Progression of Decomposed Situation Calculus Theories

Denis Ponomaryov[1] and Mikhail Soutchanski[2]

[1] Institute of Artificial Intelligence, University of Ulm,
James-Franck-Ring, Geb. O27, 89069, Germany;
A.P. Ershov Institute Of Informatics Systems,
Russian Academy of Sciences, Siberian Division,
6, Acad. Lavrentjev pr., Novosibirsk, 630090, Russia
ponom@iis.nsk.su
[2] Department of Computer Science, Ryerson University,
245 Church Street, ENG281, Toronto, ON, M5B 2K3, Canada
mes@scs.ryerson.ca

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. Progression provides a form of forward reasoning; it relies on forgetting values of those features which are subject to change and computing new values for them. We consider decomposability and inseparability, two component properties of theories known from the literature, and 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 forgetting and progression of initial theories in local–effect basic action theories of the situation calculus.

Keywords: decomposition, inseparability, forgetting, progression, reasoning about actions

1 Introduction

Modularity of theories has been established as an important research topic in knowledge representation. It includes both theoretical and practical aspects of modularity of theories formulated in different logical languages $\mathcal{L}$ ranging from weak (but practical) description logic (DL) $\mathcal{EL}$ to more expressive logics [10, 9, 24, 7], to cite a few. Surprisingly, this research topic is little explored in the context of reasoning about actions. More specifically, it is natural to decompose a large heterogeneous theory covering several loosely coupled application domains into components that have little or no intersection in terms of signatures. Potentially, such decomposition can facilitate solving the projection problem that requires answering whether a given logical formula is true after executing a sequence of actions (events). In cases, when a query is a logical formula composed from symbols occuring only in one of the components, the query can be answered more easily than in the case when the whole theory is required. In turn, this can help in solving other reasoning problems such as planning or high-level program execution that require solution to the projection problem as a prerequisite. To the best of our knowledge, the only previous work that explored decomposition of logical theories for the purposes of solving the projection problem are the papers [1, 2]. These papers investigate decomposition in the situation calculus [22, 25], a well-known logical formalism for representation of actions and their effects. The author proposed reasoning procedures for a situation calculus theory by dividing syntactically the whole theory into weakly related partitions. Specifically, he developed algorithms that use local computation inside syntactically identified partitions and message passing between partitions. We take a different approach in our paper. Instead of decomposing the whole action theory into subsets, as in [1, 2], we consider signature decompositions of an initial theory only. Our components are not necessarily syntactic subsets of the initial theory. We concentrate on foundations, and explore properties of components produced by our decomposition. Whenever possible, we try to formulate these properties in a general logical language $\mathcal{L}$ that is a fragment of second order logic, but when necessary, we talk about a specific logic.

This paper considers the decomposability and inseparability properties of logical theories. These properties are well-known in research on modularization in the area of knowledge representation [9, 24, 10, 19]. 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 sharing a strictly defined set Δ of signature symbols. Inseparability of theories wrt some signature Δ means that the theories have the same set of logical consequences in the signature Δ. If a theory $\mathcal{T}$ is Δ–decomposable into Δ–inseparable components, then (under certain restrictions on the underlying logic) each component of the decomposition contains all information from $\mathcal{T}$ in its own signature. 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, an initial 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 caused by actions, while some other knowledge should be forgotten as no longer true in the next situation. We consider two types of update operators: 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 [14]. Forgetting a signature σ 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 [25] is a knowledge representation logical formalism, 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 an initial theory), axiomatize preconditions telling when actions can be performed, and add also the axioms about the effects of actions on situation-dependent properties. Then, one can reason about consequences of sequences of actions to check whether properties of interest hold in a given situation resulting from executing a sequence of actions and whether a certain sequence of actions is executable. In the situation calculus, the so-called basic action theories represent such axiomatizations [25]. As mentioned above, each basic action theory contains an initial theory which represents incomplete knowledge about an initial situation S_0. In a special case, when there is complete knowledge about a finite number of individuals having unique names, the initial theory can be implemented as a relational database [25]. Roughly, a basic action theory $\mathcal{D}$ is a union of an initial theory $\mathcal{D}_{S_0}$ with some theory $\mathcal{T}$, defining transitions among situations, and a set of "canonical" axioms assumed to be true for all application problems represented in the situation calculus. Informally speaking, an update of the initial theory after execution of an action is called progression of the initial theory wrt an action. More precisely, progression of $\mathcal{D}_{S_0}$ wrt some action α is a logical consequence of $\mathcal{D}$ which contains all information from $\mathcal{D}$ about the situation resulting from execution of α in the situation S_0. Ideally, it is computed as updating $\mathcal{D}_{S_0}$ with some logical consequences of $\mathcal{T}$, once all information in $\mathcal{D}_{S_0}$ which is no longer true in the resulting situation has been forgotten. 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 the area of artificial intelligence (AI). Having been developed in the 1960s by John McCarthy and his colleagues [20, 22, 6], 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 is worth mentioning that there are both conceptual and technical differences between the situation calculus, 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 [12]. In our paper, when we talk about the situation calculus, we follow the axiomatic approach and notation by R.Reiter [25] 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.

