Planning in Action Formalisms based on DLs: First Results

Maja Miličić

Institut für Theoretische Informatik
TU Dresden, Germany
maja@tcs.inf.tu-dresden.de

Abstract. In this paper, we continue the recently started work on integrating action formalisms with description logics (DLs), by investigating planning in the context of DLs. We prove that the plan existence problem is decidable for actions described in fragments of $\mathcal{ALCQIO}$. More precisely, we show that its computational complexity coincides with the one of projection for DLs between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$.

1 Introduction

The idea to investigate action formalisms based on description logics was inspired by the expressivity gap between existing action formalisms: they were either based on FO logic and undecidable, like the Situation Calculus [12] and the Fluent Calculus [14], or decidable but only propositional.

First results on integrating DLs with action formalisms from [2] show that reasoning remains decidable even if an action formalism is based on the expressive DL $\mathcal{ALCQIO}$. In [2], ABox assertions are used for describing the current state of the world, and the pre- and post-conditions of actions. Domain constraints are captured by *acyclic* TBoxes, and post-conditions may contain only atomic concept and role assertions. It is shown in [2] that the projection and executability problem for actions can be reduced to standard DL reasoning problems. Further papers in this line [11, 10] treat the problem of computing ABox updates and the ramification problem induced by GCIs.

However, in the mentioned DL-action-framework, planning, an important reasoning task, has not yet been considered. Intuitively, given an initial state $\mathcal{A}$, final state Γ and a final set of actions Op, the *plan existence problem* is the following: "is there a plan (a sequence of actions from Op) which transforms $\mathcal{A}$ into Γ?". It is known that, already in the propositional case, planning is a hard problem. For example, the plan existence problem for propositional STRIPS-style actions is PSPACE-complete [5, 7].

The planning problem in DL action formalisms is not only interesting from the theoretical point of view. It is well known that web ontology languages for the Semantic Web are based on description logics; thus actions described in DLs can be viewed as simple semantic web services. In this context, planning is a very important reasoning task as it supports, e.g., web service discovery which is needed for an automatic service execution.

This paper is, to our best knowledge, the first try to formally define the planning problem in the context of description logics. It is based on the action formalism from [2]. We investigate the computational complexity of the plan existence problem for the description logics "between" $\mathcal{ALC}$ and $\mathcal{ALCQIO}$. We show that, in these logics, the plan existence problem is decidable, and of the same computational complexity as projection. In the last section we discuss possible ways of developing practical planning algorithms for DLs.

2 Preliminaries

In this paper we will use a slightly modified version of the action formalism from [2]. We disallow occlusions, a source of a limited non-determinism in [2]. Moreover, we introduce parameterised actions (operators). The formalism is not restricted to a particular DL, but for our complexity results we will consider the DL $\mathcal{ALCQIO}$ and its fragments. We refrain from introducing the syntax and semantics of $\mathcal{ALCQIO}$ in full detail, referring instead to [1].

We give only the definition of ABoxes, as it slightly differs from the one from [1]. An *ABox assertion* is of the form $C(a)$, $r(a, b)$, or $\neg r(a, b)$ where a, b are individual names, C is a concept, and r a role name. An *ABox* is a finite set of ABox assertions.

The main ingredients of our framework are operators and actions (as defined below), ABoxes for describing the current knowledge about the state of affairs in the application domain, and acyclic TBoxes for describing general knowledge about the application domain similar to state constraints in the SitCalc and Fluent Calculus.

Definition 1 (Action, operator). *Let* $\mathsf{N_X}$ *and* $\mathsf{N_I}$ *be disjoint and countably infinite sets of* variables *and* individual names. *Moreover, let* $\mathcal{T}$ *be an acyclic TBox. A* primitive literal for $\mathcal{T}$ *is an ABox assertion*

$$A(a), \neg A(a), r(a, b), \text{ or } \neg r(a, b)$$

with A *a primitive concept name in* $\mathcal{T}$*,* r *a role name, and* $a, b \in \mathsf{N_I}$*. An* atomic atomic $\alpha = (\mathsf{pre}, \mathsf{post})$ *for* $\mathcal{T}$ *consists of*

- *a finite set* pre *of ABox assertions, the* pre-conditions*;*
- *a finite set* post *of* conditional post-conditions *of the form* φ/ψ*, where* φ *is an ABox assertion and* ψ *is a* primitive literal for $\mathcal{T}$*.*

A composite action *for* $\mathcal{T}$ *is a finite sequence* $\alpha_1, \ldots, \alpha_k$ *of atomic actions for* $\mathcal{T}$*.*

An operator for $\mathcal{T}$ *is a parametrised atomic action for* $\mathcal{T}$*, i.e., an action in which definition variables from* $\mathsf{N_X}$ *may occur in place of individual names.*

We call post-conditions of the form $\top(t)/\psi$ *unconditional* and write just ψ instead.

