Integrating Description Logics and Action Formalisms: First Results

Tracking No. 998
Keywords: Description Logic, Computational Complexity

Abstract

We propose an action formalism that is based on description logics (DLs) and may be viewed as an instance of the Situation Calculus (SitCalc). In particular, description logic concepts can be used for describing the state of the world, and the pre- and post-conditions of actions. The main advantage of such a combination is that, on the one hand, the expressive power for describing world states and conditions is higher than in other decidable fragments of the SitCalc, which are usually propositional. On the other hand, in contrast to the full SitCalc, reasoning is still decidable. In this paper, we perform a detailed investigation of how the choice of the DL influences the complexity of standard reasoning tasks in the corresponding action formalism. We also discuss semantic and computational problems in natural extensions of our framework.

Introduction

Action formalisms such as the Situation Calculus (SitCalc) use full first-order logic for describing the state of the world, and the pre- and post-conditions of actions (Reiter 2001). Consequently, reasoning in such formalisms is undecidable. In contrast, the propositional variants of these formalisms enjoy decidability, but are rather restricted in expressive power. This dichotomy raises the obvious question whether some compromise between the two extremes can be found: an action formalism that offers more expressivity than propositional logic for describing world states and pre- and post-conditions of actions, but for which reasoning is still decidable.

Description Logics (DLs) are a well-known family of knowledge representation formalisms that may be viewed as fragments of first-order logic (FO). The main strength of DLs is that they offer considerable expressive power going far beyond propositional logic, while reasoning is still decidable (Baader *et al.* 2003). In this paper, we make an initial proposal for an action formalism in which the state of the world and the pre- and post-conditions can be described using DL concepts. The proposal is generic in the sense that our framework can be instantiated with many standard DLs. We show that our action formalism can be viewed as a fragment of the Situation Calculus and thus inherits SitCalc's well-established solution of the frame problem (Reiter 1991). Concerning reasoning, we focus on the basic tasks of executability and projection, which are mutually polynomially reducible in our framework. We exhibit a close connection between projection in our formalism instantiated with a description logic $\mathcal{L}$, and standard DL reasoning tasks in a moderate extension of $\mathcal{L}$. More precisely, we show that projection in $\mathcal{L}$ can be polynomially reduced to ABox consistency in $\mathcal{LO}$, the extension of $\mathcal{L}$ with nominals. In principle, adding nominals to a DL corresponds to admitting first-order constants inside DL concepts.

This reduction allows us to prove decidability and upper complexity bounds for executability and projection in our action formalism instantiated with a large number of standard DLs. Thus, we give a positive answer to the question whether there exists a decidable compromise between propositional and FO action theories. To pinpoint the exact computational complexity of our formalism, we show that, in a certain sense, the above reduction can be reversed: standard DL reasoning in $\mathcal{LO}$ can polynomially be reduced to projection in $\mathcal{L}$. In particular, this means that the additional computational complexity (sometimes) encountered by introducing nominals cannot be avoided. By combining the two reductions, we obtain tight complexity bounds for projection in many standard DLs that range from PSPACE-complete to co-NEXPTIME-complete.

We also consider some natural extensions of our basic formalism and point out some of the problems encountered with these extensions. In particular, we show that admitting more powerful post-conditions leads to undecidability of the basic reasoning problems. Because of the space constraints, all proofs and a more detailed discussion of the relationship to the situation calculus must be omitted. They can be found in the accompanying technical report (Anonymous 2004), which is available online in anonymized form.

The description logic $\mathcal{ALCQIO}$

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 a number of its sublanguages. The reason for choosing this family of DLs is that they are very expressive, but nevertheless admit practical reasoning. Indeed, DLs from this family underly highly op-

Name	Syntax	Semantics
inverse role	s^-	$\{(y,x) \mid (x,y) \in s^{\mathcal{I}}\}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
at-least number restriction	$(\geqslant n\ r\ C)$	$\{x \mid \mathit{card}(\{y \mid (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}) \geq n\}$
nominal	$\{a\}$	$\{a^{\mathcal{I}}\}$

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

timized DL systems such as FaCT and RACER.

