -concepts C and D as follows:

$$\begin{aligned}
(\{a\})^{\mathfrak{I}} &:= \{a^{\mathfrak{I}}\}; \\
(\llbracket \varphi \rrbracket)^{\mathfrak{I}} &:= \{c \in \mathbb{C} \mid \mathcal{I}_c \models \varphi\}; \\
(C \sqcap D)^{\mathfrak{I}} &:= C^{\mathfrak{I}} \cap D^{\mathfrak{I}}; \\
(\neg C)^{\mathfrak{I}} &:= \mathbb{C} \setminus C^{\mathfrak{I}}; \\
(\exists r.C)^{\mathfrak{I}} &:= \{c \in \mathbb{C} \mid \text{there exists } c' \in \mathbb{C} \text{ such that } (c, c') \in r^{\mathfrak{I}} \text{ and } c' \in C^{\mathfrak{I}}\},
\end{aligned}$$

where $a \in \mathbb{M}_I$, $r \in \mathbb{M}_R$ and φ is an \mathcal{ALCO} -KB over \mathbb{O} .

Let ψ be an m -KB. Satisfaction of ψ in \mathfrak{I} , written as $\mathfrak{I} \models \psi$ (\mathfrak{I} is a model of ψ), is defined by induction on the structure of ψ as follows:

$$\begin{aligned}
\mathfrak{I} \models C \sqsubseteq D &\text{ iff } C^{\mathfrak{I}} \subseteq D^{\mathfrak{I}}; \\
\mathfrak{I} \models \psi_1 \wedge \psi_2 &\text{ iff } \mathfrak{I} \models \psi_1 \text{ and } \mathfrak{I} \models \psi_2; \\
\mathfrak{I} \models \neg \psi_1 &\text{ iff } \mathfrak{I} \not\models \psi_1.
\end{aligned}$$

Example 1. We describe some aspects of a conference management domain. On the meta level we talk about accounts that can be PC members of the DL workshop (meta level concept name *DL-PC*) with possibly conflict of interest (meta level role *has-conflict*) to authors (concept name *DL-Author*). Each account has a particular view on the object level where we have a domain of submissions and reviews. The object level concept names *Subs-To-Review* and *Own-Subs* describe the assigned submissions for reviewing and their own written submissions, respectively. The object level role *has-review* relates submissions to their reviews. We describe an initial situation using the meta level individual names *bob's-account* and *alice'-account* and the object level name *s* denoting a concrete submission. Intuitively, in this model the meta level concept

$$\llbracket s : \textit{Own-Subs} \rrbracket$$

describes the set of accounts (meta level domain elements) in which *s* is an instance of *Own-Subs*. Therefore, it represents the set of author accounts of *s*. The following meta level axioms represent some initial knowledge:

$$\textit{alice'-account} : (\forall \textit{has-conflict}. \neg \llbracket s : \textit{Own-Subs} \rrbracket) \quad (1)$$

$$\textit{bob's-account} : \llbracket s : \textit{Own-Subs} \rrbracket \quad (2)$$

$$\top \sqsubseteq \llbracket s : \forall \textit{has-review}. \perp \rrbracket \quad (3)$$

$$\textit{DL-Author} \equiv \neg \llbracket \textit{Own-Subs} \sqsubseteq \perp \rrbracket \quad (4)$$

$$\neg \llbracket \textit{Subs-To-Review} \sqsubseteq \perp \rrbracket \sqsubseteq \textit{DL-PC} \quad (5)$$

Alice has no conflict of interest with an author of the submission *s* (1). Bob is an author of *s* (2), which has not received any reviews yet (3). DL author accounts are defined as those accounts with own submissions (4). Only PC members are allowed to review (5).

3 Representing Context-dependent Change

We define separate *action descriptions* for the object level and the meta level. Syntactically, action descriptions are complex expressions with constructors for describing conditional and simultaneous execution. Semantically, actions update interpretations by changing the membership of named individuals in concept names or of pairs of named individuals in role names.