In this paper, we are interested in the case when the initial theory is decomposed into inseparable components and study which restrictions guarantee preservation of decomposability and inseparability of components after progression. 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. This invariance is important since progression may continue indefinitely as long as new actions are being executed. If it is the case, then it would suffice to compute a decomposition of the initial theory once, and this decomposition would remain "stable" after progression wrt any arbitrary sequence of actions. Moreover, if an executed action has effects only on one component 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 of when the decomposability and inseparability properties are preserved under progression and under forgetting. This paper contributes to general understanding of forgetting and progression, since new results about them are needed for the purposes of our investigation. Not surprisingly, both forgetting and progression have intricate interactions with properties of decomposed components. In general, it is very hard to guarantee preservation of decomposability and inseparability, 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 cases when these properties are preserved. It turns out that some of these cases have a practically important formulation. Another contribution of the paper is in formulating clear negative examples demonstrating the cases when decomposability and inseparability are lost under progression. In particular, our examples demonstrate there is little hope to preserve inseparability if the different components share a fluent. Decomposability turns out to be also a fragile property that can be easily lost after executing just one action in a simple basic action theory. Overall, this paper contributes by not only studying forgetting and progression, but also by completing a thorough and comprenehsive study when decomposability and inseparability are preserved and when they are lost.

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.

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 Δ is a signature then we say that $\mathcal{M}_1$ and $\mathcal{M}_2$ agree on Δ if they have the same domains for each sort and the same interpretation of every symbol from Δ. If $\mathcal{M}$ is a structure and σ 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 σ, i.e. the structure with predicate and function names from σ, where every symbol of σ 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 φ 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 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 the semantic equivalence. If $\mathcal{T}$ is a set of formulas in $\mathcal{L}$ and Δ 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 Δ, i.e. $\mathtt{Cons}\,(\mathcal{T},\Delta) = \{\varphi \in \mathcal{L} \mid \mathcal{T} \models \varphi \text{ and } \mathtt{sig}\,(\varphi) \subseteq \Delta\}$. 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 Δ a signature, then some models of* $\mathtt{Cons}\,(\mathcal{T},\Delta)$ *may not have an 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 Δ a signature, then* $\texttt{Cons}\,(\mathcal{T},\Delta)$ *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 $\texttt{Cons}\,(\mathcal{T},\Delta)$ *is the following infinite set of formulas*
$\forall x\ A(x) \to \exists y R(x,y)$,
$\forall x\ A(x) \to [\exists y \exists u R(x,y) \wedge R(y,u)]$,
$\forall x\ A(x) \to [\exists y \exists u \exists v R(x,y) \wedge R(y,u) \wedge R(u,v)]$,
$\ldots$
By compactness, this theory is not finitely axiomatizable in first-order logic.

There is 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 [18]).

First, we formulate two component properties of theories considered in this paper. The notion of inseparability has been previously introduced in the context of entailment in Description Logics, e.g., see [10, 19].

Definition 1 (Δ–inseparability) *Theories $\mathcal{T}_1$ and $\mathcal{T}_2$ in $\mathcal{L}$ are called Δ–inseparable for a signature Δ, if* $\texttt{Cons}\,(\mathcal{T}_1,\Delta) = \texttt{Cons}\,(\mathcal{T}_2,\Delta)$. *That is, no formula in the signature Δ "witnesses" any distinction between $\mathcal{T}_1$ and $\mathcal{T}_2$.*

In other words, $\mathcal{T}_1$ and $\mathcal{T}_2$ are Δ–inseparable, if for any formula ψ in signature Δ, $\mathcal{T}_1$ entails ψ iff $\mathcal{T}_2$ does. The following notion is introduced in [24], and is applied to the study of modularity in [9].

Definition 2 (Δ–decomposability property) *Let $\mathcal{T}$ be a theory in $\mathcal{L}$ and $\Delta \subseteq \texttt{sig}\,(\mathcal{T})$ be a subsignature. We call $\mathcal{T}$ Δ–decomposable, if there are theories $\mathcal{T}_1$ and $\mathcal{T}_2$ in $\mathcal{L}$ such that*

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

The pair $\langle \mathcal{T}_1, \mathcal{T}_2 \rangle$ is called Δ–decomposition of $\mathcal{T}$ and the theories $\mathcal{T}_1$ and $\mathcal{T}_2$ are called Δ–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 Δ is clear from the context. The sets $\texttt{sig}\,(\mathcal{T}_1) \setminus \Delta$ *and* $\texttt{sig}\,(\mathcal{T}_2) \setminus \Delta$ *are called signature (Δ–decomposition) components of $\mathcal{T}$.*

The notion of Δ–decomposition is defined 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 Δ–decomposable theory in $\mathcal{L}$ for a signature Δ, then there is a Δ–decomposition $\langle \mathcal{T}_1, \mathcal{T}_2 \rangle$ of $\mathcal{T}$, where $\mathcal{T}_1$ and $\mathcal{T}_2$ are finite. Although, the union $\mathcal{T}_1 \cup \mathcal{T}_2$ must entail all consequences of $\mathcal{T}$ in the signature Δ, the components $\mathcal{T}_1$ and $\mathcal{T}_2$ may not be Δ–inseparable, if we demand them to be finite. For example, the set of Δ–consequences of $\mathcal{T}_2$ may not be finitely axiomatizable in $\mathcal{L}$ by axioms of $\mathcal{T}_1$. This easily follows from Fact 2 which notes that this effect is already possible in weak languages such as the sub-boolean description logic $\mathcal{EL}$. On the other hand, Δ–inseparability of decompositions can always be obtained if the underlying logic $\mathcal{L}$ has uniform interpolation (cf. Proposition 2 in [24]). Both Δ–decomposition and Δ–inseparability are required to achieve modularity. Without Δ–inseparability components are not self-sufficient, since a component may not entail some of the consequences in the shared vocabulary Δ. The ideal case is when a theory $\mathcal{T}$ has Δ-decomposition into finite Δ–inseparable components, as noted in the following.

The well-known property of logics related to signature decompositions of theories is the Parallel Interpolation Property (PIP) first considered in a special form in [11] and studied later in a more general form in [9]. Note that PIP is closely related to Craig's interpolation [4, 5].