In DL, 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 a set $\mathsf{N_I}$ of *individual names*. The constructors determine the expressive power of the DL. Table 1 shows a minimal set of constructors from which all constructors of $\mathcal{ALCQIO}$ can be defined. The first row contains the only role constructor: in $\mathcal{ALCQIO}$, a *role* is either a role name $s \in \mathsf{N_R}$ or the inverse s^- of a role name s. *Concepts* of $\mathcal{ALCQIO}$ are formed using the remaining constructors shown in Table 1, where r is a role, n a positive integer, and a an individual name. Using these constructors, several other constructors can be defined as abbreviations:

- $C \sqcup D := \neg(\neg C \sqcap \neg D)$ (disjunction),
- $\top := A \sqcup \neg A$ for a concept name A (top-concept),
- $\exists r.C := (\geqslant 1\ r\ C)$ (existential restriction),
- $\forall r.C := \neg\exists r.\neg C$ (value restriction),
- $(\leqslant n\ r\ C) := \neg(\geqslant (n+1)\ r\ C)$ (at-most restriction).

The DL that allows for negation, conjunction, and value 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 sublanguages.

The semantics of $\mathcal{ALCQIO}$-concepts and roles is defined in terms of an *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$. The domain $\Delta^{\mathcal{I}}$ of $\mathcal{I}$ 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 $s \in \mathsf{N_R}$ to a binary relation $s^{\mathcal{I}}$ on $\Delta^{\mathcal{I}}$, and each individual name $a \in \mathsf{N_I}$ to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. The extension of $\cdot^{\mathcal{I}}$ to arbitrary concepts and roles is inductively defined, as shown in the third column of Table 1. Here, the function *card* yields the cardinality of the given set. Note that the third column of Table 1 suggests a straightforward translation of DL concepts into first-order formulas with one free variable, as explicated e.g. in (Baader *et al.* 2003).

A *concept definition* is an identity of the form $A \equiv C$, where A is a concept name and C an $\mathcal{ALCQIO}$-concept. A *TBox* $\mathcal{T}$ is a finite set of concept definitions with unique left-hand sides. 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* $\mathcal{T}$. The TBox $\mathcal{T}$ is *acyclic* iff there are no cyclic dependencies between the definitions (Baader *et al.* 2003).

The *semantics* of TBoxes is defined in the obvious way: the interpretation $\mathcal{I}$ is a *model* of the TBox $\mathcal{T}$ iff it satisfies all its definitions, i.e., $A^{\mathcal{I}} = C^{\mathcal{I}}$ holds for all $A \equiv C$ in $\mathcal{T}$. In the case of acyclic TBoxes, any interpretation of the primitive concepts and of the role names can uniquely be extended to a model of the TBox (Nebel 1990).

An *ABox assertion* is of the form $C(a)$, $s(a,b)$ or $\neg s(a,b)$, where $a, b \in \mathsf{N_I}$, C is a concept, and s a role name.[1] An *ABox* is a finite set of ABox assertions. The interpretation $\mathcal{I}$ is a *model* of the ABox $\mathcal{A}$ iff it satisfies all its assertions, i.e., $a^{\mathcal{I}} \in C^{\mathcal{I}}$ ($(a^{\mathcal{I}}, b^{\mathcal{I}}) \in s^{\mathcal{I}}$, $(a^{\mathcal{I}}, b^{\mathcal{I}}) \notin s^{\mathcal{I}}$) for all assertions $C(a)$ ($s(a,b)$, $\neg s(a,b)$) in $\mathcal{A}$. If φ is an assertion, then we write $\mathcal{I} \models \varphi$ to indicate that $\mathcal{I}$ satisfies φ.

Various reasoning problems are considered for DLs. For the purpose of this paper, it suffices to introduce concept satisfiability and ABox consistency: the concept C is satisfiable w.r.t. the TBox $\mathcal{T}$ iff there exists a model $\mathcal{I}$ of $\mathcal{T}$ such that $C^{\mathcal{I}} \neq \emptyset$; the ABox $\mathcal{A}$ is *consistent* w.r.t. the TBox $\mathcal{T}$ iff there exists an interpretation $\mathcal{I}$ that is a model of both $\mathcal{T}$ and $\mathcal{A}$.

Describing actions

We introduce syntax and semantics of our action formalism. As we will argue later, this formalism can be viewed as a fragment of the SitCalc in a straightforward way. However, unlike in the SitCalc we are not working with a first-order syntax: since first-order translations of DL formulas are rather awkward, we prefer to describe actions using DL syntax inside a STRIPS-like formalism. An acyclic TBox is used to define the background information, i.e., the meaning of concept names.

Definition 1 (Action). Let $\mathcal{T}$ be an acyclic TBox. An *atomic 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 $s(a,b)$, with A primitive concept in $\mathcal{T}$, s role name, and $a, b \in \mathsf{N_I}$;
- 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* $\mathcal{T}$, i.e., an ABox assertion $A(a)$, $\neg A(a)$, $s(a,b)$, or $\neg s(a,b)$ with A primitive concept name in $\mathcal{T}$ and s role name.