Applying an action changes the state of affairs, and thus transforms an interpretation $\mathcal{I}$ into an interpretation $\mathcal{I}'$. Intuitively, the pre-conditions specify under which conditions the action is applicable. The post-condition φ/ψ says that, if φ is true in the original interpretation $\mathcal{I}$, then ψ is true in the interpretation $\mathcal{I}'$ obtained by applying the action.

Definition 2. *Let $\mathcal{T}$ be an acyclic TBox, $\alpha = (\mathsf{pre}, \mathsf{post})$ an atomic action for $\mathcal{T}$, and $\mathcal{I}, \mathcal{I}'$ models of $\mathcal{T}$ respecting the unique name assumption (UNA) and sharing the same domain and interpretation of all individual names. We say that α* may transform *$\mathcal{I}$ to $\mathcal{I}'$ ($\mathcal{I} \Rightarrow_{\alpha}^{\mathcal{T}} \mathcal{I}'$) iff, for each primitive concept A and role name r, we have*

$$\begin{aligned} A^{\mathcal{I}'} &:= (A^{\mathcal{I}} \cup \{a^{\mathcal{I}} \mid \varphi/A(a) \in \mathsf{post} \textit{ and } \mathcal{I} \models \varphi\}) \setminus \\ &\qquad\qquad \{a^{\mathcal{I}} \mid \varphi/\neg A(a) \in \mathsf{post} \textit{ and } \mathcal{I} \models \varphi\} \\ r^{\mathcal{I}'} &:= (r^{\mathcal{I}} \cup \{(a^{\mathcal{I}}, b^{\mathcal{I}}) \mid \varphi/r(a,b) \in \mathsf{post} \textit{ and } \mathcal{I} \models \varphi\}) \setminus \\ &\qquad\qquad \{(a^{\mathcal{I}}, b^{\mathcal{I}}) \mid \varphi/\neg r(a,b) \in \mathsf{post} \textit{ and } \mathcal{I} \models \varphi\}. \end{aligned}$$

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

Note that this definition does not check whether the action is indeed executable, i.e., whether the pre-conditions are satisfied. It just says what the result of applying the action is, irrespective of whether it is executable or not. Since we use acyclic TBoxes to describe background knowledge, there cannot exist more than one $\mathcal{I}'$ such that $\mathcal{I} \Rightarrow_{\alpha}^{\mathcal{T}} \mathcal{I}'$. Thus, actions are deterministic.

Like in [2], we assume that actions $\alpha = (\mathsf{pre}, \mathsf{post})$ are *consistent with $\mathcal{T}$* in the following sense: for every model $\mathcal{I}$ of $\mathcal{T}$, there exists $\mathcal{I}'$, such that $\mathcal{I} \Rightarrow_{\alpha}^{\mathcal{T}} \mathcal{I}'$. It is not difficult to see that this is the case iff $\{\varphi_1/\psi, \varphi_2/\neg\psi\} \subseteq \mathsf{post}$ implies that the ABox $\{\varphi_1, \varphi_2\}$ is inconsistent w.r.t. $\mathcal{T}$.

Two standard reasoning problems about actions, *projection* and *executability*, are thoroughly investigated in [2] in the context of DLs. Executability is the problem of whether an action can be applied in a given situation, i.e. if pre-conditions are satisfied in the states of the world considered possible.

Formally, let $\mathcal{T}$ be an acyclic TBox, $\mathcal{A}$ an ABox, and let $\alpha_1, \ldots, \alpha_n$ be a composite action with $\alpha_i = (\mathsf{pre}_i, \mathsf{post}_i)$ atomic actions for $\mathcal{T}$ for $i = 1, \ldots, n$.

We say that $\alpha_1, \ldots, \alpha_n$ 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 $\mathcal{I} \Rightarrow_{\alpha_1,\ldots,\alpha_i}^{\mathcal{T}} \mathcal{I}'$, we have $\mathcal{I}' \models \mathsf{pre}_{i+1}$.

Projection is the problem of whether applying an action achieves the desired effect, i.e., whether an assertion that we want to make true really holds after executing the action. Formally,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 $\mathcal{I} \Rightarrow_{\alpha_1,\ldots,\alpha_n}^{\mathcal{T}} \mathcal{I}'$, we have $\mathcal{I}' \models \varphi$.

In [2] it was shown that projection and executability are decidable for the logics between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$. More precisely, projection in $\mathcal{L}$ can be reduced to (in)consistency of an ABox relative to an acyclic TBox in $\mathcal{LO}$. The following theorem from [2] states that upper complexity bounds obtained in this way are optimal:

Theorem 1. *([2]) Projection and executability of composite actions are:*

(a) PSpace-*complete in* $\mathcal{ALC}$,$\mathcal{ALCO}$,$\mathcal{ALCQ}$, *and* $\mathcal{ALCQO}$;
(b) ExpTime-*complete in* $\mathcal{ALCI}$ *and* $\mathcal{ALCIO}$;
(c) *co*-NExpTime-*complete in* $\mathcal{ALCQI}$ *and* $\mathcal{ALCQIO}$.

