*Bull. Nov. Comp. Center*, Comp. Science, 34 (2012), 105–133


# Component properties of forgetting and progression in the situation calculus*

Denis Ponomaryov, Mikhail Soutchanski

**Abstract.** In many tasks related to reasoning about consequences of a logical theory, it is desirable to decompose the theory into a number of weakly-related or independent components. However, a theory may represent knowledge that is subject to change due to execution of actions that have effects on some properties mentioned in the theory. Having once computed a decomposition of a theory, one would like to know whether a decomposition has to be computed again in the theory obtained from taking into account changes resulting from execution of an action. In the paper, we address this problem in the scope of the situation calculus, where a change of an initial theory is related to the notion of progression. We undertake a study of the decomposability and inseparability properties known from the literature. We contribute by studying these properties wrt progression and the related notion of forgetting. We provide negative examples and identify cases when these properties are preserved under progression of initial theories and under forgetting in local–effect basic action theories of the situation calculus.

**Keywords:** decomposition, inseparability, forgetting, progression, basic action theory, situation calculus, reasoning about actions

## 1. Introduction

This paper is related to the decomposability and inseparability properties of logical theories widely known in research on modularization in the area of knowledge representation [16, 3, 4, 12]. Both properties are concerned with subdividing theories into components to facilitate reasoning. Informally, decomposability of a theory means that it can be equivalently represented as a union of two (or several) theories having a strictly defined set $\Delta$ of common signature symbols. Inseparability of theories in a logic wrt some signature $\Delta$ means that the theories have the same set of logical consequences in the signature $\Delta$. If a theory $\mathcal{T}$ is $\Delta$–decomposable into $\Delta$–inseparable components, then (under certain restrictions on the underlying logic) each component in decomposition contains all information from $\mathcal{T}$ in

*Supported under the Grant of the President of Russian Federation (Grant No. MK-2037.2011.9), the Russian Academy of Sciences (Grant No. 15/10), and the Siberian Division of the Russian Academy of Sciences (Integration Project No. 3). The authors would also like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Department of Computer Science of the Ryerson University for providing partial financial support.

the corresponding subsignature. This is an ideal case of decomposition, since in this case the problem of entailment from $\mathcal{T}$ can be reduced to entailment from components which are potentially smaller than the theory $\mathcal{T}$.

In the area of reasoning about actions, a logical theory represents knowledge that is subject to change due to effects of actions on some of the properties mentioned in the theory. It can be updated with new information, while some other knowledge should be forgotten as no longer true in the new situation. We consider two types of updates: forgetting in arbitrary theories and progression of theories in the situation calculus. Forgetting is a well-known operation on theories first introduced by Fangzhen Lin and Ray Reiter in their seminal paper [7]. Forgetting a signature $\sigma$ in a theory $\mathcal{T}$ means obtaining a theory indistinguishable from $\mathcal{T}$ in the rest of signature symbols $\mathtt{sig}(\mathcal{T}) \setminus \sigma$. In this sense, forgetting a signature is close to the well-known notion of uniform interpolation. Forgetting a ground atom $P(\bar{t})$ in a theory $\mathcal{T}$ gives a theory which implies all consequences of $\mathcal{T}$ "modulo" the truth value of $P(\bar{t})$. The operation of forgetting is closely related to progression in basic action theories in the situation calculus.

The situation calculus [17] is a knowledge representation formalism based on first-order logic which has been designed for axiomatization of problems in planning and high-level program execution. The idea is to axiomatize a set of initial states (as some initial theory), preconditions when actions can be performed, and the effects of actions on properties which are situation-dependent. Then, one can reason about consequences of sequences of actions, to check whether properties of interest hold in a chosen situation and whether a certain sequence of actions is executable. In the situation calculus, the so-called basic action theories represent such axiomatizations. Each basic action theory contains the initial theory which represents incomplete knowledge about the initial situation. In a special case, when there is complete knowledge about a finite number of individuals represented as unique names, the initial theory can be implemented as a relational database [17]. An update of the initial theory after execution of an action is called progression of the initial theory wrt an action. Informally, a basic action theory $\mathcal{D}$ is a union of some initial theory $\mathcal{D}_{S_0}$ and some theory $\mathcal{T}$ which defines transitions from one situation to another, plus some "canonical" axioms assumed to be true for all planning problems represented in situation calculus. Then progression of $\mathcal{D}_{S_0}$ wrt some action $\alpha$ is a logical consequence of $\mathcal{D}$ which contains all information from $\mathcal{D}$ about the situation resulting from execution of $\alpha$ in $S_0$. Ideally, it is computed as an update of $\mathcal{D}_{S_0}$ with some logical consequences of $\mathcal{T}$ after forgetting some information in $\mathcal{D}_{S_0}$ which is no longer true wrt the resulting situation. Note the intuitive relation with the operation of forgetting.

Historically, the situation calculus (earlier known as situational logic) is the earliest logical framework developed in artificial intelligence (AI). It is

still one of the most popular logical frameworks for reasoning about actions, e.g., it is presented in most well-known textbooks on AI. It has been developed in the 1960s by John McCarthy and his colleagues [13, 14, 1]. It is worth mentioning that there are both conceptual and technical differences between the situation calculus, which is designed for reasoning about arbitrary actions, and the Floyd–Hoare logic, Dijkstra's predicate transformers, dynamic logic (and other related formalisms), which are designed for reasoning about the correctness of computer programs. For example, the latter formalisms would consider the operator assigning a new value to a variable in a program as a primitive action, while the former would consider as primitive the actions on higher level of abstraction, e.g., such as moving a book from its current location to the table. For this reason, the situation calculus is chosen as foundation for high-level programming languages in cognitive robotics [6]. In our paper, when we talk about the situation calculus, we follow the axiomatic approach and notation developed by R.Reiter [17] who developed a general approach to axiomatizing direct effects and non-effects of actions. It has been observed for a long time that in practical applications real world actions have no effect on most properties. However, it was Reiter who first proposed an elegant axiomatization that represents compactly non-effects of actions. The cited book covers several extensions of the situation calculus to reasoning about concurrent actions, instantaneous actions, processes extended in time, interaction between action and knowledge, stochastic actions, as well as high-level programming languages based on the situation calculus. In our paper, we concentrate on the case when actions are sequential, atemporal, and deterministic. Despite this focus of our paper, our results can be subsequently adapted to characterize more general classes of actions. The main limitation of our work is in concentrating on direct effects only. Side effects of actions remain to be considered in future work.

We are interested in having an initial theory decomposable into inseparable components, and preserving decomposition and inseparability of components because the operation of progression can be computationally costly in practice. If an executed action has effects only on one part of the initial theory, then we would like to be able to compute progression using only this part instead of the whole initial theory. This leads to the question whether the decomposability and inseparability properties are preserved under progression and under forgetting. Ideally, we would like to avoid computing a decomposition of an updated initial theory again after executing an action. Moreover, we would like to know whether the components remain inseparable after progression. If yes, then it would suffice to compute a decomposition of the initial theory once, and this decomposition remains "stable" after progression wrt any arbitrary sequence of actions. In general, it is very hard to guarantee preservation of decomposability and inseparabil-

ity, because there is a certain conceptual distance between these notions on one hand, and forgetting and progression on the other – we provide examples witnessing this. Nevertheless, we identify important cases when preservation holds and it turns out that some of them have a nice-looking formulation.

We start with some model-theoretic remarks useful in this paper, then introduce the basics of situation calculus and proceed to the component properties of forgetting in Section 3 and progression in Section 4. The last section contains a summary of the obtained results. An extended version of the paper containing all proofs can be found at
https://docs.google.com/open?id=0Bx3LCrAhU9Q3bjRHcFhOSGRRcDQ

## 2. Background

### 2.1. Model-theoretic definitions

Let $\mathcal{L}$ be a logic (possibly many-sorted) which is a fragment of second-order logic (either by syntax or by translation of formulas) and has the standard model-theoretic Tarskian semantics. We call *signature* a subset of non-logical symbols of $\mathcal{L}$. If $\mathcal{M}_1$ and $\mathcal{M}_2$ are two many–sorted structures and $\Delta$ is a signature then we say that $\mathcal{M}_1$ and $\mathcal{M}_2$ agree on $\Delta$ if they have the same domains for each sort and the same interpretation of every symbol from $\Delta$. If $\mathcal{M}$ is a structure and $\sigma$ is a subset of predicate and function symbols from $\mathcal{M}$, then we denote by $\mathcal{M}\mid_\sigma$ the *reduct* of $\mathcal{M}$ to $\sigma$, i.e. the structure with predicate and function names from $\sigma$, where every symbol of $\sigma$ names the same entity as in $\mathcal{M}$. The structure $\mathcal{M}$ is called *expansion* of $\mathcal{M}\mid_\sigma$. For a set of formulas $\mathcal{T}$ in $\mathcal{L}$, we denote by $\mathtt{sig}(\mathcal{T})$ the signature of $\mathcal{T}$, i.e. the set of all non-logical symbols which occur in $\mathcal{T}$. We will use the same notation $\mathtt{sig}(\varphi)$ for the signature of a formula $\varphi$ in $\mathcal{L}$. If $t$ is a term in the language of second–order logic then the same notation $\mathtt{sig}(t)$ will be used for the signature of $t$, i.e. the set of all non-logical symbols occurring in $t$. Throughout this paper, we use the notion of *theory* as a synonym for a set of formulas in $\mathcal{L}$ which are sentences when translated into second-order logic. Whenever we mention a set of formulas, it is assumed that this set is in $\mathcal{L}$, if the context is not specified. For two theories $\mathcal{T}_1$ and $\mathcal{T}_2$, the notation $\mathcal{T}_1 \equiv \mathcal{T}_2$ will be the abbreviation for $\mathcal{T}_1 \models \mathcal{T}_2$ and $\mathcal{T}_2 \models \mathcal{T}_1$, i.e. the symbol $\equiv$ will mean semantic equivalence. If $\mathcal{T}$ is a set of formulas in $\mathcal{L}$ and $\Delta$ is a signature then $\mathtt{Cons}(\mathcal{T}, \Delta)$ will denote the set of (semantic) consequences of $\mathcal{T}$ (in $\mathcal{L}$) in the signature $\Delta$, i.e. the set $\{\varphi \in \mathcal{L} \mid \mathcal{T} \models \varphi \text{ and } \mathtt{sig}(\varphi) \subseteq \Delta\}$. We emphasize that this is a notation for a set of formulas in $\mathcal{L}$, because $\mathcal{T}$ may semantically entail formulas which are in second-order logic, but outside of $\mathcal{L}$.

Let us recall some basic model-theoretic facts that are important for understanding the results of this paper.

**Fact 1.** *If $\mathcal{T}$ is a theory in $\mathcal{L}$ and $\Delta$ is a signature, then some models of* $\mathtt{Cons}\,(\mathcal{T}, \Delta)$ *may not have expansion to a model of $\mathcal{T}$.*

Indeed, let $\mathcal{L}$ be first-order logic and $\{P, f\}$ be a signature, where $P$ is a unary predicate and $f$ is a unary function. Let $\mathcal{T}$ be a theory saying that $f$ is a bijection between the interpretation of $P$ and its complement. Thus, $\mathcal{T}$ axiomatizes the class of models, where the interpretation of $P$ and its complement are of the same cardinality. Then, by the Löwenheim–Skolem theorem, there is a model $\mathcal{M}$ of $\mathtt{Cons}\,(\mathcal{T}, \{P\})$ in which the interpretation of $P$ is a countable set, but the complement is uncountable. This model has no expansion to a model of $\mathcal{T}$.