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

Intuitively, the pre-conditions specify under which conditions the action is applicable. The conditional post-conditions φ/ψ say that, if φ is true before executing the action, then ψ should be true afterwards. By the law of inertia, only those facts that are forced to change by the post-conditions should be changed by applying the action. However, it is well-known that enforcing this minimization of change strictly is sometimes too restrictive (Lifschitz 1990; Sandewall 1994). The rôle of occlusions is to describe those primitive literals to which minimization is not applied.

To illustrate the definition of actions, consider the actions of opening a bank account and applying for child benefit. Suppose the pre-condition of opening a bank account is that the customer a is eligible for a bank account in the UK and

[1] Disallowing inverse roles in ABox assertions is not a restriction since $s^-(a,b)$ can be expressed by $s(b,a)$.

holds a proof of address. Moreover, suppose that, if a letter from the employer is available, then the bank account comes with a credit card, otherwise not. This action can be formalised by the following action α_1, for which the set of occlusions is empty:

$$\begin{array}{ll}\mathsf{pre}: & \{\texttt{Eligible\_bank}(a), \exists \texttt{holds}.\texttt{Proof\_address}(a)\}\\ \mathsf{post}: & \{\top(a)/\texttt{holds}(a,b),\\ & \exists \texttt{holds}.\texttt{Letter}(a)/\texttt{B\_acc\_credit}(b),\\ & \neg\exists \texttt{holds}.\texttt{Letter}(a)/\texttt{B\_acc\_no\_credit}(b)\}\end{array}$$

Suppose that one can apply for child benefit in the UK if one has a child and a bank account. The action α_2 that offers this application then looks as follows, where again the set of occlusions is empty:

$$\begin{array}{ll}\mathsf{pre}: & \{\texttt{parent\_of}(a,c), \exists \texttt{holds}.\texttt{B\_acc}(a)\}\\ \mathsf{post}: & \{\top(a)/\texttt{receives\_c\_benef\_for}(a,c)\}\end{array}$$

The meaning of the concepts used in α_1 and α_2 are defined in the following acyclic TBox $\mathcal{T}$:

$$\begin{array}{rcl}\texttt{Eligible\_bank} & \equiv & \exists \texttt{permanent\_resident}.\{\texttt{UK}\}\\ \texttt{Proof\_address} & \equiv & \texttt{Electricity\_contract}\\ \texttt{B\_acc} & \equiv & \texttt{B\_acc\_credit} \sqcup \texttt{B\_acc\_no\_credit}\end{array}$$

We assume that states of the world correspond to interpretations. Thus, the semantics of actions can be defined by means of a transition relation on interpretations. Let $\mathcal{T}$ be an acyclic TBox, $\alpha = (\mathsf{pre}, \mathsf{occ}, \mathsf{post})$ an action for $\mathcal{T}$, and $\mathcal{I}$ an interpretation. For each primitive concept name A and role name s, set:

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

The transition relation on interpretations should ensure that $A^+ \subseteq A^{\mathcal{J}}$ and $A^- \cap A^{\mathcal{J}} = \emptyset$ if $\mathcal{J}$ is the result of applying α in $\mathcal{I}$. It should also ensure that nothing else changes, with the possible exception of the occluded literals. Intuitively, the part of $A^{\mathcal{I}}$ that is *not* subject to occlusions is described by I_A, and similarly for $s^{\mathcal{I}}$ and I_s. Since we restrict our attention to acyclic TBoxes, for which the interpretation of defined concepts is uniquely determined by the interpretation of primitive concepts and role names, it is not necessary to consider defined concepts when defining the transition relation.

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

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

The composite action $\alpha_1 \ldots, \alpha_k$ *may transform* $\mathcal{I}$ to $\mathcal{I}'$ ($\mathcal{I} \Rightarrow^{\mathcal{T}}_{\alpha_1,\ldots,\alpha_k} \mathcal{I}'$) 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 $\mathcal{I}_{i-1} \Rightarrow^{\mathcal{T}}_{\alpha_i} \mathcal{I}_i$ 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.

Due to the fact that we are working with acyclic TBoxes, for actions with empty occlusions there cannot exist more than one $\mathcal{I}'$ such that $\mathcal{I} \Rightarrow^{\mathcal{T}}_{\alpha} \mathcal{I}'$. Thus, such actions are deterministic. If there are post-conditions $\varphi_1/\psi, \varphi_2/\neg\psi \in \mathsf{post}$ such that both φ_1 and φ_2 are satisfied in $\mathcal{I}$, then there is no successor model $\mathcal{I}'$. In this case, we say that the action is *inconsistent with* $\mathcal{I}$.