Looking carefully at the reduction of projection in $\mathcal{L}$ to ABox inconsistency in $\mathcal{LO}$ from [2, 3], we conclude that the upper complexity bounds from Theorem 1 hold even for the "stronger" projection problem, namely the one where the assertion φ is a disjunction of ABox assertions. We will need this strengthened complexity result in the coming sections.

3 Planning problem

We continue by defining the plan existence problem in our framework. As in the previous section, we do not fix the DL, but assume it to be a sublogic of $\mathcal{ALCQIO}$.

First we introduce a bit of notation. If o is an operator (for a TBox $\mathcal{T}$), we use $\mathsf{var}(o)$ to denote the set of variables in o. A substitution v for o is a mapping $v : \mathsf{var}(o) \to \mathsf{N_I}$. An action α that is obtained by applying a substitution v to o is denoted as $\alpha := o[v]$. Intuitively, the plan existence problem is: given an acyclic TBox $\mathcal{T}$ which describes the background knowledge, ABoxes $\mathcal{A}$ and Γ describing respectively the initial and the goal state, and a set of operators Op, is there a plan (sequence of actions obtained by instantiating operators from Op) which "transforms" $\mathcal{A}$ into Γ?

In this paper, we assume that operators can be instantiated with individuals from a finite set $\mathsf{Ind} \subset \mathsf{N_I}$. Moreover, we assume that $\mathcal{T}$, $\mathcal{A}$ and Γ contain only individuals from Ind (we say that they are based on Ind). For an operator o, we set $o[Ind] := \{o[v] \mid v : \mathsf{var}(o) \to \mathsf{Ind}\}$ and for Op a set of operators, we set $\mathsf{Op}[Ind] := \{o[Ind] \mid o \in \mathsf{Op}\}$. In the following definition, we formally introduce the notion of a planing task:

Definition 3 (Planning task). *A planning task is a tuple* $\Pi = (\mathsf{Ind}, \mathcal{T}, \mathsf{Op}, \mathcal{A}, \Gamma)$, *where*

- Ind *is a finite set of individual names;*
- $\mathcal{T}$ *is an acyclic TBox based on* Ind*;*
- Op *is a finite set of atomic operators for* $\mathcal{T}$*;*
- $\mathcal{A}$ *(initial state) is an ABox based on* Ind*;*
- Γ *(goal) is an ABox based on* Ind.

A plan *in Π is a composite action $\alpha = \alpha_1, \ldots, \alpha_k$, such that $\alpha_i \in \mathsf{Op}[\mathsf{Ind}]$, $i = 1..k$. A plan $\alpha = \alpha_1, \ldots, \alpha_k$ in Π is a* solution *to the planning task Π iff:*

1. *α is executable in $\mathcal{A}$ w.r.t. $\mathcal{T}$; and*
2. *for all interpretations $\mathcal{I}$ and $\mathcal{I}'$ such that $\mathcal{I} \models \mathcal{A}, \mathcal{T}$ and $\mathcal{I} \Rightarrow_\alpha^\mathcal{T} \mathcal{I}'$, it holds that $\mathcal{I}' \models \Gamma$.*

Two common planning problems, PLANEX and PLANLEN, c.f. [7], are defined below:

Definition 4 (Planning problems). Plan existence problem (PLANEX)*: Does a given planning task Π have a solution?*

Bounded plan existence problem (PLANLEN)*: For a planning task Π and a natural number n, is there a plan of length at most 2^n which is a solution to Π?*

4 Complexity of planning

In this section, we will present a decision procedure for the plan existence problem. It turns out that PLANEX is not more difficult, at least in theory, than projection in the DLs from Theorem 1.

In what follows, for the sake of simplicity we assume that $\mathcal{T} = \emptyset$. It is not difficult to show that the complexity results form this section hold in the case of non-empty acyclic TBoxes.

Obviously, the plan existence problem is closely related to projection and executability. First we introduce some notation. Let $\mathcal{A}$ be an ABox, α a (possibly composite) action, and φ an ABox assertion or a disjunction of ABox assertions. We will write $\mathcal{A}^\alpha \models \varphi$ iff φ is a consequence of applying α in $\mathcal{A}$. For an ABox $\mathcal{B}$, we write $\mathcal{A}^\alpha \models \mathcal{B}$ iff $\mathcal{A}^\alpha \models \varphi$ for all $\varphi \in \mathcal{B}$.

Let $\Pi = (\mathsf{Ind}, \emptyset, \mathsf{Op}, \mathcal{A}, \Gamma)$ be a planning task. for which we want to decide if it has a solution. This means that we want to check if there is a sequence of actions from $\mathsf{Op}[\mathsf{Ind}]$ which transform the initial state (described by $\mathcal{A}$) into the goal state (described by Γ). In the propositional case, planning is based on step-wise computation of the next state – which corresponds to computing updated ABoxes. However, in [11], it is shown that an updated ABox may be exponentially large in the size of the initial ABox and the update, which makes this approach unsuitable. We base our approach in this paper on the following observation: possible worlds obtained by applying (composite) actions in the initial world $\mathcal{A}$ can be implicitly described by $\mathcal{A}$ together with the list of applied atomic changes (intuitively, this is an accumulated list of the triggered post-conditions).