**Fact 2.** *If $\mathcal{T}$ is a theory in $\mathcal{L}$ and $\Delta$ is a signature, then* $\mathtt{Cons}\,(\mathcal{T}, \Delta)$ *may not be finitely axiomatizable in $\mathcal{L}$.*

*Let $\mathcal{T}$ be the first-order theory axiomatized by the following two axioms:*
$\forall x[A(x) \to B(x)]$
$\forall x[B(x) \to \exists y R(x, y) \wedge B(y)]$,
*where $A$ and $B$ are unary predicates, $R$ is a binary predicate. Define the signature $\Delta = \{A, R\}$. Then* $\mathtt{Cons}\,(\mathcal{T}, \Delta)$ *is the following infinite set of formulas*
$\forall x\ A(x) \to \exists y R(x, y)$,
$\forall x\ A(x) \to [\exists y \exists u R(x, y) \wedge R(y, u)]$,
$\forall x\ A(x) \to [\exists y \exists u \exists v R(x, y) \wedge R(y, u) \wedge R(u, v)]$,
*...*
*By compactness, this theory is not finitely axiomatizable in first-order logic.*

There are plenty of known examples similar to the above mentioned, but we believe we have given the simplest ones. The example from Fact 2 is widely known in the literature on Description Logics (e.g., see Section 3.2 in [11]).

The well-known property of logics related to signature decompositions of theories is the Parallel Interpolation Property first considered in a partial case in [5] and studied later in a more general form in [3]:

**Definition 1** Parallel Interpolation Property. *The logic $\mathcal{L}$ is said to have the parallel interpolation property (PIP) if for any theories $\mathcal{T}_1$, $\mathcal{T}_2$ in $\mathcal{L}$ with* $\mathtt{sig}\,(\mathcal{T}_1) \cap \mathtt{sig}\,(\mathcal{T}_2) = \Delta$ *and any formula $\varphi$ in $\mathcal{L}$, the condition $\mathcal{T}_1 \cup \mathcal{T}_2 \models \varphi$ yields the existence of sets of formulas $\mathcal{T}_1'$ and $\mathcal{T}_2'$ in $\mathcal{L}$ such that:*

- $\mathcal{T}_i \models \mathcal{T}_i'$ *for $i = 1, 2$, and $\mathcal{T}_1' \cup \mathcal{T}_2' \models \varphi$;*
- $\mathtt{sig}\,(\mathcal{T}_i') \setminus \Delta \subseteq (\mathtt{sig}\,(\mathcal{T}_i) \cap \mathtt{sig}\,(\varphi)) \setminus \Delta$.

Next, we formulate two component properties of theories studied in this paper.

**Definition 2** $\Delta$–inseparability. *Theories $\mathcal{T}_1$ and $\mathcal{T}_2$ are called $\Delta$–inseparable for a signature $\Delta$, if $\mathtt{Cons}(\mathcal{T}_1, \Delta) = \mathtt{Cons}(\mathcal{T}_2, \Delta)$.*

*That is, no formula in the signature $\Delta$ "witnesses" any distinction between $\mathcal{T}_1$ and $\mathcal{T}_2$.*

The notion of inseparability in the scope of entailment has been intensively studied in Description Logics [4].

**Definition 3** $\Delta$–decomposability property. *Let $\mathcal{T}$ be a theory in $\mathcal{L}$ and $\Delta \subseteq \mathtt{sig}(\mathcal{T})$ be a subsignature. We call $\mathcal{T}$ $\Delta$–decomposable, if there are theories $\mathcal{T}_1$ and $\mathcal{T}_2$ in $\mathcal{L}$ such that*

- $\mathtt{sig}(\mathcal{T}_1) \cap \mathtt{sig}(\mathcal{T}_2) = \Delta$, *but* $\mathtt{sig}(\mathcal{T}_1) \neq \Delta \neq \mathtt{sig}(\mathcal{T}_2)$;
- $\mathtt{sig}(\mathcal{T}_1) \cup \mathtt{sig}(\mathcal{T}_2) = \mathtt{sig}(\mathcal{T})$;
- $\mathcal{T} \equiv \mathcal{T}_1 \cup \mathcal{T}_2$.

*The pair $\langle \mathcal{T}_1, \mathcal{T}_2 \rangle$ is called $\Delta$–decomposition of $\mathcal{T}$ and the theories $\mathcal{T}_1$ and $\mathcal{T}_2$ are called $\Delta$–decomposition components of $\mathcal{T}$. We will sometimes omit the word "decomposition" and call the sets $\mathcal{T}_1$ and $\mathcal{T}_2$ simply components of $\mathcal{T}$, when the signature $\Delta$ is clear from the context. The sets $\mathtt{sig}(\mathcal{T}_1) \setminus \Delta$ and $\mathtt{sig}(\mathcal{T}_2) \setminus \Delta$ are called signature ($\Delta$–decomposition) components of $\mathcal{T}$.*

The notion of $\Delta$–decomposition is defined above using a pair of theories, but easily extended to the case of a family of theories. It is important to realize that $\mathcal{T}_1$ and $\mathcal{T}_2$ need not be subsets of $\mathcal{T}$ in the above definition. Clearly, if $\mathcal{L}$ satisfies compactness and $\mathcal{T}$ is a finite $\Delta$–decomposable theory in $\mathcal{L}$ for a signature $\Delta$, then there is a $\Delta$–decomposition $\langle \mathcal{T}_1, \mathcal{T}_2 \rangle$ of $\mathcal{T}$, where $\mathcal{T}_1$ and $\mathcal{T}_2$ are finite. Although, by definition, the union $\mathcal{T}_1 \cup \mathcal{T}_2$ must entail all consequences of $\mathcal{T}$ in the signature $\Delta$, the components $\mathcal{T}_1$ and $\mathcal{T}_2$ may not be $\Delta$–inseparable, if we demand them to be finite. The set of $\Delta$–consequences of $\mathcal{T}_2$ may not be finitely axiomatizable in $\mathcal{L}$ by formulas in the signature $\mathtt{sig}(\mathcal{T}_1)$. This easily follows from Fact 2 which in fact notes that this effect is already possible in such weak languages as the sub-boolean description logic $\mathcal{EL}$. On the other hand, $\Delta$–inseparability of decompositions can always be obtained if the underlying logic $\mathcal{L}$ has uniform interpolation (cf. Proposition 2 in [16]).

It is easy to note that, in presence of PIP, decomposing a set $\mathcal{T}$ of formulas into inseparable components wrt a signature $\Delta$ gives a family of theories that imply all the consequences of $\mathcal{T}$ in the corresponding signatures.

**Fact 3.** *Let $\mathcal{L}$ have PIP, $\mathcal{T}$ be a theory in $\mathcal{L}$, and $\Delta$ be a signature. Let $\langle \mathcal{T}_1, \mathcal{T}_2 \rangle$ be a $\Delta$–decomposition of $\mathcal{T}$ with $\mathcal{T}_1$ and $\mathcal{T}_2$ being $\Delta$–inseparable. Then for any formula $\varphi$ with $\mathtt{sig}(\varphi) \subseteq \mathtt{sig}(\mathcal{T}_i)$, for some $i = 1, 2$, we have $\mathcal{T} \models \varphi$ iff $\mathcal{T}_i \models \varphi$.*

In other words, inseparable decomposition components can be used instead of the original theory for checking entailment of formulas in the corresponding signatures. This is the reason of our interest in the inseparability property in connection with decompositions.

### 2.2. Basics of the situation calculus

The language of the situation calculus $\mathcal{L}_{sc}$ has the first-order syntax over three sorts *action*, *situation*, *object* and is provided with the standard model-theoretic semantics. It is defined over the countably infinite alphabet $A_{sc} = \{do, \preceq, S_0, Poss\} \cup \mathcal{A} \cup \mathcal{F} \cup \mathcal{O} \cup \mathcal{P}$, where $do$ is a binary function symbol of sort situation, $\preceq$ is a binary relation on situations, $S_0$ is the constant of sort situation, $Poss(a, s)$ is a binary predicate (saying whether $a$ is possible in $s$) with the first argument of sort action and the second one of sort situation, $\mathcal{A}$ is a set of action functions with arguments of sort object, $\mathcal{F}$ is a set of so-called fluents, i.e. predicates having as arguments a tuple (vector) of sort object and one last argument of sort situation, $\mathcal{O}$ is a set of constants of sort object, and $\mathcal{P}$ is a set of static predicates and functions, i.e. those that only have objects as arguments. A symbol $v \in A_{sc}$ (predicate or function) is called *situation-independent* if $v \in A_{sc} \cup \mathcal{O} \cup \mathcal{P}$. A ground term is of sort situation iff it is either the constant $S_0$ or a term $do(A(\bar{t}), S)$, where $A(\bar{t})$ is a ground action term and $S$ is a ground situation term. For instance, a term $do(A_2(\bar{t_2}), do(A_1(\bar{t_1}), S_0))$ denotes the situation resulting from executing actions $A_1(\bar{t_1})$ and $A_2(\bar{t_2})$ consecutively from the initial situation $S_0$. Informally, static predicates specify object properties that do not change over time and fluents describe those object properties that are situation–dependent. The language of the situation calculus is used to formulate *basic action theories* ($\mathcal{BAT}$s). For example, they may serve as formal specifications of planning problems. Every $\mathcal{BAT}$ consists of a set of foundational axioms $\Sigma$ which specify constraints on how the function $do$ and fluents must be understood, a theory $D_{una}$ stating the unique name assumption for action functions and objects, an initial theory $D_{S_0}$ describing knowledge in the initial situation $S_0$, a theory $D_{ap}$ specifying preconditions of action execution, and a theory $D_{ss}$ (the set of successor-state axioms, SSAs for short) which contains definitions of fluents in the next situation in terms of static predicates and the values of fluents in the previous situation. More precisely, in every basic action theory $\mathcal{D}$ over a signature $\sigma \subseteq A_{sc}$, the theory $\Sigma$ is the set of the following axioms (note the axiom schema for induction):

$\forall\ a_1, a_2, s_1, s_2\ [do(a_1, s_2) = do(a_2, s_2) \rightarrow a_1 = a_2 \wedge s_1 = s_2]$
$\forall\ s\ \neg(s \preceq S_0 \wedge s \neq S_0)$
$\forall\ s_1, s_2\ [s_1 \preceq s_2 \leftrightarrow \exists a\ (do(a, s_1) \preceq s_2) \vee s_1 = s_2]$
$\forall P\ P(S_0) \wedge \forall a, s[P(s) \rightarrow P(do(a, s))] \rightarrow \forall s P(s)$

For every pair of distinct action functions $\{A, A'\} \subseteq \sigma$ and every pair $\langle a, b\rangle$ of distinct object constants from $\sigma$, a theory $D_{una}$ contains axioms of the form:

$a \neq b$
$\forall\, \bar{x}, \bar{y}\ A(\bar{x}) \neq A'(\bar{y})$
$\forall\, \bar{x}, \bar{y}\ A(x_1, \ldots, x_n) = A(y_1, \ldots, y_n) \rightarrow x_1 = y_1 \wedge \ldots \wedge x_n = y_n$ if $A$ is n–ary,

and no other axioms are in $D_{una}$. To define the remaining subtheories of $\mathcal{BAT}$, we need to introduce the following syntactic notion.

**Definition 4.** *A formula $\varphi$ in a language $\mathcal{L}_{sc}$ is called uniform in a situation term $s$ if:*

1. *it does not contain quantifiers over variables of sort situation;*
2. *it does not contain equalities between situation terms;*
3. *the predicates $Poss, \preceq$ do not occur in $\varphi$: $\{Poss, \preceq\} \cap \mathtt{sig}(\varphi) = \varnothing$;*
4. *for every fluent $F \in \mathtt{sig}(\varphi)$, the term in the situation argument of $F$ is $s$.*

*A set $\mathcal{T}$ of formulas in $\mathcal{L}_{sc}$ is called uniform in a situation term $s$ if every formula of $\mathcal{T}$ is uniform in $s$.*

By definition, a set $\mathcal{T}$ of formulas uniform in a situation term $S$ either does not contain any situation terms (and hence, fluents), or the only situation term is $S$ which occurs as the situation argument of each fluent from $\mathtt{sig}(\mathcal{T})$. If $\mathcal{T}$ is a set of sentences uniform in situation term $S$ (i.e., $\mathcal{T}$ has no free variables) and $S$ occurs in formulas of $\mathcal{T}$, then by items (1), (2) of the definition, $S$ must be ground and thus, it must either be the constant $S_0$, or have the form $do(A(\bar{t}), S')$, where $S'$ is a ground situation term. Note that if the constant $S_0$ or the binary function symbol $do$ is present in $\mathtt{sig}(\mathcal{T})$ and $\mathcal{T}$ is uniform in $S$, then necessarily $S_0 \in \mathtt{sig}(S)$, or $do \in \mathtt{sig}(S)$, respectively. By items (1) and (2), $\mathcal{T}$ does not restrict the interpretation of the term $S$ and the cardinality of the sort *situation*, so the observations above lead to the following property of uniform theories, which informally can be summarized by saying that, in sentences of a theory $\mathcal{T}$ uniform in a ground situation term $S$, we can understand this situation term as playing a role of an index that can remain implicit. Whenever we change the interpretation of $S$ (e.g., by choosing a different interpretation for $do$ and $S_0$) in a model of $\mathcal{T}$, it suffices to "move" interpretations of fluents to this new point to obtain again a model for $\mathcal{T}$.

