The Projection Problem for $\mathcal{EL}$ Actions

Hongkai Liu⋆, Carsten Lutz, Maja Miličić⋆⋆

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

1 Introduction

Classical action formalisms form a dichotomy regarding their expressive power and computational properties: they are either based on first-order logic (FOL) and undecidable like the Situation Calculus [13], or decidable but only propositional like STRIPS [8, 7]. In [3, 11], it was proposed to integrate description logics (DLs) into action formalisms in order to increase the expressive power beyond propositional logic while retaining decidability of reasoning. In particular, ABox assertions are used for describing the initial state of the world and the pre- and post-conditions of actions, and acyclic TBoxes are used to describe background knowledge. A similar approach based on the 2-variable fragment of FOL is described in [9]. The results in [3] show that, even if expressive DLs such as $\mathcal{ALCQIO}$ are used in the action formalism, standard reasoning problems such as executability and projection remain decidable. The proof is by a reduction of these problems in a DL $\mathcal{L}$ to instance checking in the extension $\mathcal{LO}$ of $\mathcal{L}$ with nominals, and it works for all standard extensions of the propositionally closed DL $\mathcal{ALC}$.

A recent trend in description logic is to consider lightweight DLs that are not propositionally closed and for which standard reasoning problems such as subsumption and instance checking are tractable. In particular, the $\mathcal{EL}$-family of DLs has been developed in [1, 6, 2, 4], and it has proved useful for modelling life science ontologies such as SNOMED [16] and the National Cancer Institute's NCI thesaurus [15]. Many such ontologies are acyclic TBoxes and can thus be used in a DL-based action formalism. This paves the way to new applications such as the following: one can use ABoxes to describe patient data in the medical domain, actions to represent medical treatments, and in both cases use concepts defined in an underlying medical ontology. Executability and projection can then determine, e.g., whether a certain treatment is effective or has undesired side-effects.

In this paper, we investigate the complexity of executability and projection in $\mathcal{EL}$ and $\mathcal{EL}^{(\neg)}$, the extension of $\mathcal{EL}$ with atomic negation. In both cases, we allow for negated assertions in the post-conditions of actions. Our results show that, in general, tractability does not transfer from instance checking to executability and projection. Even in $\mathcal{EL}$ without TBoxes, the latter problems are

⋆ supported by the DFG project BA 1122/13
⋆⋆ supported by the EU project TONES

co-NP-hard. This is due to two sources of intractability: (1) existential restrictions in the initial ABox together with negated assertions in post-conditions; and (2) conditional post-conditions. We remark that co-NP-hardness does not follow from hardness results for propositional action formalisms since $\mathcal{EL}$ does not have disjunction, and negation occurs only in post-conditions. We prove a matching co-NP upper bound for $\mathcal{EL}^{(\neg)}$. We also show that, in the presence of acyclic TBoxes, projection in $\mathcal{EL}$ is PSPACE-hard and thus not easier than in $\mathcal{ALC}$. Finally, we identify restrictions under which executability and projection in $\mathcal{EL}$ w.r.t. acyclic TBoxes can be decided in polynomial time. These restrictions subsume the case where only positive post-conditions are admitted, but still allow for a careful use of negated post-conditions.

2 DL Actions

Let $\mathcal{T}$ be an acyclic TBox describing general knowledge about the application domain, similarly to state constraints in the Situation Calculus [13]. An *atomic action* $\alpha = (\mathsf{pre}, \mathsf{post})$ 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* $\mathcal{T}$, i.e. an ABox assertion

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

  with A a concept name primitive in $\mathcal{T}$ and r a role name.

A *composite action* for $\mathcal{T}$ is a finite sequence $\pi = \alpha_1, \ldots, \alpha_k$ of atomic actions for $\mathcal{T}$. Note that we allow negation in post-conditions although $\mathcal{EL}$ does not provide negation. Intuitively, actions without negated post-conditions seem too restrictive to be useful. We will return to this issue in Section 5.

Applying an action changes the state of the world, and thus transforms an interpretation $\mathcal{I}$ into an interpretation $\mathcal{J}$. 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 to $\mathcal{I}$. This can be formalized as follows.

Let $\mathcal{T}$ be an acyclic TBox, $\alpha = (\mathsf{pre}, \mathsf{post})$ an atomic action for $\mathcal{T}$, and $\mathcal{I}, \mathcal{J}$ models of $\mathcal{T}$ that respect the unique name assumption (UNA), have identical domain, and agree on the interpretation of individual names. We say that α *transforms* $\mathcal{I}$ into $\mathcal{J}$ ($\mathcal{I} \Rightarrow_\alpha^\mathcal{T} \mathcal{J}$) iff, for each primitive concept A and role name r, we have

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

The composite action $\pi = \alpha_1, \ldots, \alpha_k$ *transforms* $\mathcal{I}$ to $\mathcal{J}$ (written $\mathcal{I} \Rightarrow^{\mathcal{T}}_{\pi} \mathcal{J}$) iff there are models $\mathcal{I}_0, \ldots, \mathcal{I}_k$ of $\mathcal{T}$ with $\mathcal{I} = \mathcal{I}_0$, $\mathcal{J} = \mathcal{I}_k$, and $\mathcal{I}_{i-1} \Rightarrow^{\mathcal{T}}_{\alpha_i} \mathcal{I}_i$ for $1 \leq i \leq k$. Note that, since we use acyclic TBoxes, there cannot exist more than one $\mathcal{J}$ such that $\mathcal{I} \Rightarrow^{\mathcal{T}}_{\alpha} \mathcal{J}$. Thus, actions are deterministic. We generally assume that actions $\alpha = (\mathsf{pre}, \mathsf{post})$ are *consistent* in the following sense: for every interpretation $\mathcal{I}$, there exists $\mathcal{J}$ such that $\mathcal{I} \Rightarrow^{\mathcal{T}}_{\alpha} \mathcal{J}$. It is not difficult to see that this is the case iff post does not contain any pair of post-conditions of the form $\varphi_1/\psi, \varphi_2/\neg\psi$.