We define the set of possible (negated) atomic changes as:

$$\mathcal{L} := \{\psi, \neg\psi \mid \varphi/\psi \in \alpha, \alpha \in \mathsf{Op}[\mathsf{Ind}]\}$$

An *update* for Π is a consistent subset of $\mathcal{L}$. Let $\mathfrak{U}$ be a set of all updates for Π. Then $\mathfrak{U}$ is our search space, the size of which $|\mathfrak{U}|$ is exponential in the size of $|\mathcal{L}|$ (and Π). For a $\mathcal{U} \in \mathfrak{U}$, we set $\neg\mathcal{U} := \{\neg l \mid l \in \mathcal{U}\}$. Intuitively, $\mathcal{U}_0 := \{l \in \mathcal{L} \mid \mathcal{A} \models l\}$ represents the initial state, and all updates $\mathcal{U} \in \mathfrak{U}$ such that $\mathcal{A}^{\mathcal{U}} \models \Gamma$ represent goals states[1].

In the next step, we define the transition relation "$\xrightarrow{\alpha}_{\mathcal{A}}$" between updates. Let $\mathcal{U}$ and $\mathcal{V}$ be two updates. For $\alpha = (\mathsf{pre}, \mathsf{post})$, we say that $\mathcal{U} \xrightarrow{\alpha}_{\mathcal{A}} \mathcal{V}$ iff:

(i) $\mathcal{A}^{\mathcal{U}} \models \mathsf{pre}$
(ii) $\mathcal{V} = (\mathcal{U} \setminus \neg\mathsf{post}^{\mathcal{U}}_{\alpha}) \cup \mathsf{post}^{\mathcal{U}}_{\alpha}$, where $\mathsf{post}^{\mathcal{U}}_{\alpha} = \{\psi \mid \mathcal{A}^{\mathcal{U}} \models \bigvee_{\varphi/\psi \in \mathsf{post}} \varphi\}$

Obviously, the relation "$\xrightarrow{\alpha}_{\mathcal{A}}$" is a partial function for every α. In the following lemma, we show that "$\xrightarrow{\alpha}_{\mathcal{A}}$" simulates "$\Rightarrow_\alpha$"[2] on the level of updates.

Lemma 1. *Let $\mathcal{A}$ be an ABox, and $\alpha = \alpha_1, \dots, \alpha_k$ a composite action, with $\alpha_i = (\mathsf{pre}_i, \mathsf{post}_i) \in \mathsf{Op}[\mathsf{Ind}]$. Let $\mathcal{U}_0 := \{l \in \mathcal{L} \mid \mathcal{A} \models l\}$. Then the following holds:*

(a) There exist unique $\mathcal{U}_1, \dots, \mathcal{U}_k$ such that $\mathcal{U}_0 \xrightarrow{\alpha_1}_{\mathcal{A}} \mathcal{U}_1 \cdots \xrightarrow{\alpha_k}_{\mathcal{A}} \mathcal{U}_k$ iff $\alpha_1, \dots, \alpha_k$ is executable in $\mathcal{A}$;
(b) Let $\mathcal{U}_k$ be defined as in (a). Then for all interpretations $\mathcal{I}, \mathcal{I}'$ such that $\mathcal{I} \models \mathcal{A}$, we have that $\mathcal{I} \Rightarrow_{\alpha_1,\dots,\alpha_k} \mathcal{I}'$ iff $\mathcal{I} \Rightarrow_{\mathcal{U}_k} \mathcal{I}'$.

Proof. Proof by induction on k. For $k = 0$, trivially true. Assume that the claim holds for $k = m$, and let us prove that it implies the same for $k = m + 1$. (a) follows directly from the point (i) of the definition of $\xrightarrow{\alpha_{m+1}}_{\mathcal{A}}$. As for (b), let $\mathcal{I} \models \mathcal{A}$ and let $\mathcal{I} \Rightarrow_{\alpha_1,\dots,\alpha_{m+1}} \mathcal{I}'$. The latter holds iff there exists $\mathcal{I}''$ such that $\mathcal{I} \Rightarrow_{\alpha_1,\dots,\alpha_m} \mathcal{I}''$ and $\mathcal{I}'' \Rightarrow_{\alpha_{m+1}} \mathcal{I}'$. By I.H., we have that for $\mathcal{I} \models \mathcal{A}$ it holds that $\mathcal{I} \Rightarrow_{\alpha_1,\dots,\alpha_m} \mathcal{I}''$ iff $\mathcal{I} \Rightarrow_{\mathcal{U}_m} \mathcal{I}''$. Finally, the point (ii) of the definition of $\xrightarrow{\alpha_{m+1}}_{\mathcal{A}}$ implies that there exists $\mathcal{I}''$ such that $\mathcal{I} \Rightarrow_{\mathcal{U}_m} \mathcal{I}''$ and $\mathcal{I}'' \Rightarrow_{\alpha_{m+1}} \mathcal{I}'$ iff $\mathcal{I} \Rightarrow_{\mathcal{U}_m} \mathcal{I}''$ and $\mathcal{I}'' \Rightarrow_{\mathsf{post}^{\mathcal{U}_m}_{\alpha_{m+1}}} \mathcal{I}'$. It is not difficult to see that the latter holds iff $\mathcal{I} \Rightarrow_{\mathcal{U}_{m+1}} \mathcal{I}'$.