**Lemma 1.** *Let $\mathcal{T}$ be a set of sentences uniform in a ground situation term $S$. Let $\mathcal{M} = \langle Act \cup Sit \cup Obj,\ \mathbf{do}, \mathbf{S_0}, \mathbf{F_1}, \ldots, \mathbf{F_n},\ \mathcal{I}\ \rangle$ be a model of $\mathcal{T}$, where Act, Sit, and Obj are domains for the corresponding sorts action, situation, and object, $\mathbf{do}$ and $\mathbf{S_0}$ are the interpretations of the function do and constant $S_0$, respectively, $\mathbf{F_1}, \ldots, \mathbf{F_n}$, $n \leqslant \omega$, are the interpretations of fluents from $\mathtt{sig}(\mathcal{T})$, and $\mathcal{I}$ is the interpretation of the rest of symbols from $\mathtt{sig}(\mathcal{T})$. For example, $\mathbf{F_i}$ is a set of tuples $\langle u_1, \ldots, u_{m-1}, \mathbf{S}\rangle$, where $\mathbf{S}$ is the interpretation of the ground term $S$ in $\mathcal{M}$.*

*Consider the structure $\mathcal{M}' = \langle Act \cup Sit' \cup Obj,\ \mathbf{do}', \mathbf{S_0}', \mathbf{F_1}', \ldots, \mathbf{F_n}',\ \mathcal{I}\ \rangle$, where $Sit'$ is an arbitrary set, the domain for sort situation, $\mathbf{do}'$ and $\mathbf{S_0}'$ are arbitrary interpretations of do and $S_0$ on $Sit'$, respectively, and for $i \leqslant n$, $\mathbf{F_i}'$ denotes the interpretation of a fluent $F_i$ as a set of tuples $\langle u_1, \ldots, u_{m-1}, \mathbf{S}'\rangle$, with $\mathbf{S}'$ being the interpretation of the term $S$ in $\mathcal{M}'$ and $\langle u_1, \ldots, u_{m-1}, \mathbf{S}\rangle \in \mathbf{F_i}$.*

*Then, $\mathcal{M}'$ is a model of $\mathcal{T}$. By definition, the interpretation of situation–independent predicates and functions is the same in $\mathcal{M}'$ and $\mathcal{M}$.*

If $S$ and $S'$ are two situation terms and $\mathcal{T}$ is a set of formulas uniform in $S$, then we denote by $\mathcal{T}(S/S')$ the set of formulas obtained from $\mathcal{T}$ by replacing every occurrence of $S$ with $S'$. This notation will be extensively used in Section 4. Obviously, $\mathcal{T}(S/S')$ is uniform in $S'$.

The initial theory $\mathcal{D}_{S_0}$ of $\mathcal{D}$ is defined as an arbitrary set of sentences in the signature $\sigma$ that are uniform in the situation constant $S_0$. Throughout the paper, we assume that $\mathcal{D}_{S_0}$ can be a theory in (any fragment of) second-order logic which can be translated into a set of sentences of first-order logic uniform in $S_0$. In particular, $\mathcal{D}_{S_0}$ can include both an ABox and a TBox in an appropriate Description Logic, as argued in [2, 19].

Next, for every n-ary action function $A \in \sigma$, a theory $\mathcal{D}_{ap}$ includes an axiom of the form

$$\forall\ \bar{x}, s\ [Poss(A(\bar{x}), s) \leftrightarrow \Pi_A(\bar{x}, s)],$$

where $\Pi_A(\bar{x}, s)$ is a formula uniform in $s$ with free variables among $\bar{x}$ and $s$. Informally, $\Pi_A(\bar{x}, s)$ characterizes preconditions for executing the action $A$ in the situation $s$. No other formulas are in $\mathcal{D}_{ap}$.

Finally, for every fluent $F \in \sigma$, a theory $D_{ss}$ contains an axiom of the form

$$\forall\ \bar{x}, a, s\ [F(\bar{x}, do(a, s)) \leftrightarrow \gamma_F^+(\bar{x}, a, s) \vee\ F(\bar{x}, s) \wedge \neg\gamma_F^-(\bar{x}, a, s)].$$

Here $\gamma_F^+$ is a disjunction of formulas of the form $[\exists \bar{y}](a = A^+(\bar{t}) \wedge \phi^+(\bar{x}, \bar{y}, s))$, where $A^+$ is an action function, $\bar{t}$ is a (possibly empty) vector of object terms with variables at most among $\bar{x}$ and $\bar{y}$, and $\phi^+$ is a formula uniform in $s$

with variables at most among $\bar{x}$, $\bar{y}$, and $s$. We write $[\exists \bar{y}]$ to show that $\exists \bar{y}$ is optional; it is present only if $\bar{t}$ includes $\bar{y}$ or if $\phi$ has an occurrence of $\bar{y}$. The formula $\phi^+$ is called a (positive) *context condition* meaning that $A^+(\bar{t})$ makes the fluent $F$ true if this context condition holds in $s$, but otherwise, $A^+(\bar{t})$ has no effect on $F$. Similarly, $\gamma_F^-$ is a disjunction of formulas of the form $[\exists \bar{z}](a = A^-(\bar{t}') \wedge \phi^-(\bar{x}, \bar{z}, s))$, where $A^-$ is an action function, $\bar{t}'$ is a (possibly empty) vector of object terms with variables at most among $\bar{x}$ and $\bar{z}$, and $\phi^-$ is a formula uniform in $s$ with variables at most among $\bar{x}$, $\bar{z}$, and $s$. The formula $\phi^-$ is called a (negative) *context condition* meaning that $A^-(\bar{t})$ makes the fluent $F$ false if this context condition holds in $s$, but otherwise, $A^-(\bar{t})$ has no effect on $F$. In the definition above, we assume that the empty disjunction is equal to *false*. No other formulas are in $\mathcal{D}_{ss}$. This completes the definition of $\mathcal{D}_{ss}$.

**Definition 5** SSA and active position of an action. *The axioms of $\mathcal{D}_{ss}$ in the form above are called successor state axioms (SSAs) of a basic action theory $\mathcal{D}$.*

*An action function $f$ is said to be in active position of some SSA $\varphi \in \mathcal{D}_{ss}$ if $f$ occurs either as $A^+$, or $A^-$ in the definition of $\mathcal{D}_{ss}$ above.*

*We say that $\varphi \in \mathcal{D}_{ss}$ is SSA for a fluent $F$ if $F$ is the fluent from the left-hand side of $\varphi$.*

Following the original consistency requirement on SSAs by Reiter (see Proposition 3.2.6 in [17]), we require that in case an action function $A^+$ occurs in active position in some disjunct of $\gamma^+$, then it must not occur in active position in $\gamma^-$. Analogously, if $A^-$ occurs in active position in $\gamma^-$, then it must not be in active position in $\gamma^+$. Informally, this means that an action cannot have both positive and negative effects on $F$.

Each SSA for a fluent $F$ completely defines the truth value of $F$ in the situation $do(a, s)$ in terms of what holds in the situation $s$. Also, SSA compactly represents non-effects by quantifying $\forall a$ over variables of sort action. Only action terms that occur explicitly on the right hand side of SSA for a fluent $F$ have effects on this fluent, while all other actions have no effect.

We note that the original version of Reiter's situation calculus admits functional fluents, e.g. functions having a vector of arguments of sort object and one last argument of sort situation. Reiter defines the notion of SSA for functional fluents in an appropriate form. We omit functional fluents in our version of the situation calculus.

**Proposition 1** Theorem 1 in [15]. *A basic action theory $\Sigma \cup D_{una} \cup D_{S_0} \cup D_{ap} \cup D_{ss}$ is satisfiable iff $D_{una} \cup D_{S_0}$ is satisfiable.*

Suppose $\alpha_1, \cdots, \alpha_n$ is a sequence of ground action terms, and $\varphi(s)$ is a formula with one free variable $s$ of sort situation which is uniform in $s$. One of the most important reasoning tasks in the situation calculus is the *projection problem*, that is, to determine whether

$$\mathcal{D} \models \varphi(do(\alpha_n, do(\alpha_{n-1}, do(\cdots, do(\alpha_1, S_0))))).$$

Informally, $\varphi$ represents some property of interest and entailment holds iff this property is true in the situation resulting from performing the sequence of actions $\alpha_1, \cdots, \alpha_n$ starting from $S_0$.

Another basic reasoning task is the *executability problem*. Let

$$executable(do(\alpha_n, do(\alpha_{n-1}, do(\cdots, do(\alpha_1, S_0)))))$$

be an abbreviation of the formula

$$Poss(\alpha_1, S_0) \wedge \bigwedge_{i=2}^{n} Poss(\alpha_i, do(\alpha_1, do(\cdots, do(\alpha_{i-1}, S_0)))).$$

Then, the executability problem is to determine whether

$$\mathcal{D} \models executable(do(\alpha_n, do(\alpha_{n-1}, do(\cdots, do(\alpha_1, S_0))))),$$

i.e. whether it is possible to perform the sequence of actions starting from $S_0$.

Planning and high-level program execution are two important settings, where the executability and projection problems arise naturally. *Regression* is a central computational mechanism that forms the basis for automated solution to the executability and projection tasks in the situation calculus ([17]). Regression requires reasoning backwards: a given formula

$$\varphi(do(\alpha_n, do(\alpha_{n-1}, do(\cdots, do(\alpha_1, S_0))))))$$

is recursively transformed into a logically equivalent formula by using SSAs until the resulting formula has only occurrences of the situation term $S_0$. It is easy to see that regression becomes computationally intractable if the sequence of actions grows indefinitely [2]. In this case, an alternative to regression is progression, which provides forward style reasoning. The initial theory $\mathcal{D}_{S_0}$ is updated to take into account effects of an executed action. Computing progression of a given theory $\mathcal{D}_{S_0}$ requires forgetting the facts in $\mathcal{D}_{S_0}$ which are no longer true after executing an action. The closely related notions of progression and forgetting are discussed in the next sections of our paper.

**Definition 6** local-effect SSA and $\mathcal{BAT}$. *SSA $\varphi \in \mathcal{D}_{ss}$ for the fluent $F$ is called local-effect if the set of arguments of every action function in active position of $\varphi$ contains all object variables from $F$. A basic action theory is said to be local-effect if every axiom of $\mathcal{D}_{ss}$ is a local-effect SSA.*

Local-effect $\mathcal{BAT}$s are a well-known class of theories for which the operation of progression (Section 4) can be computed effectively, even independently of decidability of the underlying theory itself. They are special in the sense that the truth value of each fluent defined by a local-effect SSA can change only for a finite set of objects after performing an action. Thus,

each action has only some local effect on fluents. This allows for employing forgetting, the operation considered in Section 3.

Finally, as an illustration we provide an example of a local-effect basic action theory in Situation Calculus describing the well-known Blocks World.

**Example 1** Example of $\mathcal{BAT}$: The Blocks World. The blocks world consists of a table and a finite set of blocks located either somewhere on a table or on top of each other. Any number of blocks can be on the table, but only one block can be on top of another block. In addition, there is a manipulator that can move a block from one location to another. In particular, it can move a block from the table on top of another block, or it can move a block from its current location to the table provided this block is not on the table already and there is nothing on the top of this block. In this example, we also make two common assumptions. First, all moving actions are deterministic, i.e., whenever the manipulator moves a block, it always succeeds: in the result of the action, the block will be at its destination, but not somewhere else. Second, there are no other agents (people, robots, manipulators, etc), who can move blocks.