Projection and *executability* are the most important reasoning problems on actions. Executability is the problem of deciding whether an action can be applied in a given situation. Formally, we say that an action $\pi = \alpha_1, \ldots, \alpha_n$ is *executable* in an ABox $\mathcal{A}$ w.r.t. an acyclic TBox $\mathcal{T}$ if 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 $\mathcal{J}$ with $\mathcal{I} \Rightarrow^{\mathcal{T}}_{\alpha_1,\ldots,\alpha_i} \mathcal{J}$, we have $\mathcal{J} \models \mathsf{pre}_{i+1}$.

Projection is the problem of deciding whether applying an action achieves a desired effect. Formally, the ABox assertion φ is a *consequence* of applying the action $\pi = \alpha_1, \ldots, \alpha_n$ in $\mathcal{A}$ w.r.t. $\mathcal{T}$ (written $\mathcal{T}, \mathcal{A}^{\pi} \models \varphi$) iff for all models $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$ and for all $\mathcal{J}$ with $\mathcal{I} \Rightarrow^{\mathcal{T}}_{\pi} \mathcal{J}$, we have $\mathcal{J} \models \varphi$. In this context, we also call φ the *goal*. If $\mathcal{T}$ is empty, we write $\mathcal{A}^{\pi} \models \varphi$ instead of $\mathcal{T}, \mathcal{A}^{\pi} \models \varphi$. It has been shown in [3] that projection and executability are mutually reducible in polynomial time. Thus, we will focus only on projection in the rest of the paper.

In lower bound proofs, we use actions of a restricted form: only atomic actions are admitted, the set of pre-conditions is empty, and post-conditions are unconditional, i.e., of the form $\top(a)/\psi$. We call such actions *updates* and write $\alpha = \{\psi_1, \ldots, \psi_n\}$ instead of $\alpha = (\emptyset, \{\top(a)/\psi_1, \ldots, \top(a)/\psi_n\})$.

3 Projection in $\mathcal{EL}$ with empty TBoxes

We show that, without TBoxes, projection in $\mathcal{EL}$ and $\mathcal{EL}^{(\neg)}$ is co-NP-complete. The lower bound, which is proved for $\mathcal{EL}$, is a variation of Schaerf's proof that instance checking in $\mathcal{EL}^{(\neg)}$ w.r.t. empty TBoxes is co-NP-hard regarding data complexity [14]. It uses a reduction of a variation of the propositional SAT problem called 2+2-SAT, which is shown to be NP-complete in [14].

A *2+2 clause* is of the form $(p_1 \vee p_2 \vee \neg n_1 \vee \neg n_2)$, where each of p_1, p_2, n_1, n_2 is a propositional letter or a truth constant $\top$, $\bot$. A *2+2 formula* is a finite conjunction of 2+2 clauses. Now, 2+2-SAT is the problem of deciding whether a given 2+2 formula is satisfiable. Let $\varphi = c_1 \wedge \cdots \wedge c_n$ be a 2+2-formula in m propositional letters $q_1, \ldots, q_m$, and let $c_i = p_{i,1} \vee p_{i,2} \vee \neg n_{i,1} \vee \neg n_{i,2}$ for all $1 \leq i \leq n$. We construct an ABox $\mathcal{A}_{\varphi}$, an update α, and a goal ψ such that φ is satisfiable iff $\mathcal{A}^{\alpha} \not\models \psi$. The individual names in $\mathcal{A}_{\varphi}$ are f (which corresponds to the formula φ); the clauses $c_1, \ldots, c_n$; the propositional letters $q_1, \ldots, q_m$; $q_{\top}$ and $q_{\bot}$ (corresponding to the truth constants); and a and b, whose purpose will be explained later. Define the ABox $\mathcal{A}_\varphi$ as follows, where c, p_1, p_2, n_1, n_2, and t are role names:

$$\begin{aligned}\mathcal{A}_\varphi := &\{c(f,c_1),\dots,c(f,c_n)\}\\ &\cup \textstyle\bigcup_{1\le i\le n}\{p_1(c_i,p_{i,1}),p_2(c_i,p_{i,2}),n_1(c_i,n_{i,1}),n_2(c_i,n_{i,2})\}\\ &\cup \{(A\sqcap\overline{A})(a),A(b),t(q_\top,b),t(q_\bot,a)\}\\ &\cup \{\exists t.A(q_1),\dots,\exists t.A(q_m)\}\end{aligned}$$