We now present a procedure which decides if a state $\mathcal{V} \in \mathfrak{U}$ is reachable from $\mathcal{U} \in \mathfrak{U}$ by executing a sequence of actions from $\mathsf{Op}[\mathsf{Ind}]$ (an adaption of the reachability algorithm from [13]). Since the search space $\mathfrak{U}$ is of size $3^{\frac{|\mathcal{L}|}{2}}$ $(< 2^{|\mathcal{L}|})$, there is no need to check for the existence of longer paths.

reachable($\Pi, \mathcal{U}, \mathcal{V}$)
    if **path**($\Pi, \mathcal{U}, \mathcal{V}, |\mathcal{L}|$)
        then return TRUE
        return FALSE

path($\Pi, \mathcal{U}, \mathcal{V}, i$) checks if $\mathcal{V}$ is reachable from $\mathcal{U}$ by a path of length at most 2^i:

[1] Starting from here, we will sometimes write $\mathcal{U}$ as short for the action $(\emptyset, \mathcal{U})$. Please note that $\mathcal{A}^{\mathcal{U}} \models \varphi$ is only an abbreviation for "φ is a consequence of applying $(\emptyset, \mathcal{U})$ in $\mathcal{A}$", and does not imply computing the update of the ABox $\mathcal{A}$ with $\mathcal{U}$ as in [11].

[2] $\Rightarrow_\alpha$ is short for $\Rightarrow^{\emptyset}_{\alpha}$

path(Π,$\mathcal{U}$, $\mathcal{V}$, i)
  if ($i = 0$ and ($\mathcal{U} = \mathcal{V}$ or **one_step**(Π,$\mathcal{U}$, $\mathcal{V}$)))
    then return TRUE
  for all ($\mathcal{W} \in \mathfrak{U}$)
    if (**path**(Π,$\mathcal{U}$, $\mathcal{W}$, $i-1$) and **path**(Π, $\mathcal{W}$, $\mathcal{V}$, $i-1$))
      then return TRUE
  return FALSE

The predicate **one_step**(Π,$\mathcal{U}$, $\mathcal{V}$) checks if $\mathcal{V}$ can be reached from $\mathcal{U}$ in exactly one step by applying an action $\alpha \in \mathsf{Op}[\mathsf{Ind}]$.

one_step(Π,$\mathcal{U}$, $\mathcal{V}$)
  for all $\alpha \in \mathsf{Op}[\mathsf{Ind}]$
    if ($\mathcal{U} \xrightarrow{\alpha}_{\mathcal{A}} \mathcal{V}$)
      then return TRUE;
  return FALSE;

Lemma 2. *Let $\Pi = (\mathsf{Ind}, \mathcal{T}, \mathsf{Op}, \mathcal{A}, \Gamma)$ be a planning task and let $\mathcal{U}_0 := \{l \in \mathcal{L} \mid \mathcal{A} \models l\}$. Then Π has a solution iff there exists an $\mathcal{U}_\Gamma \in \mathfrak{U}$ such that $\mathcal{A}^{\mathcal{U}_\Gamma} \models \Gamma$ and* **reachable**(Π,$\mathcal{U}_0$,$\mathcal{U}_\Gamma$) *returns TRUE.*

Proof. "$\Rightarrow$" Let the plan $\alpha_1, \dots, \alpha_k$ be a solution to Π such that $k < 2^{|\mathcal{L}|}$. This means that (i)$\alpha_1, \dots, \alpha_k$ is executable w.r.t. $\mathcal{A}$ and (ii) $\mathcal{A}^{\alpha_1,\dots,\alpha_k} \models \Gamma$. By Lemma 1 (a), there exist unique $\mathcal{U}_i$, $1 \leq i \leq k$, such that $\mathcal{U}_0 \xrightarrow{\alpha_1}_{\mathcal{A}} \mathcal{U}_1 \cdots \xrightarrow{\alpha_k}_{\mathcal{A}} \mathcal{U}_k$. Thus, **reachable**($\Pi$,$\mathcal{U}_0$,$\mathcal{U}_k$) returns TRUE. Let $\mathcal{U}_\Gamma = \mathcal{U}_k$. By Lemma 1 (b), we have that $\mathcal{A}^{\alpha_1,\dots,\alpha_k} \models \Gamma$ implies $(\mathcal{A}^{\mathcal{U}_\Gamma} =)\mathcal{A}^{\mathcal{U}_k} \models \Gamma$.