Reasoning about actions

We are now ready to define reasoning problem for actions. Assume that we want to apply a composite action $\alpha_1, \ldots, \alpha_k$ for the acyclic TBox $\mathcal{T}$. Usually, we do not have complete information about the world, i.e., the model $\mathcal{I}$ of $\mathcal{T}$ is not known completely. All we know are some facts about this world: we have an ABox $\mathcal{A}$, and all models of $\mathcal{A}$ together with $\mathcal{T}$ are considered to be possible states of the world.

Before trying to apply the action, we want to know whether it is indeed executable, i.e., whether all necessary pre-conditions are satisfied in the states of the world considered possible. If the action is executable, we want to know whether applying it achieves the desired effect, i.e., whether an assertion that we want to make true really holds after executing the action. These two problems are called executability and projection (Reiter 2001).

Definition 3 (Reasoning problems). Let $\mathcal{T}$ be an acyclic TBox, $\alpha_1, \ldots, \alpha_k$ a composite action for $\mathcal{T}$ with $\alpha_i = (\mathsf{pre}_i, \mathsf{occ}_i, \mathsf{post}_i)$, and $\mathcal{A}$ an ABox.

- *Executability:*. The composite action $\alpha_1, \ldots, \alpha_k$ 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 $1 \leq i < k$ 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 $\alpha_1, \ldots, \alpha_k$ in $\mathcal{A}$ w.r.t. $\mathcal{T}$* iff, for all models $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$, and all $\mathcal{I}'$ with $\mathcal{I} \Rightarrow^{\mathcal{T}}_{\alpha_1,\ldots,\alpha_k} \mathcal{I}'$, we have $\mathcal{I}' \models \varphi$.

Note that executability alone does not guarantee that we cannot get stuck while executing a composite action: it may be that the action to be applied is inconsistent with the current interpretation. This cannot happen if we additionally know that all actions α_i are *consistent with* $\mathcal{T}$ in the following sense: α_i is not inconsistent with any model $\mathcal{I}$ of $\mathcal{T}$. Given the definition of consistency *with a model*, it is not difficult to see that this is the case iff $\{\varphi_1/\psi, \varphi_2/\neg\psi\} \subseteq \mathsf{post}_i$ implies that the ABox $\{\varphi_1, \varphi_2\}$ is inconsistent w.r.t. $\mathcal{T}$. Thus, consistency of an action w.r.t. $\mathcal{T}$ can be reduced to standard DL reasoning.

In our example, both actions are consistent with $\mathcal{T}$. Given an ABox that says that customer a is a permanent resident of the UK, has an electricity contract, and a child l, the composite action α_1, α_2 is executable, and $\texttt{receives\_c\_benef\_for}(a,c)$ is a consequence of applying α_1, α_2. The presence of the TBox is crucial here.

Note that our action formalism is restricted to ground actions, i.e., actions where the input parameters have already been instantiated by individual names. Parametric actions, which contain variables in place of individual names, should be viewed as a compact representation of all its ground instances, i.e., all the ground actions obtained by replacing variables by individual names. It is outside the scope of this paper to consider parametric actions in detail. However, the reasoning tasks executability and projection are only meaningful for ground actions, anyway.

Relationship with the situation calculus

The situation calculus is an established and widely used formalism to represent actions (Reiter 2001). It can be seen as a sorted first order logic framework that provides a methodology to axiomatise the effects of actions. We can show that, for actions without occlusions, our approach can be seen as an instance of Reiter's action formalism. For actions with occlusions, related formalisms can be found in (Shanahan 1997).

Suppose an ABox $\mathcal{A}$, an acyclic TBox $\mathcal{T}$, and a composite action $\alpha_1, \ldots, \alpha_k$ are given. First, we can get rid of the TBox by expanding it (i.e., recursively substituting defined concepts by their definitions) and then replacing in $\mathcal{A}$ and the actions $\alpha_1, \ldots, \alpha_k$ the defined concepts by their definitions.[2] Next, we can use the standard translation of $\mathcal{ALCQIO}$ into first-order logic (Baader *et al*. 2003) to translate the semantics of actions as given in Definition 2 into *action pre-conditions* and *successor state axioms* in the sense of (Reiter 2001). In this setting, primitive concepts and role names are regarded as fluents. Moreover, by taking as the description of the initial state the first-order translation of the ABox, we can show that our notions of executability and projection are instances of Reiter's definitions (see (Anonymous 2004) for details).