Note that the first two lines in the definition of $\mathcal{A}_\varphi$ are a straightforward representation of φ. Define $\alpha = \{\neg A(a)\}$. Together with the last line of $\mathcal{A}_\varphi$, the execution of α induces a choice for each q_i, $1 \le i \le m$. To see this, let $\mathcal{I} \models \mathcal{A}_\varphi$ and $\mathcal{I} \Rightarrow_\alpha \mathcal{J}$. Then we have $q_i \in (\exists t.A)^\mathcal{I}$, and the t-successor of $q_i^\mathcal{I}$ that is in $A^\mathcal{I}$ can be $a^\mathcal{I}$ or not. If it is $a^\mathcal{I}$, we have $q_i \in (\exists t.\overline{A})^\mathcal{J}$. Thus, if $(\exists t.A)^\mathcal{J}$ and $(\exists t.\overline{A})^\mathcal{J}$ are disjoint, then $\mathcal{J}$ describes a truth assignment for $q_1,\dots,q_m$ as follows: q_i is true if $q_i^\mathcal{I} \in (\exists t.A)^\mathcal{I}$ and false if $q_i^\mathcal{I} \in (\exists t.\overline{A})^\mathcal{I}$. The use of a and b in $\mathcal{A}_\varphi$ ensure that $\top$ and $\bot$ are interpreted in the expected way (relying on the UNA). Executions of α in $\mathcal{A}_\varphi$ give us all possible truth assignments, encoded as interpretations $\mathcal{J}$. We use the goal $\psi = C(f)$ to express falsity of φ under these assignments:

$$C := \exists c.(\exists p_1.\exists t.\overline{A} \sqcap \exists p_2.\exists t.\overline{A} \sqcap \exists n_1.\exists t.A \sqcap \exists n_2.\exists t.A).$$

In the above explanation, we have assumed that $(\exists t.A)^\mathcal{J}$ and $(\exists t.\overline{A})^\mathcal{J}$ are disjoint. In general, this need not be the case, and also cannot be enforced. However, this does not affect the correctness of the reduction. Indeed, it is not hard to show that φ is satisfiable iff ψ is not a consequence of applying α in $\mathcal{A}_\varphi$, c.f. the proof of Lemma 2 in [10]. We remark that only the ABox $\mathcal{A}_\varphi$ depends on φ, whereas α and ψ do not. Thus, our lower bound even applies to data complexity, where only the ABox is considered as the input, but the action and goal are fixed.

Lemma 1. *There is an update α and an $\mathcal{EL}$-goal ψ such that, given an $\mathcal{EL}$-ABox $\mathcal{A}$, it is co-*NP*-hard to decide wheter $\mathcal{A}^\alpha \models \psi$.*

We now prove a co-NP upper bound for projection in $\mathcal{EL}^{(\neg)}$. The proof focusses on countermodels, which are defined as follows. Given an ABox $\mathcal{A}$, an action $\pi = \alpha_1,\dots,\alpha_k$, and a goal φ, we say that $\mathcal{I}_0,\dots,\mathcal{I}_k$ are *countermodels* against $\mathcal{A},\pi \models \varphi$ if $\mathcal{I}_0 \models \mathcal{A}$, $\mathcal{I}_i \Rightarrow_{\alpha_{i+1}} \mathcal{I}_{i+1}$ for $i < k$, and $\mathcal{I}_k \not\models \varphi$. Clearly, $\mathcal{A}^\pi \models \varphi$ iff there are no countermodels $\mathcal{I}_0,\dots,\mathcal{I}_k$ against $\mathcal{A},\pi \models \varphi$. The main ingredient to our upper bound proof is to show that if $\mathcal{A}^\pi \not\models \varphi$, then there are countermodels $\mathcal{I}_0,\dots,\mathcal{I}_k$ whose size is bounded by the size of $\mathcal{A}$ and π. This enables an NP-algorithm for non-projection that guesses interpretations $\mathcal{I}_0,\dots,\mathcal{I}_k$ of up to this size and then verifies in polynomial time whether $\mathcal{I}_0,\dots,\mathcal{I}_k$ are countermodels.

Lemma 2. *If $\mathcal{A}^\pi \not\models \varphi$, then there are countermodels $\mathcal{I}_0,\dots,\mathcal{I}_k$ against $\mathcal{A},\pi \models \varphi$ such that the cardinality of $|\Delta^{\mathcal{I}_i}|$ is bounded by the size of $\mathcal{A}$ and π for $i < k$.*

Proof. (sketch) If $\mathcal{A}^\pi \not\models \varphi$ with $\pi = \alpha_1, \ldots, \alpha_k$, then there are countermodels $\mathcal{I}_0, \ldots, \mathcal{I}_k$ against $\mathcal{A}, \pi \models \varphi$. We use selective filtration, as known from modal logic [5], to extract small countermodels $\mathcal{J}_0, \ldots, \mathcal{J}_k$ from $\mathcal{I}_0, \ldots, \mathcal{I}_k$. Let $\Delta^\mathcal{I}$ denote the (identical!) domain of $\mathcal{I}_0, \ldots, \mathcal{I}_k$. We select a subset $\Delta^\mathcal{J} \subseteq \Delta^\mathcal{I}$ whose cardinality is bounded by the size of $\mathcal{A}$ and π, and then $\mathcal{J}_0, \ldots, \mathcal{J}_k$ are simply the restriction of $\mathcal{I}_0, \ldots, \mathcal{I}_k$ to $\Delta^\mathcal{J}$. Let $\alpha_i = (\mathsf{pre}_i, \mathsf{post}_i)$. The elements of $\Delta^\mathcal{J}$ are as follows:

1. $a^{\mathcal{I}_0}$, for all individual names a in $\mathcal{A}$, π, and φ;
2. one witness from $\Delta^{\mathcal{I}_0}$ for each occurrence of a subconcept $\exists r.D$ in $\mathcal{A}$;
3. one witness from $\Delta^{\mathcal{I}_{i-1}}$ for each occurrence of a subconcept $\exists r.D$ in $\{\varphi \mid \varphi/\psi \in \mathsf{post}_i,\ \mathcal{I}_{i-1} \models \varphi\}$, for $1 \leq i \leq n$.