"$\Leftarrow$" Let $\mathcal{U}_\Gamma \in \mathfrak{U}$ be such that $\mathcal{A}^{\mathcal{U}_\Gamma} \models \Gamma$ and **reachable**(Π,$\mathcal{U}_0$,$\mathcal{U}_\Gamma$) returns TRUE. Then there exists a sequence of actions $\alpha_1, \dots, \alpha_k$ such that $\mathcal{U}_0 \xrightarrow{\alpha_1}_{\mathcal{A}} \mathcal{U}_1 \cdots \xrightarrow{\alpha_k}_{\mathcal{A}} \mathcal{U}_k (= \mathcal{U}_\Gamma)$. By Lemma 1, we have that $\alpha_1, \dots, \alpha_k$ is executable w.r.t. $\mathcal{A}$ and $\mathcal{A}^{\alpha_1,\dots,\alpha_k} \models \Gamma$. Thus, $\alpha_1, \dots, \alpha_k$ is a solution to Π.

The previous lemma tells us that the plan existence problem can be decided by checking if **reachable**($\mathcal{U}_0$,$\mathcal{U}_F$, Π) returns TRUE for some final state $\mathcal{U}_F \in \mathfrak{U}$. If **one_step** could be decided in a constant time, **reachable** would require PSpace. However, **one_step** relies on polynomially many projection calls, which means that both **one_step** and **reachable** belong to the same complexity class as projection in DLs from Theorem 1, i.e. they are in PSpace (ExpTime, co-NExpTime) if projection is in PSpace (ExpTime, co-NExpTime).

(i) If projection is in PSpace, then we can decide the plan existence problem in PSpace: obviously, guessing a state (update) $\mathcal{U}_F$ from $\mathfrak{U}$, and checking whether $\mathcal{A}^{\mathcal{U}_F} \models \Gamma$ and **reachable**($\mathcal{U}_0$,$\mathcal{U}_F$, Π) returns TRUE, can be done in NPSpace. Finally, we use the result by Savitch [13] that PSpace = NPSpace.
(ii) If projection is in ExpTime (co-NExpTime), then we can check for all $\mathcal{U}_F$ from $\mathfrak{U}$, if $\mathcal{A}^{\mathcal{U}_F} \models \Gamma$ and **reachable**($\mathcal{U}_0$,$\mathcal{U}_F$, Π) returns TRUE. Since $\mathfrak{U}$ is

exponentially big in the size of Π, plan existence problem can thus be decided in ExpTime (co-NExpTime).

We obtained the following lemma:

Lemma 3. *Let $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCO}, \mathcal{ALCI}, \mathcal{ALCQ}, \mathcal{ALCIO}, \mathcal{ALCQO}, \mathcal{ALCQI}, \mathcal{ALCQIO}\}$. The plan existence problem in $\mathcal{L}$ has the same upper complexity bound as projection in $\mathcal{L}$.*

We show that the upper complexity bounds established in Lemma 1 are tight by the following easy reduction of projection to PLANEX. Let $\mathcal{A}$ be an ABox, α an action without pre-conditions and only with unconditional post-conditions, and φ an assertion.We define the planning task $\Gamma_{\mathcal{A},\alpha,\varphi}$ as $\Gamma_{\mathcal{A},\alpha,\varphi} := (\emptyset, \emptyset, \{\alpha\}, \mathcal{A}, \{\varphi\})$. It is not difficult to see that $\mathcal{A}^\alpha \models \varphi$ iff $\Gamma_{\mathcal{A},\alpha,\varphi}$ has a solution.

Since the lower bounds for projection from Theorem 1 hold already in the case of the empty TBox and an atomic action with no pre-conditions and no occlusions and only with unconditional post-conditions [2], we conclude that the complexity bounds from Lemma 1 are optimal, i.e., plan existence problem is of exactly the same computational complexity as projection. Moreover, the afore presented reduction implies that the same hardness results hold for the the bounded plan existence problem.

Theorem 2. *The planning problems PLANEX and PLANLEN are:*

(a) PSpace*-complete in $\mathcal{ALC}$, $\mathcal{ALCO}$, $\mathcal{ALCQ}$, and $\mathcal{ALCQO}$;*
(b) ExpTime*-complete in $\mathcal{ALCI}$ and $\mathcal{ALCIO}$;*
*(c) co-*NExpTime*-complete in $\mathcal{ALCQI}$ and $\mathcal{ALCQIO}$.*

5 Extended Planning

The previous decidability and complexity results are obtained under assumption that the set of individuals Ind used to instantiate operators is finite and a part of the input. This assumption is rather natural and in the line with the standard definitions of planning tasks for STRIPS operators from [5, 7]. Intuitively, Ind is a set of individuals the planning agent has control over.