Definition 4. Let ψ be an *m-KB* and $A \in \mathbf{M}_C$, $r \in \mathbf{M}_R$ and $a, b \in \mathbf{M}_I$ be meta level names. An *M-action description* α (*M-action for short*) is built according to the following syntax rule:

$$\alpha := \langle A \oplus a \rangle \mid \langle A \ominus a \rangle \mid \langle r \oplus (a, b) \rangle \mid \langle r \ominus (a, b) \rangle \mid (\psi \triangleright \alpha) \mid (\alpha \parallel \alpha).$$

Let C be an *m-concept* and $B \in \mathbf{O}_C$, $s \in \mathbf{O}_R$ and $o, o' \in \mathbf{O}_I$ object level names. An *O-action description* β (*O-action for short*) is built according to the following syntax rule:

$$\beta := \langle B \oplus o \rangle \mid \langle B \ominus o \rangle \mid \langle s \oplus (o, o') \rangle \mid \langle s \ominus (o, o') \rangle \mid (C \triangleright \beta) \mid (\beta \parallel \beta).$$

We write just *action* if we do not distinguish between *M-actions* and *O-actions*. Actions of the form $\langle A \oplus a \rangle$, $\langle r \oplus (a, b) \rangle$, $\langle B \oplus o \rangle$ or $\langle s \oplus (o, o') \rangle$ are called *atomic actions*.

An atomic action like $\langle A \oplus a \rangle$ over some signature \mathbf{N} changes an *N-interpretation* \mathcal{I} by adding $a^{\mathcal{I}}$ to $A^{\mathcal{I}}$ and $\langle r \ominus (a, b) \rangle$ deletes $(a^{\mathcal{I}}, b^{\mathcal{I}})$ from $r^{\mathcal{I}}$. *M-actions* have *m-KBs* as conditions and *O-actions* have *m-concepts* as conditions. A conditional *M-action* $(\psi \triangleright \alpha_1)$ takes effect in the meta-level interpretation only if ψ is satisfied and a conditional *O-action* of the form $C \triangleright \beta_1$ means that an *O-interpretation* \mathcal{I}_c in a nested interpretation \mathcal{J} is only updated with β_1 if c belongs to C in \mathcal{J} . The construct $(\alpha_1 \circ \alpha_2)$ means that α_1 and α_2 are executed simultaneously.

For the first step of the definition of the execution semantics we define how a set of atomic *M-actions* (or atomic *O-actions*) updates a *non-nested M-interpretation* (or *O-interpretation*).

Definition 5 (Update). Let $\mathbf{N} \in \{\mathbf{M}, \mathbf{O}\}$ denote either the meta-level or object-level signature, and let $\mathcal{I} := (\Delta_{\mathcal{I}}, \cdot^{\mathcal{I}})$ be an *N-interpretation* and \mathfrak{E} a set of atomic *N-actions*. The update of \mathcal{I} with \mathfrak{E} is an interpretation denoted by $\mathcal{I}^{\mathfrak{E}}$ and is defined for all $A \in \mathbf{N}_C$, all $r \in \mathbf{N}_R$ and all $a \in \mathbf{N}_I$ as follows

$$\begin{aligned} \Delta_{\mathcal{I}^{\mathfrak{E}}} &:= \Delta_{\mathcal{I}}; \\ A^{\mathcal{I}^{\mathfrak{E}}} &:= (A^{\mathcal{I}} \setminus \{a^{\mathcal{I}} \mid \langle A \ominus a \rangle \in \mathfrak{E}\}) \cup \{b^{\mathcal{I}} \mid \langle A \oplus b \rangle \in \mathfrak{E}\} \\ r^{\mathcal{I}^{\mathfrak{E}}} &:= (r^{\mathcal{I}} \setminus \{(a^{\mathcal{I}}, b^{\mathcal{I}}) \mid \langle r \ominus (a, b) \rangle \in \mathfrak{E}\}) \cup \{(a^{\mathcal{I}}, b^{\mathcal{I}}) \mid \langle r \oplus (a, b) \rangle \in \mathfrak{E}\} \\ a^{\mathcal{I}^{\mathfrak{E}}} &:= a^{\mathcal{I}}. \end{aligned}$$

Next, we define the effects of the execution of a complex action in a *nested interpretation* as a set of atomic actions. The definition tells us how the conditions attached by the “ \triangleright ” operator are evaluated for *M-actions* and *O-actions*.