Individual names not occurring in $\mathcal{A}$, π, and φ are interpreted randomly in $\mathcal{J}_0, \ldots, \mathcal{J}_k$. $\square$

Since model checking in $\mathcal{EL}^{(\neg)}$ can be done in polynomial time and together with Lemma 1, we obtain the following result.

Theorem 1. *Projection in $\mathcal{EL}$ and $\mathcal{EL}^{(\neg)}$ with empty TBoxes is co-*NP*-complete.*

4 Projection in $\mathcal{EL}$ with acyclic TBoxes

We show that projection in $\mathcal{EL}$ and $\mathcal{EL}^{(\neg)}$ becomes PSpace-complete when acyclic TBoxes are admitted. For the lower bound, we reduce validity of quantified Boolean formulas (QBF) to projection in $\mathcal{EL}$. A QBF is of the form $\varphi = Q_1 p_1 . \ldots Q_n p_n . \vartheta$, where $Q_i \in \{\forall, \exists\}$, and ϑ is a propositional formula using only the propositional variables $p_1, \ldots, p_n$. We define validity of QBFs in terms of the existence of validation trees. A *validation tree* for a QBF $\varphi = Q_1 p_1 . \ldots Q_n p_n . \vartheta$ is a tree of depth n in which every level (except the leaves) corresponds to one of the quantifiers in φ. In $\forall p_i$-levels, each node has two successors, one for $p_i = \top$ and one for $p_i = \bot$. In $\exists p_i$-levels, each node has one successor, either for $p_i = \top$ or for $p_i = \bot$. Thus, every branch of a validation tree corresponds to a truth assignment to the variables $p_1, \ldots, p_n$, and it is required that the propositional formula ϑ evaluates to true on every branch. The QBF formula φ is *valid* iff there exists a validation tree for φ. It is known that validity of QBFs is PSpace-hard, even if the matrix formula ϑ is in CNF [17].

For the reduction, let $\varphi = Q_1 p_1 . \ldots Q_n p_n . \vartheta$ be a QBF with ϑ in CNF. We define an acyclic TBox $\mathcal{T}_\varphi$, ABox $\mathcal{A}_\varphi$, goal ψ_φ, and update α_φ such that $\mathcal{T}_\varphi, \mathcal{A}_\varphi^{\alpha_\varphi} \not\models \psi_\varphi$ iff φ is valid. As in Section 3, we call models $\mathcal{I}$ and $\mathcal{J}$ of $\mathcal{T}$ *countermodels* against $\mathcal{T}_\varphi, \mathcal{A}_\varphi^{\alpha_\varphi} \models \psi_\varphi$ iff $\mathcal{I} \models \mathcal{A}_\varphi$, $\mathcal{I} \Rightarrow^\mathcal{T}_{\alpha_\varphi} \mathcal{J}$, and $\mathcal{J} \not\models \psi_\varphi$. The general idea of the reduction is to achieve that, if $\mathcal{I}, \mathcal{J}$ are such countermodels, then $\mathcal{J}$ encodes a validation tree for φ. The purpose of the reduction TBox $\mathcal{T}_\varphi$ is to establish a tree structure in $\mathcal{I}$ and $\mathcal{J}$ that is the core of this encoding. In the tree structure, we use a role name r to represent the edges of the validation tree, and the concept names $L_0, \ldots, L_n$ to identify the n levels. The truth values of the variables $p_1, \ldots, p_n$ are (for now) represented via the concept names

$P_1, \ldots, P_n$ (indicating truth) and $\overline{P}_1, \ldots, \overline{P}_n$ (indicating falsity). We also use a role name t and concept names $A_1, \ldots, A_n$, $\overline{A}_1, \ldots, \overline{A}_n$, and $N_1, \ldots, N_n$, to be explained later. Define $\mathcal{T}_\varphi$ as follows:

$$\begin{array}{lll} \mathcal{T}_\varphi := \{ & L_i \doteq \exists r.(P_{i+1} \sqcap L_{i+1}) \sqcap \exists r.(\overline{P}_{i+1} \sqcap L_{i+1}) \sqcap N_i & i < n, Q_{i+1} = \forall \\ & L_i \doteq \exists r.L_{i+1} \sqcap N_i & i < n, Q_{i+1} = \exists \\ & L_n \doteq N_n & \\ & N_i \doteq \bigsqcap_{1 \leq j \leq i} \exists t.A_j & i \leq n \ \} \end{array}$$

The reduction ABox $\mathcal{A}_\varphi$ will include an assertion $L_0(a)$. The TBox $\mathcal{T}_\varphi$ thus establishes a binary tree of depth n rooted at $a^\mathcal{I}$ in $\mathcal{I}$ with the right number of successors at each level and with the concept names P_i, $\overline{P}_i$ set as required by the definition of validation trees in "universal" levels L_i (i.e., where $Q_i = \forall$). Since none of the L_i, P_i, $\overline{P}_i$, and r will occur in the update, $a^\mathcal{J}$ is a root of the same tree in $\mathcal{J}$. To make this tree a validation tree for φ, it remains to ensure that the tree in $\mathcal{J}$ satisfies the following:

(a) On every branch of the tree, every variable is interpreted as true or false (not yet guaranteed since both P_i and $\overline{P}_i$ may be false in a level L_i with $Q_i = \exists$).
(b) On no branch, a variable is interpreted as both true and false.
(c) Every branch describes a truth assignment that satisfies ϑ.

To enforce (a)-(c), we introduce a second representation of truth values, which is used as the main such representation from now on: $\exists t.A_j$ indicates truth of p_j and $\exists t.\overline{A}_j$ indicates falsity. In contrast to the representation via P_j and $\overline{P}_j$, in which the truth value of p_j is only stored at level j, the representation via $\exists t.A_j$ and $\exists t.\overline{A}_j$ stores the truth value of p_j at any level $i \geq j$. In particular, this means that the leaf of a branch stores the whole truth assignment associated with the branch, and thus we can ensure (c) locally at the leaves.