We use the following action functions and fluents to axiomatize the blocks worlds in Situation Calculus:

**Actions**

- $move(x, y, z)$: Move block $x$ from block $y$ onto block $z$, provided both $x$ and $z$ are clear.
- $moveToTable(x, y)$: Move block $x$ from block $y$ onto the table, provided $x$ is clear and is not on the table.
- $moveFromTable(x, y)$: Move block $x$ from the table onto $y$, provided $y$ is clear.

**Fluents**

- $On(x, z, s)$: Block $x$ is on block $z$ in situation $s$.
- $Clear(x, s)$: Block $x$ has no other blocks on top of it in situation $s$.
- $Ontable(x, s)$: Block $x$ is on the table in situation $s$.

The subtheories of the corresponding basic action theory are defined as follows (all free variables are assumed to be universally quantified):

**Successor state axioms (theory $\mathcal{D}_{ss}$)**

$On(x,z,do(a,s)) \leftrightarrow \exists y(a{=}move(x,y,z)) \vee a{=}moveFromTable(x,z) \vee$
$On(x,z,s) \wedge a{\neq}moveToTable(x,z) \wedge \neg\exists y(a{=}move(x,z,y)).$

$Ontable(x,do(a,s)) \leftrightarrow \exists y(a{=}moveToTable(x,y)) \vee$
$Ontable(x,s) \wedge \neg\exists y(a{=}moveFromTable(x,y)).$

$Clear(x,do(a,s)) \leftrightarrow \exists y,z.On(y,x,s) \wedge$
$(a = move(y,x,z) \vee a = moveToTable(y,x)) \vee$
$Clear(x,s) \wedge \neg\exists w,y(a{=}move(w,y,x) \vee a{=}moveFromTable(y,x)).$

**Action precondition axioms (theory $\mathcal{D}_{ap}$)**

$Poss(move(x,y,z),s) \leftrightarrow Clear(x,s) \wedge Clear(z,s) \wedge x \neq z.$

$Poss(moveToTable(x,y),s) \leftrightarrow Clear(x,s) \wedge \neg Ontable(x,s).$

$Poss(moveFromTable(x,y),s) \leftrightarrow Clear(x,s) \wedge Clear(y,s) \wedge x \neq y \wedge$
$\neg Ontable(x,s).$

**Initial Theory ($\mathcal{D}_{S_0}$)** defined using the set of object constants $\{A,B,C\}$: $Clear(A,S_0)$, $On(A,B,S_0)$, $Clear(C,S_0)$, $Clear(x,S_0) \rightarrow \neg Ontable(x,S_0)$, $On(x,y,S_0) \rightarrow \neg Clear(y,S_0)$.

**Unique names axioms for actions and objects (theory $\mathcal{D}_{una}$)** is the set of unique-name axioms for all pairs of object constants and action functions used above.

Then $\Sigma \cup \mathcal{D}_{una} \cup \mathcal{D}_{ap} \cup \mathcal{D}_{ss} \cup \mathcal{D}_{S_0}$ is the resulting local-effect basic action theory.

## 3. Properties of forgetting

As progression is closely related to forgetting, we take a look at some properties of this operation first. Let us define a relation on structures as follows. Let $\sigma$ be a signature or a ground atom and $\mathcal{M}$, $\mathcal{M}'$ be two many–sorted structures. Then we set $\mathcal{M} \sim_\sigma \mathcal{M}'$ if:

- $\mathcal{M}$ and $\mathcal{M}'$ have the same domain for each sort;
- $\mathcal{M}$ and $\mathcal{M}'$ interpret all symbols which are not in $\sigma$ identically;
- if $\sigma$ is a ground atom $P(\bar{t})$ then $\mathcal{M}$ and $\mathcal{M}'$ agree on interpretation $\bar{u}$ of $\bar{t}$ and for every variable assignment $\theta$ with $\theta(\bar{x}) \neq \bar{u}$, we have $\mathcal{M},\theta \models P(\bar{x})$ iff $\mathcal{M}',\theta \models P(\bar{x})$.

Obviously, $\sim_\sigma$ is an equivalence relation.

**Definition 7** Forgetting an atom or a signature. *Let $\mathcal{T}$ be a theory in $\mathcal{L}$ and $\sigma$ be either a signature, or some ground atom. A set $\mathcal{T}'$ of formulas in a fragment of second-order logic is called the result of forgetting $\sigma$ in $\mathcal{T}$ (denoted by* `forget` *$(\mathcal{T},\sigma)$) if for any structure $\mathcal{M}'$, we have $\mathcal{M}' \models \mathcal{T}'$ iff there is a model $\mathcal{M} \models \mathcal{T}$ such that $\mathcal{M} \sim_\sigma \mathcal{M}'$.*

It is known that $\texttt{forget}(\mathcal{T}, \sigma)$ is always second–order definable for a finite set of formulas $\mathcal{T}$ in $\mathcal{L}$ and a signature or a ground atom $\sigma$ ([7], or Section 2.1 in [9]). On the other hand, the definition yields $\mathcal{T} \models \texttt{forget}(\mathcal{T}, \sigma)$, thus $\texttt{forget}(\mathcal{T}, \sigma)$ is a set of consequences of $\mathcal{T}$ which suggests that it may not always be definable in the logic where $\mathcal{T}$ is formulated and it may not be finitely axiomatizable in this logic, even if so is $\mathcal{T}$.

**Fact 4** Basic properties of forgetting. *If $\sigma$ and $\pi$ are signatures or ground atoms and $\mathcal{T}$, $\mathcal{T}'$ are theories in $\mathcal{L}$ then:*

- $\texttt{forget}(\mathcal{T}, \sigma \cup \pi) \equiv \texttt{forget}(\texttt{forget}(\mathcal{T}, \sigma), \pi)$ *(if $\sigma$ and $\pi$ are signatures)*
- $\texttt{forget}(\texttt{forget}(\mathcal{T}, \sigma), \pi) \equiv \texttt{forget}(\texttt{forget}(\mathcal{T}, \pi), \sigma)$
- $\texttt{forget}(\texttt{forget}(\mathcal{T}, \sigma), \sigma) \equiv \texttt{forget}(\mathcal{T}, \sigma)$
- $\texttt{forget}(\mathcal{T}, \sigma) \equiv \mathcal{T}$ *(if $\sigma$ is a signature with $\sigma \cap \texttt{sig}(\mathcal{T}) = \varnothing$, or a ground atom with predicate not contained in $\texttt{sig}(\mathcal{T})$ )*
- $\texttt{forget}(\mathcal{T} \cup \mathcal{T}', \sigma) \not\equiv \texttt{forget}(\mathcal{T}, \sigma) \wedge \texttt{forget}(\mathcal{T}', \sigma)$ *(see Example 3)*
- $\texttt{forget}(\varphi \vee \psi, \sigma) \equiv \texttt{forget}(\varphi, \sigma) \vee \texttt{forget}(\psi, \sigma)$ *(if $\varphi$ and $\psi$ are formulas in $\mathcal{L}$).*

**Proposition 2** Signature of forget($\mathcal{T}$,$\sigma$). *Let $\mathcal{T}$ be a theory in $\mathcal{L}$ and $\sigma$ a signature (or a ground atom, respectively); let $\texttt{forget}(\mathcal{T}, \sigma)$ be a set of formulas in a language $\mathcal{L}'$, a fragment of second-order logic with PIP. Then $\texttt{forget}(\mathcal{T}, \sigma)$ is logically equivalent in $\mathcal{L}'$ to a set of formulas in the signature $\texttt{sig}(\mathcal{T}) \setminus \sigma$ ($\texttt{sig}(\mathcal{T})$, respectively).*

**Corollary 1.** *Let $\mathcal{T}$ be a theory in $\mathcal{L}$ having PIP and $\sigma$ be a signature. Then $\mathcal{T} \equiv \texttt{forget}(\mathcal{T}, \sigma)$ iff $\mathcal{T}$ is equivalent to a set of formulas in the signature $\texttt{sig}(\mathcal{T}) \setminus \sigma$.*

We note that the similar statement does not hold when $\sigma$ is a ground atom. It follows from Proposition 2 that in case $\sigma$ is a signature, $\texttt{forget}(\mathcal{T}, \sigma)$ axiomatizes the class of reducts of models of $\mathcal{T}$ onto the signature $\texttt{sig}(\mathcal{T}) \setminus \sigma$. Clearly, if $\mathcal{T}$ is a theory in $\mathcal{L}$, then $\texttt{forget}(\mathcal{T}, \sigma)$ may not be in $\mathcal{L}$, however it is always expressible in second-order logic if $\mathcal{T}$ is finitely axiomatizable (note that second-order logic has PIP). For the case when $\sigma$ is a signature, $\texttt{forget}(\mathcal{T}, \sigma)$ is known as $\texttt{sig}(\mathcal{T}) \setminus \sigma$–uniform interpolant of $\mathcal{T}$ wrt the pair ($\mathcal{L}$, second-order logic), see Definition 13 in [4] and Lemma 39 in [12] for a justification. In other words, $\mathcal{T}$ and $\texttt{forget}(\mathcal{T}, \sigma)$ semantically entail the same second-order formulas in the signature $\mathcal{T} \setminus \sigma$.

If $\sigma$ is a ground atom $P(\bar{t})$ then, by definition, for any model $\mathcal{M} \models \mathcal{T}$, $\texttt{forget}(\mathcal{T}, \sigma)$ must have two "copies" of $\mathcal{M}$: a model with the value of $P(\bar{t})$

false and a model where this value is true. Let $\mathcal{L}$ be first-order logic. In contrast to forgetting a signature, for any recursively axiomatizable theory $\mathcal{T}$ in $\mathcal{L}$ and a ground atom $\sigma$, one can effectively construct the set of formulas $\texttt{forget}\,(\mathcal{T}, \sigma)$ in $\mathcal{L}$ such that $\texttt{forget}\,(\mathcal{T}, \sigma)$ is finitely axiomatizable iff $\mathcal{T}$ is. This follows from Theorem 4 in [7], where it is shown that forgetting a ground atom $P(\bar{t})$ in a theory $\mathcal{T}$ can be computed by simple syntactic manipulations:

- for an axiom $\varphi \in \mathcal{T}$, denote by $\varphi[P(\bar{t})]$ the result of replacing every occurrence of atom $P(\bar{t'})$ (with $\bar{t'}$ a term) by the formula $[\bar{t} = \bar{t'} \wedge P(\bar{t})] \vee [\bar{t} \neq \bar{t'} \wedge P(\bar{t'})]$
- denote by $\varphi^{+}[P(\bar{t})]$ the formula $\varphi[P(\bar{t})]$ with every occurrence of the ground atom $P(\bar{t})$ replaced with *true* and similarly, denote by $\varphi^{-}[P(\bar{t})]$ the formula $\varphi[P(\bar{t})]$ with $P(\bar{t})$ replaced with *false*
- then $\texttt{forget}\,(\mathcal{T}, P(\bar{t}))$ is equivalent to $(\bigwedge_{\varphi \in \mathcal{T}} \varphi^{+}[P(\bar{t})]) \bigvee (\bigwedge_{\varphi \in \mathcal{T}} \varphi^{-}[P(\bar{t})])$.

The disjunction corresponds to the union of two classes of models obtained from models of $\mathcal{T}$: with the ground atom $P(\bar{t})$ interpreted as *true* and *false*, respectively. This fact is important for effective computation of progression for local-effect $\mathcal{BAT}$s mentioned in Section 4. Note that if the theory $\mathcal{T}$ is finitely axiomatizable, computing $\texttt{forget}\,(\mathcal{T}, P(\bar{t}))$ in the way above doubles the size of the theory in the worst case. It is sometimes necessary to consider forgetting of some set $S$ of ground atoms in a theory $\mathcal{T}$. This is equivalent to iterative computation of forgetting of atoms from $S$ starting from the theory $\mathcal{T}$ (the order on atoms can be chosen arbitrary as noted in Fact 4). However, it is important to note that the size of the resulting theory is $O(2^{|S|} \times |\mathcal{T}|)$, where $|S|$ is the number of atoms in $S$ and $|\mathcal{T}|$ is the size of $\mathcal{T}$.