Definition 6 (Effects). Let $\mathfrak{J} = (\mathbb{C}, \cdot^{\mathfrak{J}}, \Delta, \{\mathcal{I}_c\}_{c \in \mathbb{C}})$ be a nested interpretation and α an M-action. The set of atomic M-actions for \mathfrak{J} and α , denoted by $\mathcal{E}(\alpha, \mathfrak{J})$, is defined by induction on the structure of α as follows

$$\begin{aligned} \mathcal{E}(\langle A \oplus a \rangle, \mathfrak{J}) &:= \{\langle A \oplus a \rangle\} \quad \text{and} \quad \mathcal{E}(\langle r \oplus (a, b) \rangle, \mathfrak{J}) := \{\langle r \oplus (a, b) \rangle\} \\ \mathcal{E}(\psi \triangleright \alpha_1, \mathfrak{J}) &:= \begin{cases} \mathcal{E}(\alpha_1, \mathfrak{J}) & \text{if } \mathfrak{J} \models \psi, \\ \emptyset & \text{otherwise;} \end{cases} \\ \mathcal{E}(\alpha_1 \parallel \alpha_2, \mathfrak{J}) &:= \mathcal{E}(\alpha_1, \mathfrak{J}) \cup \mathcal{E}(\alpha_2, \mathfrak{J}). \end{aligned}$$

Let β be an O-action and $c \in \mathbb{C}$. The set of sets of atomic O-actions for \mathfrak{J} , c and β , denoted by $\mathcal{E}(\beta, c, \mathfrak{J})$, is defined by induction on the structure of β as follows

$$\begin{aligned} \mathcal{E}(\langle B \oplus o \rangle, c, \mathfrak{J}) &:= \{\langle B \oplus o \rangle\} \quad \text{and} \quad \mathcal{E}(\langle s \oplus (o, o') \rangle, c, \mathfrak{J}) := \{\langle s \oplus (o, o') \rangle\} \\ \mathcal{E}(C \triangleright \beta_1, c, \mathfrak{J}) &:= \begin{cases} \mathcal{E}(\beta_1, c, \mathfrak{J}) & \text{if } c \in C^{\mathfrak{J}}, \\ \emptyset & \text{otherwise;} \end{cases} \\ \mathcal{E}(\beta_1 \parallel \beta_2, c, \mathfrak{J}) &:= \mathcal{E}(\beta_1, c, \mathfrak{J}) \cup \mathcal{E}(\beta_2, c, \mathfrak{J}). \end{aligned}$$

An M-action only updates the outer meta level interpretation of a nested interpretation. An O-action leaves the meta level interpretation unchanged and updates *all* object level interpretation simultaneously. How a particular object level interpretation \mathcal{I}_c in a nested interpretation \mathfrak{J} is updated as the result of executing an O-action β depends only on the membership of c in the meta level concepts used as conditions in β . The corresponding set $\mathcal{E}(\beta, c, \mathfrak{J})$ can also be empty.

Definition 7 (Nested Update). Let $\mathfrak{J} := (\mathbb{C}, \cdot^{\mathfrak{J}}, \Delta, \{\mathcal{I}_c\}_{c \in \mathbb{C}})$ be some nested interpretation, α an M-action and β an O-action. The update of \mathfrak{J} with α is the nested interpretation

$$\mathfrak{J}^\alpha := (\mathbb{C}, \cdot^{\mathfrak{J}^\alpha}, \Delta, \{\mathcal{I}_c\}_{c \in \mathbb{C}}),$$

where $(\mathbb{C}, \cdot^{\mathfrak{J}^\alpha})$ is the update of $(\mathbb{C}, \cdot^{\mathfrak{J}})$ with $\mathcal{E}(\alpha, \mathfrak{J})$ and all other components are unchanged. The update of \mathfrak{J} with β is the nested interpretation

$$\mathfrak{J}^\beta := (\mathbb{C}, \cdot^{\mathfrak{J}}, \Delta, \{\mathcal{J}_c\}_{c \in \mathbb{C}}),$$

where for each $c \in \mathbb{C}$ the O-interpretation $\mathcal{J}_c := (\Delta, \cdot^{\mathcal{J}_c})$ is the update of \mathcal{I}_c with $\mathcal{E}(\beta, c, \mathfrak{J})$.