Note that the existence of this translation into SitCalc does not mean that the inference problems introduced above can be solved using an implemented system for reasoning about action, such as GOLOG (Levesque *et al.* 1997). In fact, in Reiter's approach, *regression* (Reiter 2001) is used to solve the executability and the projection problem. However, when applied to (the translation of) our actions, regression yields a standard first-order theory, which is not in the scope of what GOLOG can handle without calling a general first-order theorem prover. Thus, the translation into SitCalc does not directly provide us with decidability or complexity results for our reasoning problems.

Deciding executability and projection

In this section, we determine the exact complexity of executability and projection for composite actions expressed in various sublanguages of $\mathcal{ALCQIO}$. In these results, we assume unary coding of numbers in number restrictions. Throughout this section, we assume that all actions are consistent with their TBox. The following is shown in (Anonymous 2004).

[2] Alternatively, we could handle the TBox as state constraints.

Lemma 4. *Executability and projection can be reduced to each other in polynomial time.*

Thus, we can restrict the attention to the projection problem. Basically, we solve this problem by an approach that is similar to the regression operation used in the situation calculus approach (Reiter 2001). However, we take care that the theory we obtain can again be expressed by a description logic TBox and ABox. This way, projection is reduced to a standard reasoning problem in DL, from which we obtain our decidability results and upper complexity bounds. Interestingly, we cannot always stay within the DL we started with since we need to introduce nominals. Given a DL $\mathcal{L}$, we denote its extension by nominals with $\mathcal{LO}$.

Theorem 5. *Let* $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}, \mathcal{ALCO}, \mathcal{ALCIO}, \mathcal{ALCQ}, \mathcal{ALCQO}, \mathcal{ALCQI}, \mathcal{ALCQIO}\}$. *Then projection of composite actions formulated in* $\mathcal{L}$ *can polynomially be reduced to non-consistency in* $\mathcal{LO}$ *of an ABox relative to an acyclic TBox.*

For lack of space, we cannot give a detailed proof of this result and refer to (Anonymous 2004) for details. Here, we only give a brief sketch of the proof for the case of an atomic action without occlusions. We reduce the complement of projection in $\mathcal{L}$ to the consistency problem for ABoxes in $\mathcal{LO}$ (and thus projection in $\mathcal{L}$ to non-consistency in $\mathcal{LO}$), where $\mathcal{L}$ is one of the languages from Theorem 5.

Given an ABox $\mathcal{A}$, an acyclic TBox $\mathcal{T}$, an action $\alpha = (\mathsf{pre}, \emptyset, \mathsf{post})$, and an ABox assertion φ (all formulated in $\mathcal{L}$), we construct a new TBox $\mathcal{T}_r$ and a new ABox $\mathcal{A}_r$, and a new assertion φ_r (all formulated in $\mathcal{LO}$) such that the following are equivalent:

1. There exist models $\mathcal{I}, \mathcal{I}'$ of $\mathcal{T}$ such that $\mathcal{I}$ satisfies $\mathcal{A}$, $\mathcal{I}'$ satisfies $\neg\varphi$ and $\mathcal{I} \Rightarrow_\alpha^{\mathcal{T}} \mathcal{I}'$.
2. $\mathcal{A}_r \cup \{\neg\varphi_r\}$ is consistent w.r.t. $\mathcal{T}_r$.

Obviously, 1. means that φ is *not* a consequence of applying α in $\mathcal{A}$ w.r.t. $\mathcal{T}$.

We now describe the general idea underlying the construction of $\mathcal{T}_r$ and $\mathcal{A}_r$. The goal is to simulate transformations $\mathcal{I} \Rightarrow_\alpha^{\mathcal{T}} \mathcal{I}'$ with $\mathcal{I} \models \mathcal{A}$ and $\mathcal{I}' \not\models \varphi$ within a *single* interpretation $\mathcal{J}$, which is a model of $\mathcal{T}_r$ and $\mathcal{A}_r \cup \{\neg\varphi_r\}$. Thus, $\mathcal{J}$ needs to encode *two* interpretations $\mathcal{I}$ and $\mathcal{I}'$. To this end, for every concept name A and role name r we introduce new primed versions A' and r'. Then, the $\mathcal{J}$-interpretation of the unprimed concept and role names corresponds to $\mathcal{I}$, and the $\mathcal{J}$-interpretation of the primed concept and role names corresponds to $\mathcal{I}'$. Let $\mathcal{T}', \varphi'$ be the version of $\mathcal{T}, \varphi$ obtained by replacing concept and role names with their primed counterparts. We construct $\mathcal{T}_r$ such that it contains $\mathcal{T}$ and (a modification of) $\mathcal{T}'$: before and after the execution of the action, the TBox should be satisfied. Also, φ_r is simply φ', and $\mathcal{A}_r$ contains (the non-primed) $\mathcal{A}$: before execution of α, $\mathcal{A}$ should be satisfied.