**Proposition 3** Interplay of forgetting and entailment. *Let $\mathcal{T}$ and $\mathcal{T}_1$ be two sets of formulas in $\mathcal{L}$ with $\mathcal{T} \models \mathcal{T}_1$ and $\sigma$ be a signature or a ground atom. Then the following holds:*

$$\begin{array}{ccc} \mathcal{T} & \models & \mathcal{T}_1 \\ \Vdots & & \Vdots \\ \texttt{forget}(\mathcal{T}, \sigma) & \models & \texttt{forget}(\mathcal{T}_1, \sigma) \end{array}$$

**Proposition 4** Preservation of consequences under forgetting. *Let $\mathcal{T}$ be a theory in $\mathcal{L}$ and $\sigma$ be either a signature or a ground atom. Let $\varphi$ be a formula such that either $\texttt{sig}\,(\varphi) \cap \sigma = \varnothing$ (in case $\sigma$ is a signature), or which does not contain the predicate from $\sigma$ (if $\sigma$ is a ground atom). Then $\mathcal{T} \models \varphi$ iff $\texttt{forget}\,(\mathcal{T}, \sigma) \models \varphi$.*

Now we provide results on preservation of inseparability under forgetting. By Proposition 4, when studying preservation of $\Delta$–inseparability of two sets of formulas for a signature $\Delta$, it is sufficient to consider the case of forgetting a subset of $\Delta$ or a ground atom with the predicate from $\Delta$, respectively.

**Proposition 5** Preservation of $\Delta$–insep. under signature forgetting. *Let $\mathcal{L}$ have PIP and $\mathcal{T}_1$ and $\mathcal{T}_2$ be two $\Delta$–inseparable sets of formulas in $\mathcal{L}$ with $\mathtt{sig}(\mathcal{T}_1) \cap \mathtt{sig}(\mathcal{T}_2) = \Delta$ for a signature $\Delta$. Let $\sigma$ be a subsignature of $\Delta$ and $\mathtt{forget}(\mathcal{T}_1, \sigma)$ and $\mathtt{forget}(\mathcal{T}_2, \sigma)$ be sets of formulas of $\mathcal{L}$. Then $\mathtt{forget}(\mathcal{T}_1, \sigma)$ and $\mathtt{forget}(\mathcal{T}_2, \sigma)$ are $\Delta$–inseparable.*

The following example demonstrates that a similar result does not hold under forgetting a ground atom with the predicate from $\Delta$.

**Example 2** $\Delta$–inseparability lost under forgetting a ground atom. We give an example of a logic $\mathcal{L}$, sets of formulas $\mathcal{T}_1$, $\mathcal{T}_2$ in $\mathcal{L}$, and a signature $\Delta = \mathtt{sig}(\mathcal{T}_1) \cap \mathtt{sig}(\mathcal{T}_2)$ such that $\mathcal{T}_1$ and $\mathcal{T}_2$ are $\Delta$–inseparable, but $\mathtt{forget}(\mathcal{T}_1, P(c,c))$ and $\mathtt{forget}(\mathcal{T}_2, P(c,c))$ are not, for a ground atom $P(c,c)$ with a predicate $P \in \Delta$. Let $\mathcal{L}$ be Description Logic $\mathcal{ELO}^{\bot}$, i.e. the sub-boolean logic $\mathcal{EL}$ augmented with nominals and the bottom concept $\bot$. Let $\Sigma = \{P, a, c\}$ be the signature, where $P$ is a role name (binary predicate) and $a, c$ are nominals (i.e. constants). Define the set of formulas $\mathcal{T}_1$ in the signature $\Sigma$ as $\{\{a\} \sqcap \{c\} \sqsubseteq \bot,\ \{c\} \sqsubseteq \exists P.\{a\},\ \top \sqsubseteq \exists P.\top\}$. Set $\Delta = \{P, c\}$ and consider the set of formulas $\mathcal{T}_2 = \{\top \sqsubseteq \exists P.\top,\ Taut(c)\}$, where $Taut(c)$ is a tautology with a nominal $c$ (for instance, the formula $\{c\} \sqsubseteq \top$). We have $\mathtt{sig}(\mathcal{T}_1) \cap \mathtt{sig}(\mathcal{T}_2) = \Delta$ and it is easy to check that $\mathcal{T}_2$ is equivalent to $\mathtt{Cons}(\mathcal{T}_1, \Delta)$ in the logic $\mathcal{ELO}^{\bot}$; thus, $\mathcal{T}_1$ and $\mathcal{T}_2$ are $\Delta$–inseparable. Now consider $\mathtt{forget}(\mathcal{T}_1, P(c,c))$ and $\mathtt{forget}(\mathcal{T}_2,\ P(c,c))$ as sets of formulas in second-order logic (we assume the standard translation of formulas of $\mathcal{ELO}^{\bot}$ into the language of second-order logic). We verify that they are not $\Delta$–inseparable and the formula $\top \sqsubseteq \exists P.\top$ is the evidence for this. By definition of $\mathcal{T}_1$, we have $\mathtt{forget}(\mathcal{T}_1, P(c,c)) \models \mathcal{T}_1$, since any model of $\mathcal{T}_1$ with a changed truth value of the predicate $P$ on the pair $\langle c, c\rangle$ is still a model of $\mathcal{T}_1$. On the other hand, $\mathtt{forget}(\mathcal{T}_2, P(c,c)) \not\models \top \sqsubseteq \exists P.\top$, because $\mathcal{T}_2$ has the one–element model $\mathcal{M}$, where $P$ is reflexive (on the sole element corresponding to $c$). Hence, by definition of forgetting, the one–element model $\mathcal{M}'$ with $P$ false on the pair $\langle c, c\rangle$ must be a model of $\mathtt{forget}(\mathcal{T}_2, P(c,c))$, but obviously, $\mathcal{M}' \not\models \top \sqsubseteq \exists P.\top$.

It turns out that preservation of inseparability under forgetting a ground atom requires rather strong model–theoretic conditions like (*) in Proposition 6 below. Specialists might notice that (*) is equivalent to *semantic* $\Delta$–*inseparability* of the initial sets of formulas (see Definition 11 in [4]) which is far from being decided effectively from the computational point of view

(see Theorem 3 in [10], Lemma 40 in [12]). Semantic $\Delta$–inseparability is strictly stronger than (syntactic) inseparability from Definition 2. On the other hand, Proposition 6 says that whenever there is a chance to satisfy (*) for two given sets of formulas, one does not need to check it again after forgetting something in their common signature. To compare condition (*) with Example 2, note that the mentioned one–element model of $\mathcal{T}_2$ does not expand to a model of $\mathcal{T}_1 \cup \mathcal{T}_2$.

**Proposition 6** Preservation of $\Delta$–inseparability under forgetting. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two sets of formulas in $\mathcal{L}$ with $\mathtt{sig}(\mathcal{T}_1) \cap \mathtt{sig}(\mathcal{T}_2) = \Delta$ for a signature $\Delta$ which satisfy the following condition (*): for $i = 1, 2$, any model of $\mathcal{T}_i$ can be expanded to a model of $\mathcal{T}_1 \cup \mathcal{T}_2$. Then:*

- *$\mathcal{T}_1$ and $\mathcal{T}_2$ are $\Delta$–inseparable;*
- *for $\sigma$, a signature or a ground atom, $\mathtt{forget}(\mathcal{T}_1, \sigma)$ and $\mathtt{forget}(\mathcal{T}_2, \sigma)$ satisfy (*) as well.*

Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two sets of formulas in $\mathcal{L}$ with $\mathtt{sig}(\mathcal{T}_1) \cap \mathtt{sig}(\mathcal{T}_2) = \Delta$ for a signature $\Delta$ and let $\sigma$ be either a subsignature of $\Delta$ or a ground atom with the predicate from $\Delta$. It is known that, in general, forgetting $\sigma$ may not be distributive over the union of sets of formulas. The entailment $\mathtt{forget}(\mathcal{T}_1 \cup \mathcal{T}_2, \sigma) \models \mathtt{forget}(\mathcal{T}_1, \sigma) \cup \mathtt{forget}(\mathcal{T}_2, \sigma)$ holds by Proposition 3, but Example 3 below easily shows that even strong semantic conditions related to modularity do not guarantee the reverse entailment. On the other hand, forgetting something outside of the common signature of $\mathcal{T}_1$ and $\mathcal{T}_2$ is distributive over union, as formulated in Corollary 2 which is a consequence of the criterion in Proposition 7.

**Example 3** Failure of componentwise forgetting in $\Delta$. Let $\mathcal{L}$ be first-order logic and $\Delta = \{P, c\}$ be the signature consisting of a unary predicate $P$ and a constant $c$. Define theories $\mathcal{T}_1$ and $\mathcal{T}_2$ as: $\mathcal{T}_1 = \{A \to P(c)\}$, $\mathcal{T}_2 = \{P(c) \to B\}$, where $A, B$ are nullary predicate symbols. We have $\mathtt{sig}(\mathcal{T}_1) \cap \mathtt{sig}(\mathcal{T}_2) = \Delta$ and for $i = 1, 2$, any model of $\mathcal{T}_i$ can be expanded to a model of $\mathcal{T}_1 \cup \mathcal{T}_2$. Clearly, $\mathcal{T}_1$ and $\mathcal{T}_2$ are $\Delta$–inseparable and for $i = 1, 2$, $\mathtt{Cons}(\mathcal{T}_i, \Delta)$ is the set of tautologies in $\Delta$. By definition of forgetting, for $i = 1, 2$, $\mathtt{forget}(\mathcal{T}_i, P(c))$ is a set of tautologies and thus, $\mathtt{forget}(\mathcal{T}_1, P(c)) \cup \mathtt{forget}(\mathcal{T}_2, P(c)) \not\models \mathtt{forget}(\mathcal{T}_1 \cup \mathcal{T}_2, P(c))$, because $\mathtt{forget}(\mathcal{T}_1 \cup \mathcal{T}_2, P(c)) \models A \to B$ (by Proposition 4). For the case of forgetting a signature, say a nullary predicate $P$, it suffices to consider $\Delta = \{P\}$ and theories $\mathcal{T}_1 = \{A \to P\}$, $\mathcal{T}_2 = \{P \to B\}$, where $A, B$ are nullary predicates.

**Proposition 7** A criterion for componentwise forgetting. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two sets of formulas and $\sigma$ be either a signature or a ground atom. Then the following statements are equivalent:*

- $\texttt{forget}\,(\mathcal{T}_1, \sigma) \cup \texttt{forget}\,(\mathcal{T}_2, \sigma) \models \texttt{forget}\,(\mathcal{T}_1 \cup \mathcal{T}_2, \sigma)$
- *for any two models* $\mathcal{M}_1 \models \mathcal{T}_1$ *and* $\mathcal{M}_2 \models \mathcal{T}_2$*, with* $\mathcal{M}_1 \sim_\sigma \mathcal{M}_2$*, there exists a model* $\mathcal{M} \models \mathcal{T}_1 \cup \mathcal{T}_2$ *such that* $\mathcal{M} \sim_\sigma \mathcal{M}_i$ *for some* $i = 1, 2$*.*

To compare this criterion with Example 3, observe that there exist models $\mathcal{M}_1 \models \mathcal{T}_1$ and $\mathcal{M}_2 \models \mathcal{T}_2$ with a common domain such that $\mathcal{M}_1 \models A \wedge P(c) \wedge \neg B$ and $\mathcal{M}_2 \models A \wedge \neg P(c) \wedge \neg B$. Thus, $\mathcal{M}_1 \sim_{P(c)} \mathcal{M}_2$, however, there does not exist a model $\mathcal{M}$ of $\mathcal{T}_1 \cup \mathcal{T}_2$ such that $\mathcal{M} \sim_{P(c)} \mathcal{M}_i$ for some $i = 1, 2$. Neither $\mathcal{M}_1$, nor $\mathcal{M}_2$ is a model for $\mathcal{T}_1 \cup \mathcal{T}_2$.