Let σ be a sequence of M-actions and O-actions the update \mathfrak{J}^σ is defined in the obvious way by induction on σ .

As in [1] the semantics of actions is defined independent from global constraints formulated as GCIs. Checking whether or not a given sequence of actions preserves certain GCIs is viewed as a reasoning task (called *projection problem*). Preserving global GCIs is not something we try to enforce in the semantics. For a discussion on this view on state constraints we refer to [9].

Note that it is possible to write an action that adds and deletes an object (or a pair of objects) to and from a name simultaneously. The semantics of updates

gives precedence to add effects but we want to exclude those descriptions. In the following we assume that for any M-action α , any O-action β and any nested interpretation \mathcal{I} and meta level domain element c the sets $\mathcal{E}(\alpha, \mathcal{I})$ and $\mathcal{E}(\beta, c, \mathcal{I})$ are non-contradictory.

We are interested in checking whether a certain consequence formulated as an m-KB holds after executing a sequence of actions given an incomplete representation of the initial state in terms of an m-KB.

Definition 8 (Projection Problem). *Let ψ, ψ' be m-KBs and σ a sequence of actions. We say that ψ' is a consequence of executing σ in ψ iff for all models \mathcal{I} of ψ , we have that $\mathcal{I}^\sigma \models \psi'$. The projection problem is then to decide whether ψ' is a consequence of executing σ in ψ .*

We continue our example about the conference management system.

Example 2 (Example 1 continued). An M-action for adding Alice as a PC Member of DL is given by

$$\text{add-pc} := \langle \text{DL-PC} \oplus \text{alice'-account} \rangle.$$

It updates a nested interpretation by adding `alice'-account` to `DL-PC` on the meta level and changing nothing else. Next, we want to define an action which assigns Alice as a reviewer for the submission `s` under the condition that she has no conflict of interest with an author of this submission. It is specified as an O-action:

$$\begin{aligned} \text{add-sub} := & (\{\text{alice'-account}\} \sqcap \neg \exists \text{has-conflict.} \llbracket \text{s} : \text{Own-Subs} \rrbracket) \triangleright \\ & \langle \text{Subs-To-Review} \oplus \text{s} \rangle \end{aligned}$$

Note that only the account of Alice is affected. The action only updates the object level interpretation associated with `alice'-account` by adding `s` to the review set (`Subs-To-Review`).

Let `r` be the name for the review Alice has written for `s`. We define an O-action that enters this review to the system and removes `s` from the review list of Alice simultaneously.

$$\begin{aligned} \text{finish} := & \text{enter} \parallel \text{remove}; \\ \text{enter} := & (\text{DL-PC} \sqcap \neg \exists \text{has-conflict.} \llbracket \text{s} : \text{Own-Subs} \rrbracket) \triangleright \langle \text{has-review} \oplus (\text{s}, \text{r}) \rangle; \\ \text{remove} := & \{\text{alice'-account}\} \triangleright \langle \text{Subs-To-Review} \ominus \text{s} \rangle. \end{aligned}$$

The review `r` is only visible for PC members with no conflict of interest with someone that is an author of `s`.

Assume initially we have axioms (1)-(5) from Example 1. After performing the sequence `add-pc; add-sub` it holds that

$$\text{alice'-account} : \llbracket \text{s} : \text{Subs-To-Review} \rrbracket$$

is true and the constraint

$$\neg \llbracket \text{Subs-To-Review} \sqsubseteq \perp \rrbracket \sqsubseteq \text{DL-PC}$$

is preserved. Furthermore, after `add-pc; add-sub; finish` we have that

$$\text{alice'-account} : \llbracket (s, r) : \text{has-review} \rrbracket$$

is true.

4 Deciding the Projection Problem in $\mathcal{ALCO} \llbracket \mathcal{ALCO} \rrbracket$

The approach of solving the projection problem in a DL-based action formalism by reducing it to a standard consistency problem in the underlying DL has been applied already in several settings (e.g. [5,4,13]). The overall idea we use here is similar to previous techniques extended to nested structures in our case.