We start using the new representation by enforcing a central property:

($*$) If $d \in L_j^\mathcal{I}$, then $d \in (\exists t.A_i)^\mathcal{J}$ or $(\exists t.\overline{A}_i)^\mathcal{J}$ is true, for $1 \leq i \leq j \leq n$.

To establish ($*$), we exploit the same effect as in the co-NP-hardness proof in Section 3. More precisely, we use (i) the concepts N_i in $\mathcal{T}_\varphi$, (ii) assertions $(A_1 \sqcap \overline{A}_1)(b_1), \ldots, (A_n \sqcap \overline{A}_n)(b_n)$ in $\mathcal{A}_\varphi$, and (iii) the update, which is defined as

$$\alpha_\varphi := \{\neg A_1(b_1), \ldots, \neg A_n(b_n)\}.$$

Due to the use of the N_i concepts in $\mathcal{T}_\varphi$, $d \in L_j^\mathcal{I}$ satisfies $\exists t.A_j$ in $\mathcal{I}$, for $1 \leq i \leq j \leq n$. The choice in ($*$) then corresponds to whether or not the r-successor stipulated by this concept is b_j.

Obviously, ($*$) guarantees (a) for the second representation of truth values. The definition of α_φ explains why we cannot use this representation already in $\mathcal{T}_\varphi$. Namely, $\mathcal{T}_\varphi$ is used together with the assertion $L_0(a) \in \mathcal{A}_\varphi$, and thus talks about $\mathcal{I}$. Since $A_1, \ldots, A_n$ occur negated in α_φ, truth of concepts $\exists t.A_i$ and $\exists t.\overline{A}_i$ in $\mathcal{I}$ may be destroyed when moving with α_φ from $\mathcal{I}$ to $\mathcal{J}$.

We proceed by ensuring that every node $d \in L_i^\mathcal{I}$, $i \leq n$ satisfies the following three properties:

1. if d is in $P_i^{\mathcal{J}}$, then it is in $(\exists t.A_i)^{\mathcal{J}}$, and likewise for $\overline{P}_i$ and $\exists t.\overline{A}_i$;
2. d is in at most one of $(\exists t.A_j)^{\mathcal{J}}$ and $(\exists t.\overline{A}_j)^{\mathcal{J}}$, for $1 \leq j \leq i$.
3. if d is in $(\exists t.A_j)^{\mathcal{J}}$ with $1 \leq j \leq i$, then so are all its r-successors; and likewise for $\exists t.\overline{A}_j$;

Point 1 links the two representations of truth values, Point 2 addresses (b), and Point 3 ensures that truth values stored via the second representation are pushed down towards the leafs.

We define a set of concepts $\mathcal{C}$ such that, to enforce Points 1 to 3, it suffices to ensure that all concepts in $\mathcal{C}$ are false at the root of the validation tree in $\mathcal{J}$:

$$\begin{aligned}\mathcal{C} := \{&\exists r^i.(P_i \sqcap \exists t.\overline{A}_i), \exists r^i.(\overline{P}_i \sqcap \exists t.A_i) \mid 1 \leq i \leq n\} \cup \\ \{&\exists r^i.(\exists t.A_j \sqcap \exists r.\exists t.\overline{A}_j), \exists r^i.(\exists t.\overline{A}_j \sqcap \exists r.\exists t.A_j) \mid 1 \leq j \leq i < n\} \cup \\ \{&\exists r^i.(\exists t.A_j \sqcap \exists t.\overline{A}_j) \mid 1 \leq j \leq i \leq n\}\end{aligned}$$

Note that the i-th line in the definition of $\mathcal{C}$ corresponds to Point i above. Also note that the first two lines of $\mathcal{C}$ rely on $(*)$ to have the desired effect.

Before we describe how $\mathcal{C}$ can be incorporated into the reduction, let us describe how to ensure that, at every leaf, the formula ϑ evaluates to true. The idea is to come up with another set of concepts $\mathcal{D}$ that are made false at the root of the validation tree in $\mathcal{J}$. We use $\overline{\vartheta}$ to denote the dual of ϑ, i.e. the formula obtained from ϑ by swapping $\vee$ and $\wedge$ and p_i and $\neg p_i$, for $1 \leq i \leq n$. Obviously, $\overline{\vartheta}$ is equivalent to $\neg\vartheta$, $\overline{\vartheta}$ is of the same length as ϑ, and $\overline{\vartheta}$ is in DNF. Let $\overline{\vartheta} = \overline{\vartheta}_1 \vee \cdots \vee \overline{\vartheta}_m$, where the $\overline{\vartheta}_i$ are conjunctions of literals. Now $\mathcal{D}$ consists of the following concepts:

$$\overline{\vartheta}_i[p_j/\exists t.A_j, \neg p_j/\exists t.\overline{A}_j, \wedge/\sqcap] \text{ for } 1 \leq i \leq m.$$

Clearly, $\mathcal{D}$ is as required. If $\mathcal{EL}$ would include disjunction, we could now easily put $\mathcal{C}$ and $\mathcal{D}$ to work and thus finish the reduction by setting $\psi_\varphi := \bigsqcup_{C \in \mathcal{C} \cup \mathcal{D}} C(a)$, where a denotes the root of the validation tree. Since $\mathcal{J}$ is a part of a counter-model and thus violates ψ_φ, this has the desired effect that all concepts in $\mathcal{C} \cup \mathcal{D}$ are false at a. Alas, there is no disjunction in $\mathcal{EL}$ and we need to invest more work to employ $\mathcal{C}$ and $\mathcal{D}$.