**Corollary 2** Forgetting in the scope of one component. *Let* $\mathcal{T}_1$ *and* $\mathcal{T}_2$ *be two sets of formulas with* $\texttt{sig}\,(\mathcal{T}_1) \cap \texttt{sig}\,(\mathcal{T}_2) = \Delta$ *for a signature* $\Delta$ *and* $\sigma$ *be either a subsignature of* $\texttt{sig}\,(\mathcal{T}_1) \setminus \Delta$ *or a ground atom with the predicate from* $\texttt{sig}\,(\mathcal{T}_1) \setminus \Delta$*. Then* $\texttt{forget}\,(\mathcal{T}_1 \cup \mathcal{T}_2, \sigma)$ *is equivalent to* $\texttt{forget}\,(\mathcal{T}_1, \sigma) \cup \texttt{forget}\,(\mathcal{T}_2, \sigma)$*. Moreover, if* $\mathcal{T}_1$ *and* $\mathcal{T}_2$ *are* $\Delta$*–inseparable, then so are* $\texttt{forget}\,(\mathcal{T}_1, \sigma)$ *and* $\texttt{forget}\,(\mathcal{T}_2, \sigma)$*.*

In general, the results of this section prove that the operation of forgetting does not behave well wrt syntactic modularity properties of the input. Stronger model-theoretic conditions on the input are needed due to the model-theoretic nature of forgetting.

## 4. Properties of progression

We have considered some component properties of forgetting. It turns out that the operation of progression is closely related to forgetting in initial theories. However, in case of progression, we can not restrict ourselves to working with initial theories only; we need also to take into account information from successor state axioms. The aim of this section is to demonstrate some component properties of progression wrt different forms of SSAs and common signatures $\Delta$'s (*deltas*) of components of initial theories. We will consider local-effect SSAs discussed in [9] and *deltas*, which do not contain fluents.

We use the following notations further in this paper. For a ground action term $\alpha$ in the language of the situation calculus, we denote by $S_\alpha$ the situation term $do(\alpha, S_0)$.

To define progression, let us introduce an equivalence relation on many-sorted structures in the situation calculus signature. For two structures $\mathcal{M}$, $\mathcal{M}'$ and a ground action $\alpha$, we set $\mathcal{M} \sim_{S_\alpha} \mathcal{M}'$ if:

- $\mathcal{M}$ and $\mathcal{M}'$ have the same sorts for action and object;

- $\mathcal{M}$ and $\mathcal{M}'$ interpret all situation-independent predicate and function symbols identically;
- $\mathcal{M}$ and $\mathcal{M}'$ agree on interpretation of all fluents at $S_\alpha$, i.e. for every fluent $F$ and every variable assignment $\theta$, we have $\mathcal{M}, \theta \models F(\bar{x}, S_\alpha)$ iff $\mathcal{M}', \theta \models F(\bar{x}, S_\alpha)$.

**Definition 8.** *Let $\mathcal{D}$ be a basic action theory with unique name axioms $\mathcal{D}_{una}$ and the initial theory $\mathcal{D}_{S_0}$ and let $\alpha$ be a ground action term. A set $\mathcal{D}_{S_\alpha}$ of formulas in a fragment of second-order logic is called progression of $\mathcal{D}_{S_0}$ wrt $\alpha$ if it is uniform in the situation term $S_\alpha$ and for any structure $\mathcal{M}$, $\mathcal{M}$ is a model of $\mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha}$ iff there is a model $\mathcal{M}'$ of $\mathcal{D}$ such that $\mathcal{M} \sim_{S_\alpha} \mathcal{M}'$.*

Below, we use $\mathcal{D}_{S_\alpha}$ to denote progression of the initial theory wrt the action term $\alpha$, if the context of $\mathcal{BAT}$ is clear. We sometimes abuse terminology and call progression not only the theory $\mathcal{D}_{S_\alpha}$, but also the operation of computing this theory (when existence of an effective operation is implicitly assumed).

For any $\mathcal{BAT}$ $\mathcal{D}$, we have $\mathcal{D} \models \mathcal{D}_{S_\alpha}$ and, similarly to the operation of forgetting, it is possible to provide an example (see Definition 2, Conjecture 1, and Theorem 2 in [18]), when the progression $\mathcal{D}_{S_\alpha}$ is not definable (even by an infinite set of formulas) in the logic in which $\mathcal{D}$ is formulated. On the other hand, it can be seen (Theorem 2.10 in [9]) that progression is always second-order definable and is finitely axiomatizable in second-order logic if the signature of $\mathcal{BAT}$ is finite and the initial theory $\mathcal{D}_{S_0}$ is finitely axiomatizable.

To understand the notion of progression intuitively, note the following. The progression $\mathcal{D}_{S_\alpha}$ is a set of consequences of $\mathcal{BAT}$ which are uniform in the situation term $S_\alpha$, thus informally, $\mathcal{D}_{S_\alpha}$ is information about the situation $S_\alpha$ implied by $\mathcal{BAT}$. Moreover, it contains all information from $\mathcal{BAT}$ about the situation $S_\alpha$, as guaranteed by the model-theoretic property with relation $\sim_{S_\alpha}$ in the definition. Recall that the initial theory of $\mathcal{BAT}$ describes information in the initial situation $S_0$ and SSAs are essentially the rules for obtaining new definitions of fluents after performing actions. Thus, progression $\mathcal{D}_{S_\alpha}$ can be viewed as "modification" of the initial theory obtained after executing the action $\alpha$. In particular, the initial theory of $\mathcal{BAT}$ can be replaced with $\mathcal{D}_{S_\alpha}(S_\alpha/S_0)$ (recall the notation from Section 2.2) which gives a new $\mathcal{BAT}$, with $S_\alpha$ as the initial situation. To check whether a certain property, a formula $\varphi(s)$ uniform in a situation variable $s$, holds in the situation $S_\alpha$ wrt $\mathcal{BAT}$ $\mathcal{D}$, one may try to compute the progression $\mathcal{D}_{S_\alpha}$ and check whether $\mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha} \models \varphi(S_\alpha)$ (or equivalently, $\mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha}(S_\alpha/S_0) \models \varphi(S_0)$) holds. By Proposition 1, this is equivalent to $\mathcal{D} \models \varphi(S_\alpha)$.

Of interest are cases when progression can be computed effectively as a theory in the same logic which is used to formulate underlying $\mathcal{D}_{S_0}$, independently of the fact whether satisfiability in this logic is decidable. The well-known approach is to consider the local-effect $\mathcal{BAT}$s (recall Definition 6) in which progression can be obtained by just a syntactic modification of the initial theory $\mathcal{D}_{S_0}$ with respect to SSAs. The approach is based on effective forgetting of a certain set of ground atoms (extracted from SSAs) in the initial theory of $\mathcal{BAT}$. Recall the well–known observation from Section 3 that, given a theory $\mathcal{T}$ (in an appropriate logic $\mathcal{L}$), forgetting a set of ground atoms in $\mathcal{T}$ can be computed effectively by straightforward syntactic manipulations with the axioms of $\mathcal{T}$. Thus, the essence of computing progression in the local-effect case is to extract effectively the set of ground atoms from SSAs that need to be forgotten. Subsequently, in $\mathcal{D}_{S_\alpha}$, they are replaced with new values of fluents; the new values are computed from SSAs. An interested reader may consult the whole paper [9], while here we only introduce necessary notations from Definition 3.4 of [9] which will be used in Theorem 2.

Let $\mathcal{D}$ be a $\mathcal{BAT}$ with a set $\mathcal{D}_{ss}$ of SSAs, an initial theory $\mathcal{D}_{S_0}$, and a unique name assumption theory $\mathcal{D}_{una}$, and let $\alpha$ be a ground action term. Denote

$\Delta_F = \{\bar{t} \mid \bar{x} = \bar{t}$ appears in $\gamma^+_F(\bar{x}, \alpha, s)$ or $\gamma^-_F(\bar{x}, \alpha, s)$ from an SSA $\varphi \in \mathcal{D}_{ss}$instantiated with $\alpha$ and equivalently rewritten wrt $\mathcal{D}_{una}\}$,

$\Omega(s) = \{F(\bar{t}, s) \mid \bar{t} \in \Delta_F\}$

Note that $\Omega(S_0)$ is a finite set of ground actions to be forgotten. According to Fact 4, forgetting several ground atoms can be accomplished consecutively in any order.

An *instantiation* of $\mathcal{D}_{ss}$ wrt $\Omega(S_0)$, denoted by $\mathcal{D}_{ss}[\Omega(S_0)]$, is the set of formulas of the form:

$$F(\bar{t}, do(\alpha, S_0)) \leftrightarrow \gamma^+_F(\bar{t}, \alpha, S_0) \vee \ F(\bar{t}, S_0) \wedge \neg\gamma^-_F(\bar{t}, \alpha, S_0).$$

Observe that $\mathcal{D}_{ss}[\Omega(S_0)]$ effectively defines new values for those fluents which are affected by the action $\alpha$. However, these definitions use fluents wrt $S_0$, which may include fluents to be forgotten. For this reason, forgetting should be performed not only in $\mathcal{D}_{S_0}$, but in $\mathcal{D}_{ss}[\Omega(S_0)]$ as well.

**Proposition 8** Theorem 3.6 in [9]. *In the notations above, the following is a progression of $\mathcal{D}_{S_0}$ wrt $\alpha$ in the sense of Definition 8:*

$$\mathcal{D}_{S_\alpha} = [\ forget(\mathcal{D}_{ss}[\Omega(S_0)] \cup \mathcal{D}_{S_0},\ \Omega(S_0))\ ](S_0/S_\alpha).$$

Thus, computing a progression in a local-effect $\mathcal{BAT}$ is an effective syntactic transformation of the initial theory, which leads to the *unique* form

of the updated theory $\mathcal{D}_{S_\alpha}$. This fact will be used in Theorem 2. It is important to realize that this transformation can lead to exponential blow-up of the initial theory, as noted after Theorem 3.6 in [9], due to the possible exponential blow-up after forgetting a set of ground atoms. This is not a surprise, because even in propositional logic, forgetting a symbol in a formula is essentially elimination of a "middle term" (introduced by Boole), which results in the disjunction of two instances of the input formula [8]. As a consequence, forgetting may result in a formula that is roughly twice as long as the input formula. It is important to realize that the exponential blowup is not inevitable in the case of progression. As shown in [9], there are practical classes of the initial theories for which there is no blow-up and the size of progressed theory is actually linear wrt the size of the initial theory.

Now we are ready to formulate the results on component properties of progression in terms of decomposability and inseparability. We start with negative examples in which every $\mathcal{BAT}$ is local-effect and the initial theories are formulated in the language of situation calculus, i.e. in first-order logic. All free variables in axioms of $\mathcal{BAT}$s are assumed to be universally quantified. As the progression $\mathcal{D}_{S_\alpha}$ is a set of formulas uniform in some situation term $S_\alpha$ which may occur in every formula of $\mathcal{D}_{S_\alpha}$ (thus potentially spoiling decomposability), we consider the mentioned component properties for the theory $\mathcal{D}_{S_\alpha}(S_\alpha/S_0)$ instead of $\mathcal{D}_{S_\alpha}$. Otherwise, in every result we would have to speak of $\Delta \cup \mathtt{sig}(S_\alpha)$–decomposability of progression instead of just $\Delta$–decomposability.

Consider a $\mathcal{BAT}$ $\mathcal{D}$ with $\Delta$–decomposable initial theory $\mathcal{D}_{S_0}$ for some signature $\Delta$. The general definition of a successor state axiom gives enough freedom to design examples showing loss or gain of the decomposability property of $\mathcal{D}_{S_0}$ or inseparability of its components. As SSA may contain symbols that are even not present in $\mathtt{sig}(\mathcal{D}_{S_0})$, or symbols from both components of $\mathcal{D}_{S_0}$ (if decomposition exists), this should not be a surprise for the reader. Therefore, it makes sense to restrict our study to those $\mathcal{BAT}$s, where SSAs have one of the well-studied forms, for instance, to local-effect theories. It turns out that this form is still powerful enough to easily formulate negative results when the mentioned properties are not preserved.