Additional efforts are required to describe how the interpretation of the primed versions of concepts and roles is obtained from the interpretation of the unprimed ones. Intuitively, this task is split into two parts: (i) describe the evolution of the *named elements*, i.e., elements $x \in \Delta^{\mathcal{I}}$ such that $a^{\mathcal{I}} = x$ for some individual name a; and (ii) describe

the evolution of the unnamed elements. Roughly, (i) can be achieved by adding additional statements to $\mathcal{A}_r$ that can be derived straightforwardly from Definition 2. To achieve (ii), the TBox $\mathcal{T}'$ is contained in $\mathcal{T}_r$ in a strongly modified form. In this modified version of $\mathcal{T}'$, it is necessary to distinguish named elements from unnamed ones. This, in turn, can be achieved by making intense use of nominals. In this way, it is not necessary to introduce any constructors not occurring in $\mathcal{T}, \mathcal{A}$ or available in $\mathcal{ALCO}$.

Theorem 6. *Projection and executability of composite actions is*

- PSPACE*-complete for* $\mathcal{ALC}, \mathcal{ALCO}, \mathcal{ALCQ}, \mathcal{ALCQO}$*;*
- EXPTIME*-complete for* $\mathcal{ALCI}, \mathcal{ALCIO}$*;*
- *co-*NEXPTIME*-complete for* $\mathcal{ALCQI}, \mathcal{ALCQIO}$.

The complexity *upper-bounds* follow from Theorem 5 together with either known results for ABox consistency w.r.t. an acyclic TBox or results shown in the long version of this paper (Anonymous 2004):

- ABox consistency in $\mathcal{ALCO}$ and $\mathcal{ALCQO}$ w.r.t. acyclic TBoxes is PSPACE-complete (Anonymous 2004).
- ABox consistency in $\mathcal{ALCIO}$ w.r.t. acyclic TBoxes is EXPTIME-complete (Areces, Blackburn, & Marx 1999);
- ABox consistency in $\mathcal{ALCQIO}$ w.r.t. acyclic TBoxes is NEXPTIME-complete (Tobies 2000).

It is easy to obtain matching *lower-bounds* for those DLs $\mathcal{L}$ where the complexity of ABox consistency w.r.t. an acyclic TBox is the same in $\mathcal{L}$ and in $\mathcal{LO}$. In fact, it suffices to note that we can easily reduce ABox non-consistency in $\mathcal{L}$ to projection/executability in $\mathcal{L}$: $\mathcal{A}$ is inconsistent w.r.t. $\mathcal{T}$ iff $\neg\top(a)$ is a consequence of applying the empty action $(\emptyset, \emptyset, \emptyset)$ in $\mathcal{A}$ w.r.t. $\mathcal{T}$.

This argument does not provide matching lower bounds for $\mathcal{ALCI}$ and $\mathcal{ALCQI}$ since, for these DLs, adding nominals increases the complexity of the ABox consistency problem. However, for $\mathcal{L} \in \{\mathcal{ALCI}, \mathcal{ALCQI}\}$, we may establish such bounds by reducing unsatisfiability of $\mathcal{LO}$ concepts (w.r.t. the empty TBox) to projection in $\mathcal{L}$. Intuitively, this result states that the additional complexity obtained by introducing nominals in the reduction of projection to ABox inconsistency cannot be avoided.

Theorem 7. *There exists an ABox $\mathcal{A}$ and an atomic action α formulated in $\mathcal{ALCI}$ ($\mathcal{ALCQI}$) such that the following tasks are* EXPTIME*-hard (co-*NEXPTIME*-hard): given an ABox assertion φ,*

- *decide whether φ is a consequence of applying α in $\mathcal{A}$;*
- *decide whether $\alpha, (\{\varphi\}, \emptyset, \emptyset)$ is executable in $\mathcal{A}$.*