Definition 3 (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 φ 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 $\mathcal{T}_1' \cup \mathcal{T}_2' \models \varphi$;*
- *$\mathtt{sig}(\mathcal{T}_i') \setminus \Delta \subseteq (\mathtt{sig}(\mathcal{T}_i) \cap \mathtt{sig}(\varphi)) \setminus \Delta$.*

In fact, PIP can be understood as an iterated version of Craig's interpolation in the logics that have compactness and deduction theorem (see Lemma 1 in [24]). Many logics known to have Craig interpolation, e.g. second and first-order logics, numerous modal logics and some description logics, also have PIP. It is easy to note that, in presence of PIP, decomposing a set $\mathcal{T}$ of formulas into inseparable components wrt a signature Δ gives a family of theories that imply all the consequences of $\mathcal{T}$ in their own subsignatures.

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

Proof. Assume $\mathtt{sig}(\varphi) \subseteq \mathtt{sig}(\mathcal{T}_1)$. If $\mathcal{T}_1 \models \varphi$ then $\mathcal{T} \models \varphi$ by definition of Δ–decomposability. If $\mathcal{T} \models \varphi$ then $\mathcal{T}_1 \cup \mathcal{T}_2 \models \varphi$ and by PIP, there are $\mathcal{T}_1'$ and $\mathcal{T}_2'$ such that $\mathcal{T}_1 \models \mathcal{T}_1'$, $\mathcal{T}_2 \models \mathcal{T}_2'$, $\mathcal{T}_1' \cup \mathcal{T}_2' \models \varphi$, and $\mathtt{sig}(\mathcal{T}_2') \subseteq \Delta$. As $\mathcal{T}_1$ and $\mathcal{T}_2$ are Δ–inseparable, we obtain $\mathcal{T}_1 \models \mathcal{T}_2'$ and conclude that $\mathcal{T}_1 \models \varphi$. $\square$

In other words, in presence of PIP, inseparable decomposition components can be used instead of the original theory for checking entailment of formulas in the corresponding subsignatures. This is the reason for our interest in the inseparability property in connection with decompositions. As shown in [1, 2], having a decomposed theory can be beneficial even without inseparability by applying the known methods of distributed reasoning via message passing between components. However, having inseparability of components allows the reasoner to avoid message passing completely.

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 Σ 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 Σ 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 σ, a theory D_{una} contains axioms of the form:

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

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 φ in 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 φ: $\{Poss, \preceq\} \cap \mathtt{sig}(\varphi) = \varnothing$;*
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, $\mathbf{F_1}, \ldots, \mathbf{F_n}$ 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 the 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 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 σ 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 [8, 27].

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

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

where $\mathit{\Pi}_A(\bar{x}, s)$ is a formula uniform in s with free variables among $\bar{x}$ and s. Informally, $\mathit{\Pi}_A(\bar{x}, s)$ 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 γ_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 ϕ^+ 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 ϕ has an occurrence of $\bar{y}$. The formula ϕ^+ 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, γ_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 ϕ^- is a formula uniform in s with variables at most among $\bar{x}$, $\bar{z}$, and s. The formula ϕ^- 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 the fluent F if F is the fluent from the left-hand side of φ.

Following the original consistency requirement on SSAs by Reiter (see Proposition 3.2.6 in [25]), we require that in case an action function A^+ occurs in active position in some disjunct of γ^+, then it must not occur in active position in γ^-.

Analogously, if A^- occurs in active position in γ^-, then it must not be in active position in γ^+. 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 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 [23]) *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, φ 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 ([25]). 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 [8]. 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 the effects of an executed action. Computing progression of a given theory $\mathcal{D}_{S_0}$ requires forgetting 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 φ 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 (first introduced in [16]) 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.

Before we proceed to component properties of forgetting (Section 3) and to progression of initial theories (Section 4), we consider an example that helps to illustrate the notion of $\mathcal{BAT}$ and advantages of decomposition of its initial theory. Our example combines the simplified Blocks World (BW) with a kind of Stacks World. A complete axiomatization of BW modelled as a finite collection of finite chains can be found in [3].

Example 1 (Running example of $\mathcal{BAT}$). The blocks-and-stacks-world consists of a finite set of blocks and a finite set of other entities. Blocks can be located on top of each other, while other entities can be either in a heap of unlimited capacity, or can be organized in stacks. There is an unnamed manipulator that can move a block from one block to another, provided that there is nothing on the top of the blocks. It can also put an entity from the heap upon a stack with a named top element, or move the top element of a stack into the heap. For stacking/unstacking operations we adopt the push/pop terminology and use the unary predicate *Block* to distinguish between blocks and other entities. We use the following action functions and relational fluents to axiomatize the mentioned world as a local-effect $\mathcal{BAT}$ in SC.

Actions

- $move(x, y, z)$: Move block x from block y onto block z, provided both x and z are clear.
- $push(x,y)$: Stack entity x from the heap on top of entity y.
- $pop(x)$: Unstack entity x into the heap, provided x is the top element and is not in the heap.

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 s.
- $Top(x, s)$: Entity x is the top element of a stack in s.
- $Inheap(x, s)$: Entity x is in the heap in situation s.
- $Under(x, y, s)$: Entity y is directly under x in situation s.