First, we provide a trivial example showing that the decomposability property of the initial theory can be easily lost under progression. The example is given rather as a simple illustration of progression for readers new to this notion. Next, we show that $\Delta$–inseparability of components of the initial theory $\mathcal{D}_{S_0}$ can be easily lost when fluents are present in $\Delta$ (Example 5). The third observation is that even if there are no fluents in $\Delta$, some components of $\mathcal{D}_{S_0}$ can split after progression into theories which are no longer inseparable (Example 6). All observations hold already for local-effect $\mathcal{BAT}$s and follow from the simple fact that after progression some new information

from SSAs can be added to the initial theory which spoils its component properties. We only need to provide a combination of an initial theory with a set of SSAs appropriate for this purpose. The aim of Theorem 1 following these negative examples is to prove that if $\Delta$ does not contain fluents and the components of $\mathcal{D}_{S_0}$ do not split after progression, then $\Delta$–inseparability is preserved after progression under a slight stipulation which is caused only by generality of the theorem and non-uniqueness of progression in the general case. This stipulation is avoided in Theorem 2, where we consider the class of local-effect $\mathcal{BAT}$s.

**Example 4** Decomposability lost under progression. Consider a basic action theory $\mathcal{D}$ with $\{F, A, c_1, c_2\} \subseteq \mathtt{sig}(\mathcal{D})$, where $F$ is a ternary fluent, $A$ is a unary action function, and $c_1, c_2$ are object constants. Let the theory $\mathcal{D}_{ss}$ consist of the single axiom

$F(x, y, do(a, s)) \leftrightarrow a = A(x, y) \vee F(x, y, s)$

and let the initial theory $\mathcal{D}_{S_0}$ consist of two formulas $Taut(c_1)$ and $Taut(c_2)$ uniform in $S_0$, which are tautological sentences in signatures $\{c_1\}$ and $\{c_2\}$, respectively. Clearly, $\mathcal{D}_{S_0}$ is $\varnothing$–decomposable theory.

On the other hand, the progression $\mathcal{D}_{S_\alpha}$ of $\mathcal{D}_{S_0}$ wrt the action $\alpha = A(c_1, c_2)$ is equivalent to the theory consisting of the ground atom

$F(c_1, c_2, do(\alpha, S_0))$.

This can be verified following Definition 8 directly, or by Proposition 8, since $\mathcal{D}$ is local-effect. Anyway, it is easy to check that $\mathcal{D}_{S_\alpha}(S_\alpha/S_0)$ (and $\mathcal{D}_{S_\alpha}$, as well) is not a $\Delta$–decomposable theory (for any $\Delta$).

For a signature $\Delta$, with $S_0 \in \Delta$, and a unary action $A(c)$, we now give an example of a local-effect basic action theory $\mathcal{D}$ with $\mathcal{D}_{S_0}$, an initial theory $\Delta$–decomposable into finite $\Delta$–inseparable components, such that progression $\mathcal{D}_{S_\alpha}(S_\alpha/S_0)$ of $\mathcal{D}_{S_0}$ wrt $A(c)$ (with term $S_\alpha$ substituted with $S_0$) is finitely axiomatizable and $\Delta$–decomposable, but the decomposition components are no longer $\Delta$–inseparable, unless we allow them to be infinite.

**Example 5** $\Delta$–inseparability is lost when fluents are in $\Delta$. Consider a basic action theory $\mathcal{D}$ with $\{F, P, Q, R, A, b, c, d\} \subseteq \mathtt{sig}(\mathcal{D})$, where $F$ is a binary fluent, $P, Q$ are unary predicates, $R$ is a binary predicate, $A$ is a unary action function, and $b, c, d$ are object constants. Let $\Delta = \{F, R, S_0\}$ and define subtheories of $\mathcal{D}$ as follows:

- $\mathcal{D}_{ss} = \{F(x, do(a, s)) \equiv (a = A(x)) \wedge P(x) \wedge Q(d) \ \vee \ F(x, s)\}$;
- $\mathcal{D}_{S_0} = \mathcal{D}_1 \cup \mathcal{D}_2$, where
  - $\mathcal{D}_1$ consists of a tautological formula in the signature $\{F, R, b, S_0\}$ which is uniform in $S_0$,

– $\mathcal{D}_2 = \{P(x) \to \exists y(R(x,y) \wedge P(y)),\ \neg F(x, S_0)\}$.

By the syntactic form, $\mathcal{D}_{S_0}$ is $\Delta$–decomposable: we have $\mathcal{D}_{S_0} = \mathcal{D}_1 \cup \mathcal{D}_2$, $\mathtt{sig}(\mathcal{D}_1) \cap \mathtt{sig}(\mathcal{D}_2) = \Delta$, $\mathtt{sig}(\mathcal{D}_1) \setminus \Delta = \{b\}$, and $\mathtt{sig}(\mathcal{D}_2) \setminus \Delta = \{P\}$.

Note that $\mathcal{D}_{ss} \models F(x, do(A(c), S_0)) \equiv (x = c) \wedge P(c) \wedge Q(d) \vee F(x, S_0)$, the result of substitution of the ground action $\alpha = A(c)$ and the situation constant $S_0$ into SSA. As $\mathcal{D}_2 \models \neg F(x, S_0)$, we have $\mathcal{D}_{ss} \cup \mathcal{D}_{S_0} \models F(c, do(A(c), S_0)) \equiv P(c) \wedge Q(d)$; denote the last formula by $\varphi$.

By Proposition 8 it is easy to verify that the union of sets $\mathcal{D}_1$ and $\mathcal{D}_2' = (\mathcal{D}_2 \cup \{\varphi\}) \setminus \{\neg F(x, S_0)\}$, with every occurrence of $S_0$ in $\mathcal{D}_1$ substituted with $S_\alpha = do(A(c), S_0)$, is a progression ($\mathcal{D}_{S_\alpha}$) of $\mathcal{D}_{S_0}$ wrt $A(c)$.

By the syntactic form, $\mathcal{D}_{S_\alpha}(S_\alpha/S_0)$ $\Delta$–decomposable theory. On the other hand, we have $\varphi \models F(c, do(A(c), S_0)) \to P(c)$ and thus, $\mathcal{D}_2'(S_\alpha/S_0) \models \{F(c, S_0) \to \exists y R(c, y),\quad F(c, S_0) \to [\exists y \exists z R(c, y) \wedge R(y, z)], \ldots\}$. It follows from Fact 2 that this theory is not finitely axiomatizable by formulas of first order logic in the signature $\Delta$ and it is not hard to verify that the obtained theory $\mathcal{D}_{S_\alpha}(S_\alpha/S_0)$ can not have a decomposition into $\Delta$–inseparable components.

We note that in the example above, the initial theory $\mathcal{D}_{S_0}$ is in fact $\varnothing$–decomposable with one signature component equal to $\{b\}$ and the other component containing the rest of the symbols. It is easy to see that progression of $\mathcal{D}_{S_0}$ wrt $A(c)$ is $\varnothing$–decomposable as well. We use tautologies in the example just to illustrate the idea that information from SSA can propagate to the initial theory after progression, thus making the components lose the inseparability property. There is a plenty of freedom to formulate similar examples with the help of non-tautological formulas which syntactically "bind" symbols $F, R, b, S_0$ in the theory $\mathcal{D}_1$. We appeal to a similar observation in Example 6.

**Example 6** Split of a component and loss $\Delta$–inseparability. Consider $\mathcal{BAT}$ $\mathcal{D}$ with $\{F_1, F_2, D, B, R, A, c\} \subseteq \mathtt{sig}(\mathcal{D})$, where $F_1, F_2$ are binary fluents, $D, B$ are unary predicates, $R$ is a binary predicate, $A$ is a unary action function, and $c$ is an object constant. Let $\Delta = \{D, R, S_0\}$ and define the subtheories of $\mathcal{D}$ as follows:

- $\mathcal{D}_{ss} = \{F_1(x, do(a, s)) \equiv F_1(x, s) \wedge \neg(a = A(x)),\quad F_2(x, do(a, s)) \equiv F_2(x, s)\}$
- $\mathcal{D}_{S_0} = \mathcal{D}_1 \cup \mathcal{D}_2$, where $\mathcal{D}_1$ is the set of formulas:
  - $D(x) \vee R(x, y) \to F_1(c, S_0)$
  - $D(x) \to P(x)$
  - $P(x) \to \exists y(R(x, y) \wedge P(y))$

and $\mathcal{D}_2$ consists of the following three:

- $D(x) \rightarrow B(x)$
- $B(x) \rightarrow \exists y(R(x,y) \wedge B(y))$
- $Taut(F_2, S_0)$, a tautology in the signature $\{F_2, S_0\}$ which is uniform in $S_0$.

By definition, $\mathcal{D}_{S_0}$ is $\Delta$–decomposable into $\Delta$–inseparable components $\mathcal{D}_1$ and $\mathcal{D}_2$. Note that $\mathcal{D}_{ss} \models \neg F_1(c, do(A(c), S_0))$, the result of substitution of the ground action $A(c)$, situation constant $S_0$, and object constant $c$ in SSA.

Consider progression of $\mathcal{D}_{S_0}$ wrt the action $\alpha = A(c)$. By Proposition 8, it is equivalent to the theory $\mathcal{D}_{S_\alpha} = \mathcal{D}'_1 \cup \mathcal{D}''_1 \cup \mathcal{D}'_2$, where $\mathcal{D}'_1$ is the set of the following formulas:

- $\neg F_1(c, do(A(c), S_0))$
- $Taut(D, R)$, a tautological formula in the signature $\{D, R\}$ which is uniform in $S_\alpha$

$\mathcal{D}''_1$ is the set of formulas:

- $D(x) \rightarrow P(x)$
- $P(x) \rightarrow \exists y(R(x,y) \wedge P(y))$
- $Taut(F_2, S_\alpha)$, a tautological formula in the signature $\{F_2, do, A, c, S_0\}$ which is uniform in $S_\alpha$

and $\mathcal{D}'_2$ is the theory $\mathcal{D}_2$ with every occurrence of $S_0$ substituted with $S_\alpha$.

Clearly, $\mathcal{D}_{S_\alpha}(S_\alpha/S_0)$ is $\Delta$–decomposable, but the components $\mathcal{D}'_1(S_\alpha/S_0)$ and $\mathcal{D}''_1(S_\alpha/S_0)$ are not $\Delta$–inseparable (similarly, $\mathcal{D}'_1(S_\alpha/S_0)$ and $\mathcal{D}'_2(S_\alpha/S_0)$). Moreover, by Fact 2, they can not be made $\Delta$–inseparable while remaining finitely axiomatizable.

To formulate the theorems below, we let $\mathcal{D}$ denote a $\mathcal{BAT}$ with an initial theory $\mathcal{D}_{S_0}$, a set of successor state axioms $\mathcal{D}_{ss}$, and a unique name assumption theory $\mathcal{D}_{una}$.

**Definition 9** Fluent–free signature. *A signature $\Delta$ is called fluent–free if no fluent (from the alphabet of situation calculus) is contained in $\Delta$.*

Theorem 1 is provided as a general theoretical result on preservation of inseparability of components of the initial theory after progression. As we have already seen in Example 5, the initial theory and progression may differ in consequences involving symbols of fluents. Thus in general, preservation

of $\Delta$–inseparability can be guaranteed only for fluent-free signatures $\Delta$. Besides, by the model–theoretic Definition 8, progression is not uniquely defined. There is no restriction on occurrences of the unique–name–assumption formulas in progression which may easily lead to loss of inseparability of the components. In other words, progression may logically imply unique–name–assumption formulas even if the initial theory did not imply them. Some decomposition components of progression may imply such formulas, while the others may not. For this reason, we have to speak of inseparability "modulo" theory $\mathcal{D}_{una}$ in the theorem below. In particular, we have to make the assumption that not only the components $\{D_i\}_{i\in I\subseteq\omega}$ of the initial theory are pairwise $\Delta$–inseparable, but so are the theories $\{\mathcal{D}_{una}\cup D_i\}_{i\in I}$. In the theorem, we do not specify how the progression was obtained (cf. Theorem 2) and the only condition that relates the components of progression with those of the initial theory says about containment of $\Delta$–consequences. Thereby, we formulate the idea that components of progression do not split $\Delta$–consequences of the components of the initial theory (cf. Example 6).