For the proof of Theorem 7, let $\mathcal{L} \in \{\mathcal{ALCIO}, \mathcal{ALCQIO}\}$ and C an $\mathcal{L}$-concept whose unsatisfiability is to be decided. For simplicity, we assume that C contains only a single nominal $\{n\}$. This can be done w.l.o.g. since the complexity of unsatisfiability in $\mathcal{ALCIO}$ ($\mathcal{ALCQIO}$) is already EXPTIME-hard (co-NEXPTIME-hard) if only a single nominal is available and TBoxes are not admitted (Areces, Blackburn, & Marx 1999; Tobies 2000). For the reduction, we reserve a concept name O and a role name u that do not occur in C. Let $\mathsf{rol}(C) := \{r, r^- \mid r \in \mathsf{N_R} \text{ used in } C\}$ and let $C[O/\{n\}]$ denote the result of replacing the nominal $\{n\}$ in C with the concept name O. We define an ABox $\mathcal{A}$, an atomic action $\alpha = (\emptyset, \emptyset, \mathsf{post}_\alpha)$, and a concept D_C as follows:

$$\mathcal{A}_C := \{(\neg O \sqcap \forall u.\neg O \sqcap \forall u. \bigsqcap_{r \in \mathsf{rol}(C)} \forall r.\exists u^-.\neg O)(a)\}$$

$$\mathsf{post}_\alpha := \{\top(a)/O(a)\}$$

$$D_C := \exists u.C[O/\{n\}] \sqcap (\forall u. \bigsqcap_{r \in \mathsf{rol}(C)} \forall r.\forall u^-.O)$$

Theorem 7 is an immediate consequence of the following lemma.

Lemma 8. *The following statements are equivalent:*

1. *C is satisfiable.*
2. *$\neg D_C(a)$ is not a consequence of applying α in $\mathcal{A}_C$.*
3. *the composite action $\alpha, (\{\neg D_C(a)\}, \emptyset, \emptyset)$ is not executable in $\mathcal{A}_C$.*

A detailed proof can be found in (Anonymous 2004). Here, we only sketch the underlying intuitions for why Point 2 of Lemma 8 implies Point 1. Let $\mathcal{I}$ and $\mathcal{I}'$ be models witnessing that $\neg D_C(a)$ is *not* a consequence of α, i.e., $\mathcal{I} \models \mathcal{A}_C, \mathcal{I} \Rightarrow_\alpha \mathcal{I}'$, and $\mathcal{I}' \models D_C(a)$. Then the following holds:

- By the first conjunct of (the concept in) $\mathcal{A}_C$ and the post-condition, the only difference between $\mathcal{I}$ and $\mathcal{I}'$ is that $a^\mathcal{I} = a^{\mathcal{I}'} \in O^{\mathcal{I}'} \setminus O^\mathcal{I}$;
- Using the first and third conjunct of $\mathcal{A}_C$ together with the post-condition and the second conjunct of D_C, it can be shown that $(a^\mathcal{I}, x) \in u^\mathcal{I} = u^{\mathcal{I}'}$ for each x from the relevant part rel of $\Delta^\mathcal{I}$, where rel is defined as the smallest set that contains $a^\mathcal{I}$ and is closed under taking successors for the roles from $\mathsf{rol}(C)$;
- Thus, the second conjunct of $\mathcal{A}_C$ ensures that $O^\mathcal{I} \cap \mathsf{rel} = \emptyset$ and $O^{\mathcal{I}'} \cap \mathsf{rel} = \{a^\mathcal{I}\}$.
- Due to the first conjunct of D_C, $C[O/\{n\}]$ is satisfied in the relevant part of $\mathcal{I}'$. By the previous item, the concept name O behaves like a nominal.

Problematic extensions

The purpose of this section is to provide a justification for the restrictions that we have adopted in our formalism for describing actions:

1. we only allow for acyclic TBoxes rather than arbitrary (also cyclic) ones or even so-called general concept inclusions (GCIs);
2. in post-conditions $\varphi/C(a)$, we require C to be a primitive concept or its negation, rather than admitting arbitrary concepts.

Removing the first restriction leads to *semantic problems*. In fact, if the TBox is cyclic, then it is no longer the case that the interpretation of the primitive concepts and the role names uniquely determines the interpretation of the defined concepts. This can lead to very unintuitive results. For example, consider the ABox and TBox

$$\mathcal{A} := \{\mathsf{Dog}(a)\} \text{ and } \mathcal{T} := \{\mathsf{Dog} \doteq \exists \mathsf{parent}.\mathsf{Dog}\}$$

Then, $\mathsf{Dog}(a)$ is *not* a consequence of applying the action $\alpha = (\emptyset, \emptyset, \{\top(b)/\mathsf{Cat}(b)\})$ in $\mathcal{A}$ w.r.t. $\mathcal{T}$. The reason is that the transition relation $\Rightarrow_\alpha$ only considers primitive concepts and role names (see (Anonymous 2004) for more details).

One may be tempted to think that an easy solution to this problem is to modify the semantics such that the changes on defined concepts are minimized in a similar way as Definition 2 minimizes the changes of primitive concepts. However, it is well-known in the reasoning about actions community that such a naive approach leads to very unintuitive results as well (Lifschitz 1990). A more promising approach could be to adopt a fixpoint semantics for cyclic TBoxes (Nebel 1991). Under such a semantics, the interpretation of the defined concept names is still uniquely determined by the interpretation of the primitive concept names and role names.