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

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

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

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

$$Inheap(x,do(a,s)) \leftrightarrow a{=}pop(x) \vee Inheap(x,s) \wedge \neg\exists y(a{=}push(x,y))$$

$$Top(x,do(a,s)) \leftrightarrow \exists y(\, a{=}push(x,y)\,) \vee \exists y(\, a{=}pop(y) \wedge Under(y,x,s)\,) \vee \\ Top(x,s) \wedge a{\neq}pop(x) \wedge \neg\exists y(a{=}push(y,x))$$

$$Under(x,y,do(a,s)) \leftrightarrow a{=}push(x,y) \vee Under(x,y,s) \wedge a{\neq}pop(x)$$

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

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

$$Poss(push(x,y),s) \leftrightarrow \neg Block(x) \wedge \neg Block(y) \wedge Top(y,s) \wedge Inheap(x,s)$$

$$Poss(pop(x),s) \leftrightarrow \neg Block(x) \wedge Top(x,s)$$

Initial Theory ($\mathcal{D}_{S_0}$) is defined using the set of object constants $\{A, B, C\}$ as the set of axioms:

$$\neg\exists y On(y,x,S_0) \wedge \exists y On(x,y,S_0) \wedge \neg Inheap(x,S_0) \rightarrow Clear(x,S_0)$$
$$\exists y\ On(x,y,S_0) \rightarrow Block(x)$$
$$(Top(x,S_0) \vee Inheap(x,S_0)) \rightarrow \neg Block(x)$$
$$On(A,B,S_0)\wedge Block(B)\wedge Block(C)\wedge Clear(A,S_0)\wedge Clear(C,S_0)$$

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.

Notice that all fluents are syntactically related in $\mathcal{D}_{S_0}$, so purely syntactic techniques fail to decompose $\mathcal{D}_{S_0}$ into components sharing no fluents. $\mathcal{D}_{ss}$ is the union of two theories with the intersection of signatures equal to $\{do\}$. At the same time, the initial theory $\mathcal{D}_{S_0}$ is Δ–decomposable for $\Delta = \{Block, S_0\}$ into two distinct Δ–inseparable components:

$$\neg\exists y\ On(y,x,S_0) \wedge \exists y\ On(x,y,S_0) \rightarrow Clear(x,S_0)$$
$$\exists y\ On(x,y,S_0) \rightarrow Block(x)$$
$$On(A,B,S_0)\wedge Block(B)\wedge Block(C)\wedge Clear(A,S_0)\wedge Clear(C,S_0)$$

and

$$(Top(x,S_0) \vee Inheap(x,S_0)) \rightarrow \neg Block(x)$$
$$\exists x\ Block(x)$$

The Example is continued after Theorem 2 in Section 4, where we will show that progression for $\mathcal{BAT}$s of this kind preserves both decomposability and inseparability of the decomposition components.

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 σ 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 σ identically;
- if σ is a ground atom $P(\bar{t})$ then $\mathcal{M}$ and $\mathcal{M}'$ agree on interpretation $\bar{u}$ of $\bar{t}$ and for every vector of elements $\bar{v} \neq \bar{u}$, we have $\mathcal{M} \models P(\bar{v})$ iff $\mathcal{M}' \models P(\bar{v})$.

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

The following notion summarizes the well-known Definitions 1 and 7 in [14].

Definition 7 (Forgetting an atom or signature) *Let $\mathcal{T}$ be a theory in $\mathcal{L}$ and σ 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 σ in $\mathcal{T}$ (denoted by $\texttt{forget}\,(\mathcal{T},\sigma)$) 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 $\texttt{forget}\,(\mathcal{T}, \sigma)$ always exists, i.e. is second–order definable, for a finite set of formulas $\mathcal{T}$ in $\mathcal{L}$ and a finite signature or a ground atom σ (see [14], or Section 2.1 in [16]). 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 second-order 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 σ and π 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 σ and π 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 σ is a signature with $\sigma \cap \texttt{sig}\,(\mathcal{T}) = \varnothing$, or a ground atom with predicate not contained in $\texttt{sig}\,(\mathcal{T})$)*
- $\texttt{forget}\,(\mathcal{T} \cup \mathcal{T}',\sigma) \not\equiv \texttt{forget}\,(\mathcal{T},\sigma) \cup \texttt{forget}\,(\mathcal{T}',\sigma)$ *(see Example 3)*
- $\texttt{forget}\,(\varphi \vee \psi,\sigma) \equiv \texttt{forget}\,(\varphi,\sigma) \vee \texttt{forget}\,(\psi,\sigma)$ *(if φ and ψ are formulas in $\mathcal{L}$).*

Proposition 2 (Signature of forget($\mathcal{T}$,σ)) *Let $\mathcal{T}$ be a theory in $\mathcal{L}$, σ be a signature (or a ground atom, respectively) and 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).*