As a first step we introduce a normal form of action descriptions by conjoining conditions and pushing them inside. We say that an N-action μ with $N \in \{M, O\}$ is in *normal form* if it is of the form

$$(\psi_1 \triangleright e_1) \parallel \dots \parallel (\psi_n \triangleright e_n),$$

where each e_i , for any i , $1 \leq i \leq n$ is an atomic N-action and ψ_i is either an m-KB (in case of an M-action) or an m-concept (in case of an O-action). We normalize an arbitrary N-action by applying exhaustively the following rules:

$$\psi_1 \triangleright (\psi_2 \triangleright \mu) \rightsquigarrow (\psi_1 \star \psi_2) \triangleright \mu \quad \psi \triangleright (\mu_1 \parallel \mu_2) \rightsquigarrow (\psi \triangleright \mu_1) \parallel (\psi \triangleright \mu_2),$$

where $\star \in \{\wedge, \sqcap\}$ stands for \wedge in case of an M-action and for \sqcap in case of an O-action. W.l.o.g., we assume from now on that any action is in normal form. For convenience, we denote a normal form of an N-action μ as a set of atomic N-actions with a single condition attached: $\mu = \{(\psi_1 \triangleright e_1), \dots, (\psi_n \triangleright e_n)\}$.

Let an m-KB ψ (initial state), ψ' (goal state) and a sequence of actions $\sigma = \mu_1, \dots, \mu_n$ in normal form be the input of the projection problem. Our goal is to construct a *reduction m-KB* that is consistent iff ψ' is a consequence of executing σ in ψ .

We say that concepts, roles, and individuals are relevant if they occur in the input of the projection problem. For the reduction we use fresh concept names and role names of the corresponding sort. For each execution step $0 \leq i \leq n$, we introduce fresh time-stamped copies $A^{(i)}$ of all relevant concept names, $r^{(i)}$ of all relevant role names, and fresh time-stamped concept names $T_C^{(i)}$ for every relevant complex subconcept C . $A^{(0)}$ refers to the initial content of A and the further copies $A^{(j)}$, $1 \leq j \leq n$ refer only to the set of named individual names of the corresponding sort that are instance of A after the j th execution step. This holds for both concept and role names. The copies of the form $T_C^{(i)}$ represent the content (both named and unnamed) of the complex concept C after the i th execution step. The distinction between named and unnamed is made because actions only affect named individuals.

Furthermore, for the set of all named individuals of sort object in the input (denoted by Obj_O) and for the set of all named meta level individuals in the input (Obj_M) two fresh concept names N_O and N_M , respectively, are introduced.

The meaning of the new names is now axiomatized using meta level axioms as follows. For $N_{\mathcal{O}}$ and $N_{\mathcal{M}}$ we have

$$\psi_{\text{obj}} = (N_{\mathcal{M}} \equiv \bigsqcup_{c \in \text{Obj}_{\mathcal{M}}} \{c\}) \wedge (\top \sqsubseteq \llbracket N_{\mathcal{O}} \equiv \bigsqcup_{a \in \text{Obj}_{\mathcal{O}}} \{a\} \rrbracket).$$

We use $\tau(C, i)$ to denote the concept definition we introduce to define the names of the form $T_C^{(i)}$. It is defined by induction on the structure of C as follows:

$$\begin{aligned} T_A^{(i)} &\equiv (N_{\mathcal{O}} \sqcap A^{(i)}) \sqcup (\neg N_{\mathcal{O}} \sqcap A^{(0)}) \\ T_{\{a\}}^{(i)} &\equiv \{a\} \quad T_{\neg C}^{(i)} \equiv \neg T_C^{(i)} \quad T_{C_1 \sqcap C_2}^{(i)} \equiv T_{C_1}^{(i)} \sqcap T_{C_2}^{(i)} \\ T_{\exists r, C}^{(i)} &\equiv (N_{\mathcal{O}} \sqcap ((\exists r^{(0)}. (\neg N_{\mathcal{O}} \sqcap T_C^{(i)})) \sqcup (\exists r^{(i)}. (N_{\mathcal{O}} \sqcap T_C^{(i)})))) \sqcup (\neg N_{\mathcal{O}} \sqcap \exists r^{(0)}. T_C^{(i)}) \end{aligned}$$