Semantic problems are also encountered when removing the second restriction. In particular, admitting arbitrary concepts in post-conditions means that we cannot anymore give a straightforward semantics as in Definition 2. One possible way to obtain a semantics for actions with complex post-conditons is to adopt the possible models approach (PMA) initially proposed in (Winslett 1988). The formal definition of such a semantics can be found in (Anonymous 2004). Unfortunately, adopting the PMA semantics results in two problems. The first problem is again of a *semantic* nature: using complex concepts in post-conditions under PMA results in massive non-determinism. Such nondeterminism requires special mechanisms to be used meaningfully, e.g. based on notions of causality (Zhang & Foo 2000; Thielscher 2000; Lin 1996). It seems unlikely that a suitable mechanism can be found for the case of *arbitrary* concepts as post-conditions.

Second, we now also have *algorithmic problems*: the basic reasoning tasks are not decidable anymore. Let a *generalized* action be an action where post-conditions are of the form φ/ψ for arbitrary ABox assertions φ and ψ.

Theorem 9. *Executability and projection are undecidable for generalized actions in $\mathcal{ALCQI}$ under PMA semantics.*

This result is proved in (Anonymous 2004) by showing that there exist a fixed generalized action α formulated in $\mathcal{ALCQI}$ [3] and a fixed ABox $\mathcal{A}$ such that, given a concept C, it is undecidable whether $C(a)$ is a consequence of applying α in $\mathcal{A}$ w.r.t. the empty TBox. The proof is by a non-trivial reduction of the domino problem.

Conclusion

In this paper, we have proposed an initial framework for integrating DLs and action formalisms into a decidable hybrid. In particular, our framework allows the use of DL concepts for describing the state of the world, and the pre- and post-conditions of actions. Our main technical result is that the computational complexity of projection and executability coincides with the complexity of ABox non-consistency in the underlying DL extended with nominals.

[3] Even in its fragment $\mathcal{ALCFI}$ where only the numbers zero and one may be used inside number restrictions.

As this work is only a first proposal, there is room for extensions in several directions. We only note two options: first, it is clearly desirable to identify a semantics that overcomes the problems with cyclic TBoxes and GCIs laid out in the previous section. And second, one may attempt to carefully enhance the expressive power of post-conditions without running into the troubles obtained by admitting *arbitrary* concepts as post-conditions.

References

Anonymous. 2004. Integrating description logics and action formalisms for reasoning about web services. Technical report. http://aaai05sub.tripod.com/index.htm.

Areces, C.; Blackburn, P.; and Marx, M. 1999. A roadmap on complexity for hybrid logics. In *Proc. of CSL'05*, number 1683 in LNCS, 307–321. Springer-Verlag.

Baader, F.; Calvanese, D.; McGuinness, D.; Nardi, D.; and Patel-Schneider, P. F., eds. 2003. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.

Levesque, H. J.; Reiter, R.; Lesperance, L.; Lin, F.; and Scherl, R. B. 1997. GOLOG: a logic programming language for dynamic domains. *Journal of Logic Programming* 31(1-3):59–83.

Lifschitz, V. 1990. Frames in the space of situations. *Artificial Intelligence Journal* 46:365–376.

Lin, F. 1996. Embracing causality in specifying the indeterminate effects of actions. In *Proc. of AAAI-96*, 670–676. MIT Press.

Nebel, B. 1990. Terminological reasoning is inherently intractable. *Artificial Intelligence* 43:235–249.

Nebel, B. 1991. Terminological cycles: Semantics and computational properties. In *Principles of Semantic Networks: Explorations in the Representation of Knowledge*. Morgan Kaufmann. 331–361.

Reiter, R. 1991. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In *AI and Mathematical Theory of Computation*. Academic Press. 359–380.

Reiter, R. 2001. *Knowledge in Action*. MIT Press.

Sandewall, E. 1994. *Features and Fluents*. Oxford University Press.

Shanahan, M. 1997. *Solving the Frame Problem*. MIT Press.

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

Tobies, S. 2000. The complexity of reasoning with cardinality restrictions and nominals in expressive description logics. *JAIR* 12:199–217.

Winslett, M. 1988. Reasoning about action using a possible models approach. In *Proc. of AAAI'88*, 89–93.

Zhang, Y., and Foo, N. 2000. Updating knowledge bases with disjunctive information. *Computational Intelligence* 16:1–22.