Proof. We consider the case when σ is a signature; the case of a ground atom is proved analogously. Assume that $\sigma \cap \mathtt{sig}\,(\mathtt{forget}\,(\mathcal{T},\sigma)) \neq \varnothing$. Denote by $\mathtt{forget}\,(\mathcal{T},\sigma)^*$ a "copy" of the set of formulas $\mathtt{forget}\,(\mathcal{T},\sigma)$, where each symbol from $\sigma \cup [\mathtt{sig}\,(\mathtt{forget}\,(\mathcal{T},\sigma)) \setminus \mathtt{sig}\,(\mathcal{T})]$ is uniquely replaced with a fresh symbol, not present in $\mathtt{sig}\,(\mathtt{forget}\,(\mathcal{T},\sigma))$. We claim that $\mathtt{forget}\,(\mathcal{T},\sigma)^* \models_{\mathcal{L}'} \mathtt{forget}\,(\mathcal{T},\sigma)$. There is nothing to prove if $\mathtt{forget}\,(\mathcal{T},\sigma)^*$ is unsatisfiable. Note that, by definition of forgetting, $\mathtt{forget}\,(\mathcal{T},\sigma)^*$ and $\mathtt{forget}\,(\mathcal{T},\sigma)$ are satisfiable iff $\mathcal{T}$ is. Let us assume that $\mathcal{T}$ is satisfiable. Take an arbitrary model $\mathcal{M}^* \models \mathtt{forget}\,(\mathcal{T},\sigma)^*$; then there exists a model $\mathcal{M}' \models \mathtt{forget}\,(\mathcal{T},\sigma)$ which agrees on $\mathtt{sig}\,(\mathtt{forget}\,(\mathcal{T},\sigma)^*)$ with $\mathcal{M}^*$ and interprets symbols from $\sigma \cup [\mathtt{sig}\,(\mathtt{forget}\,(\mathcal{T},\sigma)) \setminus \mathtt{sig}\,(\mathcal{T})]$ equally to the interpretation of the corresponding fresh symbols in $\mathcal{M}^*$. Therefore, we may assume that $\mathcal{M}^* \sim_\sigma \mathcal{M}'$. By definition of forgetting, there is a model $\mathcal{M} \models \mathcal{T}$ such that $\mathcal{M}' \sim_\sigma \mathcal{M}$, hence $\mathcal{M}^* \sim_\sigma \mathcal{M}$ and $\mathcal{M}^* \models \mathtt{forget}\,(\mathcal{T},\sigma)$. We have $\mathtt{forget}\,(\mathcal{T},\sigma)^* \models_{\mathcal{L}'} \mathtt{forget}\,(\mathcal{T},\sigma)$ and $\mathtt{sig}\,(\mathtt{forget}\,(\mathcal{T},\sigma)^*) \cap \mathtt{sig}\,(\mathtt{forget}\,(\mathcal{T},\sigma)) \subseteq \mathtt{sig}\,(\mathcal{T}) \setminus \sigma$. By PIP, there is a set of formulas Θ in signature $\mathtt{sig}\,(\mathcal{T}) \setminus \sigma$ such that $\mathtt{forget}\,(\mathcal{T},\sigma)^* \models_{\mathcal{L}'} \Theta$ and $\Theta \models_{\mathcal{L}'} \mathtt{forget}\,(\mathcal{T},\sigma)$. Note that $\mathtt{forget}\,(\mathcal{T},\sigma)^* \models_{\mathcal{L}'} \Theta$ yields $\mathtt{forget}\,(\mathcal{T},\sigma) \models_{\mathcal{L}'} \Theta$, because every model of $\mathtt{forget}\,(\mathcal{T},\sigma)$ can be expanded to a model of $\mathtt{forget}\,(\mathcal{T},\sigma)^*$ and the reduct of this model onto (a subset of) $\mathtt{sig}\,(\mathcal{T}) \setminus \sigma$ suffices to satisfy Θ. Thus, we conclude that $\mathtt{forget}\,(\mathcal{T},\sigma)$ is equivalent to Θ. $\square$

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

We note that the similar statement does not hold when σ is a ground atom. It follows from Proposition 2 that in case σ is a signature, $\mathtt{forget}\,(\mathcal{T},\sigma)$ axiomatizes the class of reducts of models of $\mathcal{T}$ onto the signature $\mathtt{sig}\,(\mathcal{T}) \setminus \sigma$. Clearly, if $\mathcal{T}$ is a theory in language $\mathcal{L}$, then $\mathtt{forget}\,(\mathcal{T},\sigma)$ 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 σ is a signature, $\mathtt{forget}\,(\mathcal{T},\sigma)$ is known as $\mathtt{sig}\,(\mathcal{T}) \setminus \sigma$–uniform interpolant of $\mathcal{T}$ wrt the language $\mathcal{L}$ and second-order queries, that is wrt the pair ($\mathcal{L}$, second-order logic), see Definition 13 in [10] and Lemma 39 in [19] for a justification. In other words, $\mathcal{T}$ and $\mathtt{forget}\,(\mathcal{T},\sigma)$ semantically entail the same second-order formulas in signature $\mathcal{T} \setminus \sigma$.

If σ is a ground atom $P(\bar{t})$ then, by definition, for any model $\mathcal{M} \models \mathcal{T}$, $\mathtt{forget}\,(\mathcal{T},\sigma)$ 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 σ, one can effectively construct the set of formulas $\mathtt{forget}\,(\mathcal{T},\sigma)$ in $\mathcal{L}$ such that $\mathtt{forget}\,(\mathcal{T},\sigma)$ is finitely axiomatizable iff $\mathcal{T}$ is. This follows from Theorem 4 in [14], 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 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 in case a theory $\mathcal{T}$ is finitely axiomatizable, computing $\texttt{forget}\,(\mathcal{T}, P(\bar{t}))$ in the way above doubles the size of 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 σ be a signature or a ground atom. Then the following holds:*

$$\begin{array}{ccc} \mathcal{T} & \models & \mathcal{T}_1 \\ \rotatebox{90}{$\models$} & & \rotatebox{90}{$\models$} \\ \texttt{forget}(\mathcal{T}, \sigma) & \models & \texttt{forget}(\mathcal{T}_1, \sigma) \end{array}$$

Proof. By definition of forgetting, every model of $\mathcal{T}$ is a model of $\texttt{forget}\,(\mathcal{T}, \sigma)$, so we have $\mathcal{T} \models \texttt{forget}\,(\mathcal{T}, \sigma)$ and similarly, $\mathcal{T}_1 \models \texttt{forget}\,(\mathcal{T}_1, \sigma)$. Now let $\mathcal{M}'$ be an arbitrary model of $\texttt{forget}\,(\mathcal{T}, \sigma)$. Then there is a model $\mathcal{M} \models \mathcal{T}$ such that $\mathcal{M} \sim_\sigma \mathcal{M}'$. Since $\mathcal{T} \models \mathcal{T}_1$, we have $\mathcal{M} \models \mathcal{T}_1$, so we conclude that $\mathcal{M}' \models \texttt{forget}\,(\mathcal{T}_1, \sigma)$, because $\mathcal{M}$ is a model satisfying the conditions of Definition 7 for $\mathcal{T}_1$ and $\mathcal{M}'$. □