Alternatively, one can omit individuals from the input and define a planning task Π as $\Pi = (\mathcal{T}, \mathsf{Op}, \mathcal{A}, \Gamma)$. The *extended plan existence problem* is the one of whether there is a solution for Π, defining a plan for Π to be a sequence of actions $\alpha_1, \ldots, \alpha_k$, where each α_i is obtained by instantiating an operator from Op with individuals from an infinitely countable set $\mathsf{N_I}$.

The extended planning raises new interesting questions:

Q1 In order to solve Π, do we have to use infinitely many individuals?
Q2 If the number of needed individuals can be shown to be bounded by $f(|\Pi|)$, is f a polynomial (exponential, double-exponential,...) function?

In the case of the datalog STRIPS, it is shown that the extended plan existence problem is undecidable [7, 6]. However, this undecidabilty result does not automatically carry over to the action formalism used in this paper. Indeed, the undecidability result from [7, 6] relies on the closed world assumption and negative pre-conditions. By using these two, one can define operators which are applicable only if instantiated with "unused" individuals. Such operators would have $\neg\mathsf{Used}(x)$ among its pre-conditions, and $\mathsf{Used}(x)$ in the list of post-conditions. Like this, one can enforce a usage of infinitely many individuals.

In the case of DLs considered in the previous sections, due to the open world assumption, it is not possible to state that all individuals not appearing in the initial ABox are instances of the concept $\neg\mathsf{Used}$.

However, in the presence of the universal role U, we can make assertions over the whole domain. For example, the assertion $\forall U.\neg\mathsf{Used}(a)$ can ensure that all element domains are unused in the initial state. We will show that extended planning in $\mathcal{ALC}_U$ (extension of $\mathcal{ALC}$ with the universal role) is undecidable. Undecidability us shown by reducing the halting problem of a deterministic Turing machine to the extended plan existence problem, similar to [6].

Let $M = (Q, \Sigma, \delta, q_0, q_f)$ be a deterministic Turing machine, where

- $Q = \{q_0, \ldots, q_n\}$ a finite set of states;
- $\Sigma = \{\mathsf{blank}, a_1, \ldots, a_m\}$ a finite alphabet;
- $\delta : Q \times \Sigma \rightarrow Q \times \Sigma \times \{L, R\}$ is a transition function;
- q_0 is the initial state;
- $q_f \in Q$ is the final state.

Let $a = a_{i_0}, \ldots a_{i_k} \in \Sigma^*$ be an input word. We will define a planning task $\Pi_{M,a} = (\emptyset, \mathsf{Op}_{M,a}, \mathcal{A}_{M,a}, \Gamma_{M,a})$ such that a planner for Π simulates moves of the Turing Machine M.

In the reduction, we use concept names $Q_0, \ldots, Q_n$, Blank, $A_1, \ldots, A_m$, Used, Last, M, Done, and a role name right. We define the initial state $\mathcal{A}_{M,a}$, the goal $\Gamma_{M,a}$, and the set of operators $\mathsf{Op}_{M,a}$ as:

$$\begin{aligned}
\mathcal{A}_{M,a} &:= \{(M \sqcap \forall U.\neg\mathsf{Used})(t_0)\} \cup \{A_{i_0}(t_0), \ldots, A_{i_k}(t_k)\} \\
&\quad \cup \{\mathsf{right}(t_0, t_1), \ldots \mathsf{right}(t_{k-1}, t_k)\} \\
\Gamma_{M,a} &:= \{\mathsf{Done}(t_0)\} \\
\mathsf{Op}_{M,a} &:= \{\mathsf{start}, \mathsf{create_succ}(x.y), \mathsf{done}(x), \mathsf{done_to_left}(x, y)\} \cup \\
&\qquad \bigcup_{\delta(q,a)=(q',b,R)} \{\mathsf{right}_{q,a,q',b}(x, y)\} \cup \bigcup_{\delta(q,a)=(q',b,L)} \{\mathsf{left}_{q,a,q',b}(x, y)\}
\end{aligned}$$

where the single operators are defined as follows:

$$\begin{aligned}
\mathsf{start} &:= (\{M(t_0)\}, \{\mathsf{Used}(t_0), \ldots, \mathsf{Used}(t_k), \mathsf{Last}(t_k), \neg M(t_0), Q_0(t_0)\}) \\
\mathsf{create_succ}(x.y) &:= (\{\mathsf{Last}(x), \neg\mathsf{Used}(y)\}, \\
&\qquad \{\mathsf{right}(x, y), \neg\mathsf{Last}(x), \mathsf{Last}(y), \mathsf{Used}(y), \mathsf{Blank}(y)\}
\end{aligned}$$

$$\mathsf{right}_{q,a,q',b}(x,y) := (\{Q(a), A(x), \mathsf{right}(x,y)\}, \{\neg Q(x), \neg A(x), B(x), Q'(y)\}$$
$$\mathsf{left}_{q,a,q',b}(x,y) := (\{Q(a), A(x), \mathsf{right}(y,x)\}, \{\neg Q(x), \neg A(x), B(x), Q'(y)\}$$
$$\mathsf{done}(x) := (\{Q_f(x)\}, \{\mathsf{Done}(x)\})$$
$$\mathsf{done\_to\_left}(x,y) := (\{\mathsf{Done}(x), \mathsf{right}(y,x)\}, \{\mathsf{Done}(y)\})$$