We introduce individual names $a_0, a_1, \ldots, a_k$, where a_k denotes the root of the validation tree. Suppose we ensure that $\mathcal{J}$ is a model of the ABox

$$\mathcal{A} = \{s(a_i, a_{i+1}), s(a_i, a_k) \mid 0 \leq i < k\} \cup \{C_i(a_i), D_{i+1}(a_k) \mid 1 \leq i < k\}.$$

Then the structure of $\mathcal{J}$ is as shown in Figure 1. Let $\mathcal{C} \cup \mathcal{D} = \{C_1, \ldots, C_k\}$, and recursively define concepts $D_1, \ldots, D_k$ as follows:

$$\begin{aligned} D_i &:= \exists s.(C_i \sqcap D_{i+1}), \text{ for } 1 \leq i \leq k-1, \\ D_k &:= \exists s.C_k \end{aligned}$$

To enforce that all concepts $C_1, \ldots, C_k$ are false at a_k in the model $\mathcal{J}$ in Figure 1, we can choose

$$\psi_\varphi := D_1(a_0).$$

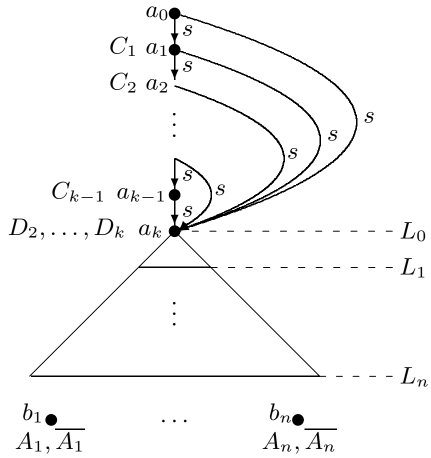

Fig. 1. Structure of the reduction ABox $\mathcal{A}_\psi$.

To see that this has the intended effect, note that the above choice of ψ_φ enforces that $\neg D_1 \equiv \forall s.(\neg C_1 \sqcup \neg D_2)$ is true at a_0. Thus, $\neg C_1$ is true at a_k and $\neg D_2$ is true at a_1. It remains to repeat this argument $k-1$ times.

Unfortunately, including the ABox $\mathcal{A}$ as part of $\mathcal{A}_\varphi$ does not result in $\mathcal{J}$ being a model of $\mathcal{A}$. The reason is that if the part of $\mathcal{I}$ that witnesses the truth of the assertions $C_i(a_i), D_{i+1}(a_k) \in \mathcal{A}_\varphi$ involves the individuals $b_1, \ldots, b_n$ that occur in α_φ, then these assertions may be invalidated by α_φ while transforming $\mathcal{I}$ to $\mathcal{J}$. The solution to this problem is as follows. For any concept C and individual name a, let $\mathsf{tree}_{C(a)}$ be an ABox that enforces a tree-shaped structure of connected individual names such that C is true at a. For example, if $C = \exists r.(A \sqcap \exists s.B \sqcap \exists r.(A \sqcap B))$, then

$$\mathsf{tree}_C(a) = \{r(a,c), A(c), s(c,c'), B(c'), r(c,c'')A(c''), B(c'')\}.$$

W.l.o.g., we assume that the individuals $b_1, \ldots, b_n$ do not occur in such ABoxes. In $\mathcal{A}_\varphi$, we now use ABoxes of the form $\mathsf{tree}_{C(a)}$ instead of the original assertions $C_i(a_i)$ and $D_{i+1}(a_k)$. Since $b_1, \ldots, b_n$ are the only individuals occurring in α_φ and we adopt the UNA, the generated structures are left untouched when α_φ transforms $\mathcal{I}$ into $\mathcal{J}$. Summing up, the ABox $\mathcal{A}_\varphi$ is thus as follows:

$$\begin{aligned}\mathcal{A}_\varphi := \{L_0(a_k)\} \cup \{(A_j \sqcap \overline{A}_j)(b_j) \mid 1 \leq j \leq n\} \cup \\ \{s(a_i, a_{i+1}), s(a_i, a_k) \mid i < k\} \cup \\ \bigcup_{1 \leq i < k} \mathsf{tree}_{C_i(a_i)} \cup \bigcup_{1 \leq i < k} \mathsf{tree}_{D_{i+1}(a_k)}\end{aligned}$$

Since the size of $\mathcal{T}_\varphi$, $\mathcal{A}_\varphi$, ψ_φ and α_φ is polynomial in n, we have established the intended PSPACE lower bound. A formal proof of correctness is given in [12].

A corresponding upper bound can be obtained from [3], where it is proved that projection in $\mathcal{ALC}$ is PSpace-complete w.r.t. acyclic TBoxes.

Theorem 2. *Projection in $\mathcal{EL}$ with acyclic TBoxes is* PSpace*-complete, even if only updates are allowed as actions.*

5 A Tractable Case

The results in the previous sections show that, in contrast to subsumption and other standard reasoning problems, projection in $\mathcal{EL}$ is not tractable. The purpose of this section is to identify a tractable case. From the perspective of modelling, this case is not very appealing. However, it may be used by a reasoner to avoid complex reasoning mechanisms when they are not really needed. As in the previous section, we admit acyclic TBoxes.