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

Proof. From Proposition 3, we have $\mathcal{T} \models \texttt{forget}\,(\mathcal{T}, \sigma)$, thus $\texttt{forget}\,(\mathcal{T}, \sigma) \models \varphi$ yields $\mathcal{T} \models \varphi$. Now let $\mathcal{T} \models \varphi$ and assume there is a model $\mathcal{M}'$ of $\texttt{forget}\,(\mathcal{T}, \sigma)$ such that $\mathcal{M}' \not\models \varphi$. By definition of forgetting, there exists a model $\mathcal{M}$ of $\mathcal{T}$ such that $\mathcal{M} \sim_\sigma \mathcal{M}'$, i.e. $\mathcal{M}$ and $\mathcal{M}'$ have the same universe and may differ only on interpretation of signature σ (ground atom σ). By the condition on signature of φ, then $\mathcal{M}$ is not a model of φ, which contradicts $\mathcal{T} \models \varphi$. □

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

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

Proof. Let φ be a formula with $\mathtt{sig}\,(\varphi) \subseteq \Delta$ such that $\mathtt{forget}\,(\mathcal{T}_1, \sigma) \models \varphi$. By Proposition 2, we may assume that for $i = 1, 2$ the signature of $\mathtt{forget}\,(\mathcal{T}_i, \sigma)$ is a subset of $\mathtt{sig}\,(\mathcal{T}_i) \setminus \sigma$.

$$\begin{array}{ccc}
\mathcal{T}_1 \models \varphi & \overset{\mathtt{sig}(\varphi)\,\subseteq\,\Delta}{\text{iff}} & \mathcal{T}_2 \models \varphi \\
\rotatebox[origin=c]{-90}{$\models$} & & \rotatebox[origin=c]{-90}{$\models$} \\
\mathtt{forget}(\mathcal{T}_1, \sigma) \models \overset{\Delta\,\setminus\,\sigma}{\mathcal{T}_1'} \models \varphi & \text{iff} & \mathtt{forget}(\mathcal{T}_2, \sigma) \models \varphi
\end{array}$$

As $\mathtt{forget}\,(\mathcal{T}_1, \sigma) \models \varphi$, by PIP, there is a set of formulas $\mathcal{T}_1'$ with $\mathtt{sig}\,(\mathcal{T}_1') \subseteq \mathtt{sig}\,(\mathtt{forget}\,(\mathcal{T}_1, \sigma)) \cap \mathtt{sig}\,(\varphi) \subseteq \Delta \setminus \sigma$ such that $\mathtt{forget}\,(\mathcal{T}_1, \sigma) \models \mathcal{T}_1'$ and $\mathcal{T}_1' \models \varphi$. Therefore, by Proposition 4, we have $\mathcal{T}_1 \models \mathcal{T}_1'$. Since $\mathcal{T}_1$ and $\mathcal{T}_2$ are Δ–inseparable and $\mathtt{sig}\,(\mathcal{T}_1') \subseteq \Delta$, we obtain $\mathcal{T}_2 \models \mathcal{T}_1'$. Again, since $\mathtt{sig}\,(\mathcal{T}_1') \cap \sigma = \varnothing$, by Proposition 4, we conclude that $\mathtt{forget}\,(\mathcal{T}_2, \sigma) \models \mathcal{T}_1'$ and thus, $\mathtt{forget}\,(\mathcal{T}_2, \sigma) \models \varphi$. $\square$

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

Example 2 (Δ–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 Δ–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 $\mathcal{ELO}^{\perp}$, i.e. the sub-boolean logic $\mathcal{EL}$ augmented with nominals and the bottom concept $\perp$. Let $\Sigma = \{P, a, c\}$ be 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 Σ as $\{\{a\} \sqcap \{c\} \sqsubseteq \perp,\ \{c\} \sqsubseteq \exists P.\{a\},\ \top \sqsubseteq \exists P.\top\}$. 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 the 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 $\mathcal{ELO}^{\perp}$; thus, $\mathcal{T}_1$ and $\mathcal{T}_2$ are Δ–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 $\mathcal{ELO}^{\perp}$ into the language of second-order logic). We verify that they are not Δ–inseparable and the formula $\top \sqsubseteq \exists P.\top$ is the witness 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 $\texttt{forget}\,(\mathcal{T}_2, P(c,c))$, 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 Δ–inseparability* of the initial sets of formulas (see Definition 11 in [10]) which is far from being decided effectively from the computational point of view (see Theorem 3 in [17], Lemma 40 in [19]). Semantic Δ–inseparability is strictly stronger than (syntactic) inseparability from Definition 1. 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 Δ–inseparability under forgetting) *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two sets of formulas in $\mathcal{L}$ with $\texttt{sig}\,(\mathcal{T}_1) \cap \texttt{sig}\,(\mathcal{T}_2) = \Delta$ for a signature Δ 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 Δ–inseparable;*
- *for σ a signature or a ground atom, $\texttt{forget}\,(\mathcal{T}_1, \sigma)$ and $\texttt{forget}\,(\mathcal{T}_2, \sigma)$ satisfy (*) as well.*