For referring meta concepts we have $T_{\llbracket \varphi \rrbracket}^{(i)} \equiv \llbracket \varphi^{(i)} \rrbracket$. For the object level, we ensure that the concept definitions hold in any context:

$$\psi_{\text{def}_{\mathcal{O}}}^{(i)} := \bigwedge_{C \in \mathcal{R}_{\mathcal{O}}} \top \sqsubseteq \llbracket \tau(C, i) \rrbracket \quad \psi_{\text{def}_{\mathcal{M}}}^{(i)} := \bigwedge_{G \in \mathcal{R}_{\mathcal{M}}} \tau(G, i),$$

where $\mathcal{R}_{\mathcal{O}}$ is the set of relevant object level subconcepts and $\mathcal{R}_{\mathcal{M}}$ the analogous set on the meta level. Given an O-GCI $\gamma = C \sqsubseteq D$ and a timestamp $0 \leq i \leq n$, we define the timestamped copy $\gamma^{(i)} := T_C^{(i)} \sqsubseteq T_D^{(i)}$, and $\varphi^{(i)}$ is the timestamped O-KB obtained from φ as the result of replacing every O-GCI γ in φ with $\gamma^{(i)}$. Timestamped copies for M-GCI $\zeta^{(i)}$ and m-KB $\psi^{(i)}$ are defined analogously.

We simply put timestamp zero for the initial knowledge base ψ , i.e., we include $\psi^{(0)}$ as a conjunct of the reduction m-KB.

Next, we encode the effects of each action μ_i using the m-KB ψ_{act_i} . Intuitively, we make sure that if the condition is satisfied, then corresponding atomic actions are applied to the next step. We define corresponding assertions for each atomic action e and a timestamp i with $\pi(e, i)$.

$$\begin{aligned} \pi(\langle A \oplus a \rangle, i) &:= (a : A^{(i)}) \quad \pi(\langle A \ominus a \rangle, i) := (a : \neg A^{(i)}) \\ \pi(\langle r \oplus (a, b) \rangle, i) &:= ((a, b) : r^{(i)}) \quad \pi(\langle r \ominus (a, b) \rangle, i) := ((a, b) : \neg r^{(i)}) \end{aligned}$$

We distinguish two cases, whether the action is an M-action or an O-action. First, we consider the case of μ_i is an M-action α_i . We define:

$$\psi_{\text{act}_{\mathcal{M}}}^{(i)} := \bigwedge_{\psi \triangleright e \in \alpha_i} (\psi^{(i-1)} \rightarrow \pi(e, i))$$

We encode the atomic O-actions similarly for $\mu_i = \beta_i$, with taking the context into account. Instead of having an m-KB, we have a timestamped m-concept as the condition. The atomic O-actions are propagated using referring meta concept for those contexts.

$$\psi_{\text{act}_{\mathcal{O}}}^{(i)} := \bigwedge_{G \triangleright e \in \beta_i} (T_G^{(i-1)} \sqsubseteq \llbracket \pi(e, i) \rrbracket)$$

In case of the other type of action happens at timestamp i , the corresponding $\psi_{\text{act}}^{(i)}$ is simply \top . For example, $\psi_{\text{act}_M}^{(i)} = \top$ if μ_i is an O-action.

Then, we make sure a change only happens if there is an effect that enforces it. For every i , $1 \leq i \leq n$, we define an m-KB $\psi_{\text{min}_M}^{(i)}$ that encodes all the non-effects, that is, for each named individual a , each relevant concept name A from the meta level it contains a conjunct of the form

$$\left((a : A^{(i-1)}) \wedge \bigwedge_{\psi \triangleright (A \ominus a) \in \mu_i} \neg \psi^{(i-1)} \right) \rightarrow a : A^{(i)} \wedge \\ (a : \neg A^{(i-1)}) \wedge \bigwedge_{\psi \triangleright (A \oplus a) \in \mu_i} \neg \psi^{(i-1)} \rightarrow a : \neg A^{(i)},$$

and analogous conjuncts for all relevant role names and pairs of named individuals. Likewise, we ensure a minimization of changes on the object level with an m-KB $\psi_{\text{min}_O}^{(i)}$ for every i , $1 \leq i \leq n$ consisting of conjuncts of the form

$$([\![o : B^{(i-1)}]\!] \wedge \bigwedge_{G \triangleright (B \ominus o) \in \mu_i} \neg T_G^{(i-1)}) \rightarrow [\![o : B^{(i)}]\!],$$