Let $\mathcal{T}$ be an acyclic TBox, $\mathcal{A}$ an ABox, $\pi = \alpha_1, \dots, \alpha_n$ an action with $\alpha_i = (\mathsf{pre}_i, \mathsf{post}_i)$ such that all post-conditions in post_i are unconditional, and $\psi = C(a)$ a goal. A symbol (i.e., concept or role name) σ is *special* if σ occurs negated in some post_i, $1 \leq i \leq n$. We say that $\mathcal{T}, \mathcal{A}, \pi, \psi$ are *nice* if the following condition is satisfied:

> if $\exists r.D$ is a subconcept of C that contains a special symbol, then there is no assertion $\exists r.E(b) \in \mathcal{A}$ such that $E \sqsubseteq_{\mathcal{T}} D$.

The aim of this section is to show that projection in $\mathcal{EL}$ w.r.t. acyclic TBoxes is tractable if the input $\mathcal{T}, \mathcal{A}, \pi, \psi$ is nice. Observe that this includes the case where negated literals in post-conditions are disallowed altogether.

We give a simple algorithm that runs in polynomial time. Let $\mathcal{T}, \mathcal{A}, \pi, \psi$ be a nice input, with $\pi = \alpha_1, \dots, \alpha_n$ and $\alpha_i = (\mathsf{pre}_i, \mathsf{post}_i)$. We assume w.l.o.g. that $\mathcal{A}$ does not contain any assertions of the form $C \sqcap D(a)$. If present, we may simply replace such an assertion with $C(a), D(a)$. Given an ABox $\mathcal{A}$ of this form and a set $\mathcal{L}$ of primitive literals, we write $\mathcal{A} \oplus \mathcal{L}$ to denote the ABox obtained from $\mathcal{A}$ by first removing the complement of every literal in $\mathcal{L}$ and then adding all literals in $\mathcal{L}$. The algorithm computes a sequence of ABoxes $\mathcal{A}_0, \dots, \mathcal{A}_n$ as follows:

- $\mathcal{A}_0 = \mathcal{A}$;
- $\mathcal{A}_{i+1} := \mathcal{A}_i \oplus \{\varphi \mid \top(a)/\varphi \in \mathsf{post}_{i+1}\}$.

The algorithm answers “yes” if $\mathcal{T}, \mathcal{A}_n \models \psi$ (i.e., ψ is true in every model of $\mathcal{T}$ and $\mathcal{A}_i$), and “no” otherwise. In $\mathcal{EL}$ with acyclic TBoxes and primitive negation, deciding whether $\mathcal{T}, \mathcal{A}_n \models \psi$ is tractable [2].

In the following, we briefly sketch the proof that our algorithm is correct. To show that $\mathcal{A}^\pi \not\models \psi$ implies $\mathcal{A}_n \not\models \psi$, we construct a sequence of interpretations $\mathcal{I}_0, \dots, \mathcal{I}_n$ such that, for any countermodel $\mathcal{J}_0, \dots, \mathcal{J}_n$ against $\mathcal{A}^\pi \models \psi$, $\mathcal{I}_i$ can be homomorphically embedded into $\mathcal{J}_i$, for all $i \leq n$. We then show that $\mathcal{A}^\pi \not\models \psi$ implies that $\mathcal{I}_n$ is a model of $\mathcal{A}_n$. To show that $\mathcal{A}_n \not\models \psi$ implies $\mathcal{A}^\pi \not\models \psi$, we unravel a model $\mathcal{J}$ of $\mathcal{A}_n \cup \{\neg\psi\}$ into a tree-like model $\mathcal{J}'$, and the define a countermodel $\mathcal{I}_0, \dots, \mathcal{I}_n$ against $\mathcal{A}^\pi \models \psi$ by setting $\mathcal{I}_n := \mathcal{J}'$ and applying the actions $\alpha_n, \dots, \alpha_1$ “backwards” to generate $\mathcal{I}_{n-1}, \dots, \mathcal{I}_0$.

Theorem 3. *Projection in $\mathcal{EL}$ w.r.t. acyclic TBoxes is tractable for nice inputs.*

We now show that there is no easy way to include conditional post-conditions in our tractable case. Let $\mathcal{CL}$ be the fragment of $\mathcal{EL}$ that admits only conjunction and concept names, but no existential restrictions. We prove that projection in $\mathcal{CL}$ is co-NP-hard, thus co-NP-complete by Theorem 2. Clearly, this also yields an alternative proof that projection in $\mathcal{EL}$ without TBoxes is co-NP-complete. However, unlike the proof in Section 3, the current proof relies on conditional post-conditions. This shows that there are two sources of intractability in projection in $\mathcal{EL}$: existential restrictions in the ABox as exploited in Section 3 and conditional post-conditions as exploited here.

Our proof is by reduction of 3SAT. Let $\varphi = c_1 \wedge \cdots \wedge c_k$ be a 3-formula in the variables $p_1, \ldots, p_n$ and $c_i = \ell_{i,1} \vee \ell_{i,2} \vee \ell_{i,3}$, for $1 \leq i \leq k$. We define an ABox $\mathcal{A}_\varphi$, a composite action $\alpha_\varphi = \alpha_{\varphi,1}; \alpha_{\varphi,2}; \alpha_{\varphi,3}$, and a goal ψ_φ such that φ is satisfiable iff $\mathcal{A}_\varphi^{\alpha_\varphi} \not\models \psi_\varphi$. We use the following concept names:

- the concept names $B_1, \ldots, B_n$ represent the truth values of $p_1, \ldots, p_n$ in the initial interpretation, where B_i means that p_i is true, and $\neg B_i$ that it is false;
- the concept names $A_1, \ldots, A_n$ and $\overline{A}_1, \ldots, \overline{A}_n$ represent the truth values of $p_1, \ldots, p_n$ in the interpretation obtained by executing $\alpha_{\varphi,1}$, where A_i means that p_i is true, and $\overline{A}_i$ that it is false;
- the concept names $F_1, \ldots, F_k$ indicates falsity of the clauses $c_1, \ldots, c_k$;
- the concept name F indicates falsity of φ.