Proof. Δ–inseparability is the immediate consequence of (*): if φ is a formula with $\texttt{sig}\,(\varphi) \subseteq \Delta$, $\mathcal{T}_1 \models \varphi$, but $\mathcal{T}_2 \not\models \varphi$, then there is a model $\mathcal{M}_2$ of $\mathcal{T}_2$ such that $\mathcal{M}_2 \not\models \varphi$. Then there is an expansion $\mathcal{M}$ of $\mathcal{M}_2$ such that $\mathcal{M} \models \mathcal{T}_1 \cup \mathcal{T}_2$, $\mathcal{M}\mid_{\texttt{sig}\,(\mathcal{T}_1)} \models \mathcal{T}_1$, but $\mathcal{M}\mid_{\texttt{sig}\,(\mathcal{T}_1)} \not\models \varphi$, a contradiction. Now let us verify that for $i = 1, 2$, any model of $\texttt{forget}\,(\mathcal{T}_i, \sigma)$ can be expanded to a model of $\texttt{forget}\,(\mathcal{T}_1, \sigma) \cup \texttt{forget}\,(\mathcal{T}_2, \sigma)$. For instance, let $\mathcal{M}_2'$ be a model of $\texttt{forget}\,(\mathcal{T}_2, \sigma)$. Consider a model $\mathcal{M}_2$ of $\mathcal{T}_2$ such that $\mathcal{M}_2 \sim_\sigma \mathcal{M}_2'$ and expand it to a model $\mathcal{M}$ of $\mathcal{T}_1 \cup \mathcal{T}_2$. Then, by definition of forgetting, there must be a model $\mathcal{M}' \models \texttt{forget}\,(\mathcal{T}_1, \sigma)$ with $\mathcal{M}' \sim_\sigma \mathcal{M}$ which agrees with $\mathcal{M}_2'$ on σ (if σ is a signature), or on the predicate of σ (if σ is a ground atom). By construction, $\mathcal{M}'$ is an expansion of $\mathcal{M}_2'$ and thus a model for $\texttt{forget}\,(\mathcal{T}_1, \sigma) \cup \texttt{forget}\,(\mathcal{T}_2, \sigma)$. $\square$

Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two sets of formulas in $\mathcal{L}$ with $\texttt{sig}\,(\mathcal{T}_1) \cap \texttt{sig}\,(\mathcal{T}_2) = \Delta$ for a signature Δ and let σ be either a subsignature of Δ or a ground atom with the predicate from Δ. It is known that in general, forgetting σ may not be distributive over union of sets of formulas. The entailment $\texttt{forget}\,(\mathcal{T}_1 \cup \mathcal{T}_2, \sigma) \models \texttt{forget}\,(\mathcal{T}_1, \sigma) \cup \texttt{forget}\,(\mathcal{T}_2, \sigma)$ 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 Δ). 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 Δ–inseparable and for $i = 1, 2$, $\mathtt{Cons}\,(\mathcal{T}_i, \Delta)$ is the set of tautologies in Δ. 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 σ be either a signature or a ground atom. Then the following statements are equivalent:*

- $\mathtt{forget}\,(\mathcal{T}_1, \sigma) \cup \mathtt{forget}\,(\mathcal{T}_2, \sigma) \models \mathtt{forget}\,(\mathcal{T}_1 \cup \mathcal{T}_2, \sigma)$
- *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 $i = 1, 2$.*

Proof. Note that in the second condition, the requirement $\mathcal{M} \sim_\sigma \mathcal{M}_i$ for some $i = 1, 2$ is equivalent to $\mathcal{M} \sim_\sigma \mathcal{M}_i$ for all $i = 1, 2$, by transitivity of $\sim_\sigma$. ($\Rightarrow$): Let $\mathcal{M}_1 \models \mathcal{T}_1$ and $\mathcal{M}_2 \models \mathcal{T}_2$ be models with $\mathcal{M}_1 \sim_\sigma \mathcal{M}_2$. Then there are models $\mathcal{M}'_1$ and $\mathcal{M}'_2$ such that for $i = 1, 2$, $\mathcal{M}'_i \models \mathtt{forget}\,(\mathcal{T}_i, \sigma)$ and $\mathcal{M}'_i \sim_\sigma \mathcal{M}_i$. Then, by transitivity of $\sim_\sigma$, for all $i, j = 1, 2$ we have $\mathcal{M}'_i \sim_\sigma \mathcal{M}_j$ and thus, $\mathcal{M}'_i \models \mathtt{forget}\,(\mathcal{T}_j, \sigma)$. Then $\mathcal{M}'_1 \models \mathtt{forget}\,(\mathcal{T}_1 \cup \mathcal{T}_2, \sigma)$, so there exists a model $\mathcal{M} \models \mathcal{T}_1 \cup \mathcal{T}_2$ such that $\mathcal{M} \sim_\sigma \mathcal{M}'_1$ and hence, $\mathcal{M} \sim_\sigma \mathcal{M}_1$. ($\Leftarrow$):Let $\mathcal{M}'$ be a model of $\mathtt{forget}\,(\mathcal{T}_1, \sigma) \cup \mathtt{forget}\,(\mathcal{T}_2, \sigma)$. There exist models $\mathcal{M}_1$ and $\mathcal{M}_2$ such that for $i = 1, 2$, $\mathcal{M}_i \models \mathcal{T}_i$ and $\mathcal{M}_i \sim_\sigma \mathcal{M}'$. Then $\mathcal{M}_1 \sim_\sigma \mathcal{M}_2$, hence, there must be a model $\mathcal{M}$ of $\mathcal{T}_1 \cup \mathcal{T}_2$ with $\mathcal{M} \sim_\sigma \mathcal{M}_i$ for some $i = 1, 2$. Then we obtain that $\mathcal{M} \sim_\sigma \mathcal{M}'$ and thus, by definition of forgetting, $\mathcal{M}'$ is a model of $\mathtt{forget}\,(\mathcal{T}_1 \cup \mathcal{T}_2, \sigma)$. $\square$