and analogously for the negative case and role names. Finally, we define the complete reduction m-KB:

$$\psi_{\text{red}} := \psi_{\text{init}} \wedge \psi_{\text{obj}} \wedge \bigwedge_{0 \leq i \leq n} \psi_{\text{def}_O}^{(i)} \wedge \bigwedge_{0 \leq i \leq n} \psi_{\text{def}_M}^{(i)} \wedge \\ \bigwedge_{1 \leq i \leq n} \psi_{\text{act}_O}^{(i)} \wedge \bigwedge_{1 \leq i \leq n} \psi_{\text{act}_M}^{(i)} \wedge \bigwedge_{1 \leq i \leq n} \psi_{\text{min}_O}^{(i)} \wedge \bigwedge_{1 \leq i \leq n} \psi_{\text{min}_M}^{(i)}$$

Lemma 1. *Let ψ be an m-KB, $\sigma = \mu_1, \dots, \mu_n$ be a sequence of actions, and ψ_{red} be defined as above. The following properties hold:*

1. *For every sequence of nested-interpretations $\mathfrak{I}_0, \dots, \mathfrak{I}_n$ such that $\mathfrak{I}_0 \models \psi$ and $\mathfrak{I}_i = \mathfrak{I}_{i-1}^{\mu_i}$ for each i , $1 \leq i \leq n$ there exists an interpretation $\mathfrak{L} \models \psi_{\text{red}}$ such that for every i , $0 \leq i \leq n$, and every relevant m-KB ψ , we have $\mathfrak{I}_i \models \psi$ iff $\mathfrak{L} \models \psi^{(i)}$.*
2. *For every nested-interpretation $\mathfrak{L} \models \psi_{\text{red}}$, there exists a sequence of nested-interpretations $\mathfrak{I}_0, \dots, \mathfrak{I}_n$ such that $\mathfrak{I}_0 \models \psi$ and $\mathfrak{I}_i = \mathfrak{I}_{i-1}^{\mu_i}$ for every i , $1 \leq i \leq n$ such that for every i , $0 \leq i \leq n$, and every relevant m-KB ψ , we have $\mathfrak{I}_i \models \psi$ iff $\mathfrak{L} \models \psi^{(i)}$.*

Lemma 1 shows the correspondence between the models of the reduction m-KB ψ_{red} and the sequence of nested interpretations we obtain by executing σ in a model of the initial m-KB. This allows one to use the following reduction of the projection problem to a (in)consistency problem in $\mathcal{ALCCO}[\mathcal{ALCCO}]$.

Lemma 2. *Let ψ, ψ' be m-KBs, and σ be a sequence of actions. It holds that ψ' is a consequence of executing σ in ψ iff $\psi_{\text{red}} \wedge \neg(\psi'^{(n)})$ is inconsistent.*

This gives us a complexity result for the projection problem.

Theorem 1. *The projection problem in $\mathcal{ALCO}[\mathcal{ALCO}]$ is EXPTIME-complete.*

Proof. ψ_{red} is polynomial in the size of the input, and obviously $\neg(\psi'^{(n)})$ as well. Since the consistency problem in $\mathcal{ALCO}[\mathcal{ALCO}]$ is EXPTIME-complete [8], we can use the following procedure: build $\psi_{\text{red}} \wedge \neg(\psi'^{(n)})$ as defined, and check using an $\mathcal{ALCO}[\mathcal{ALCO}]$ consistency checker that runs in exponential time. Hence, we get an EXPTIME procedure.

For hardness, we can reduce the consistency problem in $\mathcal{ALCO}[\mathcal{ALCO}]$ to the projection problem. It is easy to see that an m-KB ψ is consistent iff $\{a\} \sqsubseteq \top$ is a consequence of executing $\langle \rangle$ in ψ .

5 Conclusion

We have introduced an action formalism for reasoning about context and object level change in the ConDL $\mathcal{ALCO}[\mathcal{ALCO}]$. The formalism is well-behaved in the sense that the projection problem has the same complexity as standard reasoning in \mathcal{ALCO} .

From a practical point of view, choosing $\mathcal{ALCO}[\mathcal{ALCO}]$ has the advantage that an efficient reasoning tool for checking consistency already exists [7]. The reasoner even supports the more expressive combination $\mathcal{SHOIQ}[\mathcal{SHOIQ}]$.