The ingredients of the reduction are now defined as follows, where $\alpha_{\varphi,1} = (\emptyset, \mathsf{post}_i)$ for $i \in \{1, 2, 3\}$ and $\overline{L}_{i,j}$ denotes A_i if $\ell_{i,j} = \neg p_i$ and $\overline{A}_i$ if $\ell_{i,j} = p_i$:

$$\begin{aligned}
\mathcal{A}_\varphi &= \{\overline{A}_1(a), \ldots, \overline{A}_n(a)\} \\
\mathsf{post}_1 &= \{B_1(a)/A_1(a), B_1(a)/\neg\overline{A}_1(a), \ldots, B_n(a)/A_n(a), B_n(a)/\neg\overline{A}_n(a)\} \\
\mathsf{post}_2 &= \{\overline{L}_{i,1} \sqcap \overline{L}_{i,2} \sqcap \overline{L}_{i,3}(a)/F_i(a) \mid 1 \leq i \leq k\} \\
\mathsf{post}_3 &= \{F_1(a)/F(a), \ldots, F_k(a)/F(a)\} \\
\psi_\varphi &= F(a)
\end{aligned}$$

Intuitively, a valuation is “guessed” via $B_1, \ldots, B_n$ in the initial interpretation because the ABox $\mathcal{A}_\varphi$ does not specify the initial value of these concept names. Then $\alpha_{i,1}$ translates this representation to one in terms of $A_1, \ldots, A_n$ and $\overline{A}_1, \ldots, \overline{A}_n$, which allows us to say “p_i is false” in the ϑ part of post-conditions ϑ/χ of subsequent actions (where we cannot use $\neg B_i$!). The remaining actions $\alpha_{i,2}$ and $\alpha_{i,3}$ ensure that there is a clause in which all literals are false.

Theorem 4. *Projection in $\mathcal{CL}$ is co-*NP*-complete.*

This result shows that, when conditional post-conditions are allowed, projection is inherently intractable.

6 Conclusion

Our results show that, in $\mathcal{EL}$, tractability does not transfer from instance checking to projection. If no TBoxes are present, projection in $\mathcal{EL}$ is still simpler than in $\mathcal{ALC}$. This advantage is lost once that acyclic TBoxes are added. In the full version of this paper, we will additionally study planning with $\mathcal{EL}$-actions and show that most lower bounds for propositionally closed action formalisms also apply to $\mathcal{EL}$.

References

1. F. Baader. The instance problem and the most specific concept in the description logic $\mathcal{EL}$ w.r.t. terminological cycles with descriptive semantics. In *In Proc. of KI2003*, volume 2821 of *LNAI*, pages 64–78. Springer, 2003.
2. F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In *In Proc. of IJCAI-2005*, Edinburgh, UK, 2005. Morgan-Kaufmann Publishers.
3. F. Baader, C. Lutz, M. Milicic, U. Sattler, and F. Wolter. Integrating description logics and action formalisms: First results. In *Proc. of AAAI-05*, Pittsburgh, PA, USA, 2005.
4. F. Baader, C. Lutz, and B. Suntisrivaraporn. CEL—a polynomial-time reasoner for life science ontologies. In *Proc. of IJCAR'06*, volume 4130 of *LNAI*, pages 287–291. Springer, 2006.
5. P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic.* Cambridge University Press, 2001.
6. S. Brandt. Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and—what else? In *Proc. of ECAI-2004*, pages 298–302. IOS Press, 2004.
7. T. Bylander. The Computational Complexity of Propositional STRIPS Planning *Artificial Intelligence*, 69(1-2):165–204, 1994.
8. R. E. Fikes and N. J. Nilsson. STRIPS: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3-4):189–208, 1971.
9. Y. Gu and M. Soutchanski. Decidable Reasoning in a Modified Situation Calculus. In *Proc. of IJCAI-2007*, pages 1891-1897, 2007.
10. A. Krisnadhi and C. Lutz. Data complexity in the $\mathcal{EL}$ family of description logics. In *Proc. of LPAR2007*, volume 4790 of *LNAI*, pages 333–347. Springer, 2007.
11. M. Milicic. Complexity of Planning in Action Formalisms Based on Description Logics. In *Proc. of LPAR2007*, Springer, 2007.
12. M. Milicic. Action, Time and Space in Description Logics. PhD Thesis. TU Dresden, 2008. forthcoming.
13. R. Reiter. *Knowledge in Action.* MIT Press, 2001.
14. A. Schaerf. On the complexity of the instance checking problem in concept languages with existential quantification. *Journal of Intelligent Information Systems*, 2:265–278, 1993.
15. N. Sioutos, S. de Coronado, M. Haber, F. Hartel, W. Shaiu, and L. Wright. NCI thesaurus: a semantic model integrating cancer-related clinical and molecular information. *Journal of Biomedical Informatics*, 40(1):30–43, 2006.
16. K. Spackman. Normal forms for description logic expressions of clinical concepts in SNOMED RT. *Journal of the American Medical Informatics Association*, (Symposium Supplement), 2001.
17. L. J. Stockmeyer and A. R. Meyer. Word problems requiring exponential time. In *ACM Symposium on Theory of Computing (STOC '73)*, pages 1–9, New York, USA, 1973. ACM Press.