It is not difficult to show that the following lemma holds:

Lemma 4. *The Turing machine M halts for the input a iff there is a solution to the planning task $\Pi_{M,a} = (\emptyset, \mathsf{Op}_{M,a}, \mathcal{A}_{M,a}, \Gamma_{M,a})$.*

Thus, we obtained the following theorem:

Theorem 3. *The extended plan existence problem is undecidable in $\mathcal{ALC}_U$.*

To conclude this section, we are leaving questions Q1 and Q2 open for the description logics between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$. We conjecture that, without the universal role, it is not possible to enforce introduction of an unbounded number of individuals. It seems to be difficult even to enforce an exponential number of new individuals.

6 Conclusion and Future Work

In this paper, we have shown that the planning problems PLANEX and PLANLEN are decidable in action formalisms based on fragments of $\mathcal{ALCQIO}$. More precisely, both PLANEX and PLANLEN are of the same computational complexity as projection in the logics between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$. It is a not difficult to show that the same complexity results apply to the unrestricted version of the action formalism from [2], the one with occlusions. We conjecture that the extended plan existence problem for DLs without universal role is also decidable, but a proof is yet to be done.

A future work will include a development and implementation of efficient planners for description logics. Unfortunately, the complexity results we obtained are quite discouraging. Even in the propositional case, planning is a very hard combinatorial problem. An advantage in the propositional case is that, although PLANEX is PSpace-complete [5, 7], if we are only interested in polynomial-length plans (which is the case in practice), then planning becomes NP-complete. On the contrary, for DLs between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$, looking for polynomial-length plans is not easier than PLANEX, since the hardness results from Theorem 2 hold already for the plans of constant length. Thus it looks reasonable to start with "small" DLs, like $\mathcal{EL}$ or $\mathcal{EL}^{(\neg)}$ and try to adapt one of the well-known strategies from propositional planning: reduction to SAT [9]; planning based on planning graphs [4]; or a combination of the previous two [8].

There seem to be two possible methods for reducing projection and planning in $\mathcal{EL}$ to SAT. One would require "pre-computing" relevant consequences $C(a)$ of the initial ABox, where $C(a)$ is relevant if C is a sub-concept of the goal or of the concepts appearing in the pre-conditions, while the other one would use propositional formulae to describe models of the initial ABox.

Acknowledgements: The author wants to thank Carsten Lutz for inspiring discussions.

References

1. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, 2003.
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, C. Lutz, M. Milicic, U. Sattler, and F. Wolter. Integrating description logics and action formalisms for reasoning about web services. LTCS-Report 05-02, TU Dresden, Germany, 2005. See http://lat.inf.tu-dresden.de/research/reports.html.
4. A. Blum and M. Furst. Fast planning through planning graph analysis. In *Proceedings of Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 1636–1642, 1995.
5. T. Bylander. The computational complexity of propositional STRIPS planning. *Artificial Intelligence*, 69(1-2):165–204, 1994.
6. K. Erol, D. S. Nau, and V. S. Subrahmanian. Complexity, decidability and undecidability results for domain-independent planning: A detailed analysis. Technical Report CS-TR-2797, University of Maryland College Park, Maryland 20742, USA, 1991.
7. K. Erol, D. S. Nau, and V. S. Subrahmanian. Complexity, decidability and undecidability results for domain-independent planning. *Artificial Intelligence*, 76(1-2):75–88, 1995.
8. H. Kautz and B. Selman. Pushing the envelope: Planning, propositional logic, and stochastic search. In *Proceedings of In Proceedings of the 14th National Conference on Artificial Intelligence (AAAI-96)*, pages 1194–1201, Menlo Park, California, 1996.
9. H. A. Kautz and B. Selman. Planning as satisfiability. In *Proceedings of the Tenth European Conference on Artificial Intelligence (ECAI'92)*, pages 359–363, 1992.
10. H. Liu, C. Lutz, M. Milicic, and F. Wolter. Reasoning about actions using description logics with general TBoxes. In *Proceedings of the 10th European Conference on Logics in Artificial Intelligence (JELIA 2006)*, volume 4160 of *Lecture Notes in Artificial Intelligence*, pages 266–279, 2006.
11. H. Liu, C. Lutz, M. Milicic, and F. Wolter. Updating description logic ABoxes. In P. Doherty, J. Mylopoulos, and C. Welty, editors, *Proceedings of the Tenth International Conference on Principles of Knowledge Representation and Reasoning (KR'06)*, pages 46–56. AAAI Press, 2006.
12. R. Reiter. *Knowledge in Action.* MIT Press, 2001.
13. W. J. Savitch. Relationship between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4:177–192, 1970.
14. M. Thielscher. Introduction to the Fluent Calculus. *Electronic Transactions on Artificial Intelligence*, 2(3–4):179–192, 1998.