For future work, we plan to investigate whether our action formalism offers sufficient expressiveness for capturing also the dynamic features of the role-based modelling language in [11].

Furthermore, we would like to study the complexity of reasoning in several extensions of the action formalism. This for example includes operators for non-determinism in the action dimension and temporal specifications for possibly infinite action sequences.

References

1. Ahmetaj, S., Calvanese, D., Ortiz, M., Simkus, M.: Managing change in graph-structured data using description logics. *ACM Trans. Comput. Log.* **18**(4), 27:1–27:35 (2017)
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press (2003)
3. Baader, F., Horrocks, I., Lutz, C., Sattler, U.: *An Introduction to Description Logic*. Cambridge University Press (2017)
4. Baader, F., Lippmann, M., Liu, H.: Using causal relationships to deal with the ramification problem in action formalisms based on description logics. In: Fermüller, C.G., Voronkov, A. (eds.) *Logic for Programming, Artificial Intelligence, and Reasoning - 17th International Conference, LPAR-17, Yogyakarta, Indonesia, October 10–15, 2010. Proceedings. Lecture Notes in Computer Science*, vol. 6397, pp. 82–96. Springer (2010). https://doi.org/10.1007/978-3-642-16242-8_7

5. Baader, F., Lutz, C., Milicic, M., Sattler, U., Wolter, F.: Integrating description logics and action formalisms: First results. In: AAI. pp. 572–577. AAI Press / The MIT Press (2005)
6. Bachman, C.W., Daya, M.: The role concept in data models. In: Proceedings of the Third International Conference on Very Large Data Bases, October 6-8, 1977, Tokyo, Japan. pp. 464–476. IEEE Computer Society (1977)
7. Böhme, S., Kühn, T.: Reasoning on context-dependent domain models. In: Wang, Z., Turhan, A., Wang, K., Zhang, X. (eds.) Semantic Technology - 7th Joint International Conference, JIST 2017, Gold Coast, QLD, Australia, November 10-12, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10675, pp. 69–85. Springer (2017), https://doi.org/10.1007/978-3-319-70682-5_5
8. Böhme, S., Lippmann, M.: Decidable description logics of context with rigid roles. In: Lutz, C., Ranise, S. (eds.) Frontiers of Combining Systems - 10th International Symposium, FroCoS 2015, Wroclaw, Poland, September 21-24, 2015. Proceedings. Lecture Notes in Computer Science, vol. 9322, pp. 17–32. Springer (2015), https://doi.org/10.1007/978-3-319-24246-0_2
9. Herzig, A., Varzinczak, I.J.: Metatheory of actions: Beyond consistency. *Artif. Intell.* **171**(16-17), 951–984 (2007)
10. Klarman, S., Gutiérrez-Basulto, V.: Description logics of context. *J. Log. Comput.* **26**(3), 817–854 (2016), <https://doi.org/10.1093/logcom/ext011>
11. Kühn, T., Böhme, S., Götz, S., Aßmann, U.: A combined formal model for relational context-dependent roles. In: Paige, R.F., Ruscio, D.D., Völter, M. (eds.) Proceedings of the 2015 ACM SIGPLAN International Conference on Software Language Engineering, SLE 2015, Pittsburgh, PA, USA, October 25-27, 2015. pp. 113–124. ACM (2015), <https://dl.acm.org/citation.cfm?id=2814255>
12. Schütze, L., Castrillón, J.: Analyzing state-of-the-art role-based programming languages. In: Sartor, J.B., D’Hondt, T., Meuter, W.D. (eds.) Companion to the first International Conference on the Art, Science and Engineering of Programming, Programming 2017, Brussels, Belgium, April 3-6, 2017. pp. 9:1–9:6. ACM (2017). <https://doi.org/10.1145/3079368.3079386>
13. Zariwies, B.: Complexity of projection with stochastic actions in a probabilistic description logic. In: Thielscher, M., Toni, F., Wolter, F. (eds.) Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018, Tempe, Arizona, 30 October - 2 November 2018. pp. 514–523. AAI Press (2018), <https://aaai.org/ocs/index.php/KR/KR18/paper/view/18050>