**Theorem 1** Preservation of $\Delta$-insep. for fluent-free $\Delta$. *Let $\mathcal{L}$ have PIP and $\mathcal{D}$ be $\mathcal{BAT}$ in which $\mathcal{D}_{S_0}$ and $\mathcal{D}_{una}$ are theories in $\mathcal{L}$. Let $\sigma \subseteq \mathtt{sig}\,(\mathcal{D}_{S_0})$ be a fluent–free signature and denote $\Delta = \mathtt{sig}\,(\mathcal{D}_{una})\cup\sigma$. Suppose the following:*

- *$\mathcal{D}_{S_0}$ is $\sigma$–decomposable with some components $\{D_i\}_{i\in I\subseteq\omega}$ such that the theories from $\{\mathcal{D}_{una}\cup D_i\}_{i\in I}$ are pairwise $\Delta$–inseparable;*
- *$\mathcal{D}_{S_\alpha}(S_\alpha/S_0)$ is equivalent to the union of theories $\{D'_j\}_{j\in J\subseteq\omega}$ such that for every $j\in J$ and some $i\in I$, $\mathtt{Cons}\,(\mathcal{D}_{una}\cup D'_j,\Delta)\supseteq\mathtt{Cons}\,(\mathcal{D}_{una}\cup D_i,\Delta)$.*

*Then the theories from $\{\mathcal{D}_{una}\cup D'_j\}_{j\in J\subseteq\omega}$ are pairwise $\Delta$–inseparable.*

The next theorem provides a result on local-effect $\mathcal{BAT}s$ with initial theories in first-order logic for which progression becomes more concrete, since it can be computed by syntactic manipulations. In contrast to Theorem 1, this allows us to judge about inseparability without the theory $\mathcal{D}_{una}$ in background. Essentially, the conditions of the theorem are defined to guarantee componentwise computation of progression for a decomposable initial theory. For the reader's convenience, we stress that in the formulation of the theorem, the indices $i$ and $j$ vary over components of $\mathcal{D}_{ss}$ and $\mathcal{D}_{S_0}$, respectively. The signatures $\Delta_1$ and $\Delta_2$ are the sets of allowed common symbols between the components of $\mathcal{D}_{ss}$ and $\mathcal{D}_{S_0}$, respectively. We recall that $\mathcal{F}$ denotes the set of fluents from the alphabet of the language of situation calculus.

**Theorem 2** Preservation of components in local-effect $\mathcal{BAT}s$. *Let $\mathcal{D}$ be a local-effect $\mathcal{BAT}$ with $\mathcal{D}_{S_0}$, an initial theory in first-order logic. Let $\Delta_1$, $\Delta_2$*

*be fluent-free signatures and* $\alpha = A(c_1, \ldots, c_k)$, $k \in \omega$, *be a ground action term. Denote* $\Delta = \Delta_1 \cup \Delta_2 \cup \{c_1, \ldots, c_k\}$ *and suppose the following:*

- $\mathtt{sig}(\mathcal{D}_{ss}) \cap \mathcal{F} \subseteq \mathtt{sig}(\mathcal{D}_{S_0})$*;*
- $\mathcal{D}_{ss}$ *is the union of theories* $\{D_i\}_{i \in I \subseteq \omega}$*, with* $\mathtt{sig}(D_n) \cap \mathtt{sig}(D_m) \subseteq \Delta_1$ *for all* $n, m \in I$, $n \neq m$*;*
- $\mathcal{D}_{S_0}$ *is* $\Delta_2$*–decomposable with components* $\{D'_j\}_{j \in J \subseteq \omega}$ *uniform in* $S_0$*;*
- *for every* $i \in I$, *there is* $j \in J$ *such that* $\mathtt{sig}(D_i) \cap \mathtt{sig}(\mathcal{D}_{S_0}) \subseteq \mathtt{sig}(D'_j)$.

*Then* $\mathcal{D}_{S_\alpha}(S_\alpha/S_0)$ *is* $\Delta$*–decomposable. If the components* $\{D'_j\}_{j \in J}$ *are pairwise* $\Delta$*–inseparable, then so are the components of* $\mathcal{D}_{S_\alpha}(S_\alpha/S_0)$ *(in the corresponding decomposition).*

We note that a result similar to Theorem 2 can be proved in the general case, for progression of not necessarily local-effect $\mathcal{BAT}s$, by considering progression as a set of consequences of $\mathcal{D}_{una} \cup \mathcal{D}_{ss} \cup \mathcal{D}_{S_0}$ uniform in $S_\alpha$.

## 5. Conclusion

We have considered the influence of the theory update operations, such as forgetting and progression, on preserving the component properties of theories, such as decomposability and inseparability. The results of the paper are in a certain sense expected. Forgetting and progression have semantic nature, since the input and the output of these transformations are related to each other by using restrictions on the classes of models. On the contrary, the decomposability and inseparability properties are defined using entailment in a logic. Therefore, they have rather a syntactic origin, because logics (weaker than second-order) may not distinguish the needed classes of models. As a consequence, the conceptual "distance" between these two kinds of notions is potentially immense. It can be somewhat bridged by the choice of either an appropriate logic, or appropriate theories in the input. We have identified conditions that should be imposed on the components of input theories to match these notions more closely. Also, the Parallel Interpolation Property (PIP) turned out to be a relevant property of logics in our investigations. The results can be briefly summarized in the tables below. For brevity, we use $\sigma$ to denote a signature or a ground atom. We slightly abuse notation and consider $\sigma$ as a set of symbols even in the case of a ground atom implying that in the latter case $\sigma$ consists of the single predicate symbol from the atom. We assume that the input of operations of forgetting and progression is a union of theories $\mathcal{T}_1$ and $\mathcal{T}_2$ with $\mathtt{sig}(\mathcal{T}_1) \cap \mathtt{sig}(\mathcal{T}_2) = \Delta$, for a signature $\Delta$.

| Property | Condition | Result | Reference |
|---|---|---|---|
| Preservation of $\Delta$–inseparability of $\mathcal{T}_1$ and $\mathcal{T}_2$ under forgetting $\sigma$ | $\sigma \cap \Delta = \varnothing$ | YES | Corollary 2 |
| | $\sigma \subseteq \Delta$ and $\sigma$ is a ground atom | NO | Example 2 |
| | $\sigma \subseteq \Delta$ and $\sigma$ is a signature | YES, if logic has PIP | Proposition 5 |
| | $\sigma \subseteq \Delta$ and $\mathcal{T}_1$, $\mathcal{T}_2$ are semantically inseparable | YES | Proposition 6 |
| Distributivity of forgetting $\sigma$ over union of $\mathcal{T}_1$ and $\mathcal{T}_2$ | $\sigma \cap \Delta = \varnothing$ | YES | Corollary 2 |
| | $\sigma \subseteq \Delta$ | NO, even if $\mathcal{T}_1$ and $\mathcal{T}_2$ are semantically inseparable | Example 3 |
| | $\mathcal{T}_1$ and $\mathcal{T}_2$ are semantically inseparable "modulo $\sigma$" | YES | Proposition 7 |

| Property | Condition | Preservation | Reference |
|---|---|---|---|
| $\Delta$–inseparability of components of initial theory under progression | at least one fluent is present in $\Delta$ | NO | Example 5 |
| | $\Delta$ is fluent-free and some components of initial theory split under progression | NO | Example 6 |
| | $\Delta$ is fluent-free and components of initial theory do not split under progression | YES, modulo the unique name assumption theory | Theorem 1 |
| $\Delta$–decomposability and preservation of signature components of an initial theory under progression wrt an action term $\alpha$ | Unconditionally, in particular for local-effect $\mathcal{BAT}$s | NO | Example 4 |
| | $\mathcal{BAT}$ is local–effect, $\Delta$ is fluent-free, and components of $\mathcal{D}_{S_0}$ are aligned with components of $\mathcal{D}_{ss}$ | YES, modulo constants in term $\alpha$ | Theorem 2 |

## References

[1] Cordell Green C. Application of theorem proving to problem solving // Proc. IJCAI. – 1969. – P. 219–240.

[2] Gu Y., Soutchanski M. A description logic based situation calculus // Ann. Math. Artif. Intell. – 2010. – Vol. 58: 1–2. P. 3–83.

[3] Konev B., Lutz C., Ponomaryov D., Wolter F. Decomposing description logic ontologies // Proc. Twelfth International Conference on the Principles of Knowledge Representation and Reasoning, 2010.

[4] Konev B., Lutz C., Walther D., Wolter F. Formal properties of modularization // Modular Ontologies / H. Stuckenschmidt C. Parent, S. Spaccapietra (eds.). – Lect. Notes Comput. Sci. – Springer Verlag, 2009. – Vol. 5445. – P. 25–66.

[5] Kourousias G., Makinson D. Parallel interpolation, splitting, and relevance in belief change // J. of Symbolic Logic. – Vol. 72:3. – P. 994–1002.

[6] Levesque H., Lakemeyer G. Cognitive Robotics // Handbook of Knowledge Representation / Frank van Harmelen, V. Lifschitz, and B. Porter (eds.). – Elsevier, 2007.

[7] Lin F., Reiter R. Forget it! // Proc. of the AAAI Fall Symposium on Relevance. – 1994. – P. 154–159.

[8] Lin F. On strongest necessary and weakest sufficient conditions // Artif. Intell. – 2001. – Vol. 128:1-2. P. 143–159.

[9] Liu Y., Lakemeyer G. On first-order definability and computability of progression for local-effect actions and beyond // Proc. 21st International Jont Conference on Artifical Intelligence. – Morgan Kaufmann Publishers Inc., 2009. – P 860–866.

[10] Lutz C., Walther D., Wolter F. Conservative extensions in expressive description logics // Proc. of the 20th Int. Joint Conf. on Artificial Intelligence IJCAI-07. – AAAI Press, 2007. – P. 453–458.

[11] Lutz C., Wolter F. Mathematical logic for life science ontologies // Proc. of the 16th International Workshop on Logic, Language, Information and Computation. – Lect. Notes Comput. Sci. –Springer Verlag, 2009. – Vol. 5514. – P. 37–47.

[12] Lutz C., Wolter F. Deciding inseparability and conservative extensions in the description logic $\mathcal{EL}$ // J. of Symbolic Computation. – 2010. – Vol. 45:2. – P. 194–228.

[13] McCarthy J. Situations, Actions and Causal Laws. – Memo 2, Stanford University, Department of Computer Science, 1963. – Reprinted in"Semantic Information Processing"/ M. Minsky, (ed.). – The MIT Press, Cambridge (MA), 1968. – P. 410–417.

[14] McCarthy J., Hayes P. Some philosophical problems from the standpoint of artificial intelligence // Machine Intelligence / B. Meltzer and D. Michie (eds.). – Edinburgh University Press, 1969. – Vol. 4. – P. 463–502.

[15] Pirri F., Reiter R. Some contributions to the metatheory of the situation calculus // J. of the ACM. – 1999. – Vol. 46:3. – P. 325–361.

[16] Ponomaryov D. On decomposibility in logical calculi // Bull. Novosibirsk Comp. Center. Ser. Computer Science. – Novosibirsk, 2008. – IIS Special Iss. 28. – P. 111–120.

[17] Reiter R. Knowledge in Action: Logical Foundations for Describing and Implementing Dynamical Systems. – The MIT Press, 2001.

[18] Vassos S., Levesque H. On the progression of situation calculus basic action theories: resolving a 10-year-old conjecture // Proc. of the 23rd National Conference on Artificial intelligence (AAAI'08). – AAAI Press, 2008. – Vol. 2. – P. 1004–1009.

[19] Yehia W., Lippmann M., Liu H., Baader F., Soutchanski M. Experimental results on solving the projection problem in action formalisms based on description logics // Proc. of the 25th Intern. Workshop on Description Logics, 2012.