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 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 $\mathtt{sig}\,(\mathcal{T}_1) \cap \mathtt{sig}\,(\mathcal{T}_2) = \Delta$ for a signature Δ and σ be either a subsignature of $\mathtt{sig}\,(\mathcal{T}_1) \setminus \Delta$ or a ground atom with the predicate from $\mathtt{sig}\,(\mathcal{T}_1) \setminus \Delta$. Then $\mathtt{forget}\,(\mathcal{T}_1 \cup \mathcal{T}_2, \sigma)$ is equivalent to $\mathtt{forget}\,(\mathcal{T}_1, \sigma) \cup \mathcal{T}_2$. Moreover, if $\mathcal{T}_1$ and $\mathcal{T}_2$ are Δ–inseparable, then so are $\mathtt{forget}\,(\mathcal{T}_1,\, \sigma)$ and $\mathcal{T}_2$.*

Proof. Note that by the choice of σ, $\mathcal{T}_2$ is equivalent to $\mathtt{forget}\,(\mathcal{T}_2, \sigma)$ and thus, by Proposition 3, it suffices to verify the entailment $\mathtt{forget}\,(\mathcal{T}_1, \sigma) \cup \mathtt{forget}\,(\mathcal{T}_2, \sigma) \models \mathtt{forget}\,(\mathcal{T}_1 \cup \mathcal{T}_2, \sigma)$. If there are models $\mathcal{M}_1 \models \mathcal{T}_1$ and $\mathcal{M}_2 \models \mathcal{T}_2$, with $\mathcal{M}_1 \sim_\sigma$

$\mathcal{M}_2$, then in fact, $\mathcal{M}_1 \models \mathcal{T}_1 \cup \mathcal{T}_2$, by the choice of σ and definition of $\sim_\sigma$. Thus, the criterion from Proposition 7 obviously yields the required entailment. It only remains to note that Δ–inseparability of $\texttt{forget}\,(\mathcal{T}_1, \sigma)$ and $\texttt{forget}\,(\mathcal{T}_2, \sigma)$ follows from the choice of σ, Proposition 4, and Δ–inseparability of $\mathcal{T}_1$ and $\mathcal{T}_2$. $\square$

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 Δ's (*deltas*) of components of initial theories. We will consider local-effect SSAs discussed in [16] and *deltas*, which do not contain fluents.

We use the following notations further in this paper. For a ground action term α in the language of the situation calculus, we denote by S_α the situation term $do(\alpha, S_0)$. To define progression, we introduce an equivalence relation on many-sorted structures in the situation calculus signature. For two structures $\mathcal{M}$, $\mathcal{M}'$ and a ground action α, 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_α, i.e. for every fluent F and every variable assignment θ, we have $\mathcal{M}, \theta \models F(\bar{x}, S_\alpha)$ iff $\mathcal{M}', \theta \models F(\bar{x}, S_\alpha)$.

Definition 8 (Progression, modified Definition 9.1.1 in [25]) *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 α 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 α if it is uniform in the situation term S_α and for any structure $\mathcal{M}$, $\mathcal{M}$ is a model of $\Sigma \cup \mathcal{D}_{ss} \cup \mathcal{D}_{ap} \cup \mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha}$ 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 α, 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). It can be seen (Theorem 2 in [15] and Theorem 2.10 in [16]) that progression always exists, i.e. is second-order definable, if the signature of $\mathcal{BAT}$ is finite and the initial theory $\mathcal{D}_{S_0}$ is finitely axiomatizable. On the other hand, by the definition, 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 [26]), 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.

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_α, thus informally, $\mathcal{D}_{S_\alpha}$ is an information about the situation S_α implied by $\mathcal{BAT}$. Moreover, it contains all information from $\mathcal{BAT}$ about the situation S_α, as guaranteed by the model-theoretic property with the 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 α. In particular, the initial theory of $\mathcal{BAT}$ can be replaced with $\mathcal{D}_{S_\alpha}(S_0/S_\alpha)$ (recall the notation from Section 2.2) which gives a new $\mathcal{BAT}$, with S_α 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_α 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, $\mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha}(S_0/S_\alpha) \models \varphi(S_0)$) 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 [16], while here we only introduce necessary notations from Definition 3.4 of [16] 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 α be a ground action term. Denote

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

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

Note that $\Omega(S_0)$ is a finite set of ground atoms 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 α. 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 [16]) *In the notations above, the following is a progression of $\mathcal{D}_{S_0}$ wrt α in the sense of Definition 8:*

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

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 [16], 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 [13]. 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 [16], 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 the 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_α which may occur in every formula of $\mathcal{D}_{S_\alpha}$ (thus potentially spoiling decomposability), we consider the mentioned decomposability and inseparability properties regarding the theory $\mathcal{D}_{S_\alpha}(S_0/S_\alpha)$ 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 Δ–decomposability.

Consider a $\mathcal{BAT}$ $\mathcal{D}$ with Δ–decomposable initial theory $\mathcal{D}_{S_0}$ for a signature Δ. 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 general enough to easily formulate negative results showing that the mentioned properties are not preserved without stipulations.

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 Δ–inseparability of components of the initial theory $\mathcal{D}_{S_0}$ can be easily lost when fluents are present in Δ (Example 5). The third observation is that even if there are no fluents in Δ, 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 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 Δ does not contain fluents and the components of $\mathcal{D}_{S_0}$ do not split after progression, then Δ–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 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 binary 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 $Taut(c_2)$, 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 $\mathcal{D}_{S_\alpha}(S_0/S_\alpha)$ (and $\mathcal{D}_{S_\alpha}$, as well) is not Δ–decomposable theory (for any Δ).

For a signature Δ, 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 Δ–decomposable into finite Δ–inseparable components, such that progression $\mathcal{D}_{S_\alpha}(S_0/S_\alpha)$ of $\mathcal{D}_{S_0}$ wrt $A(c)$ (with term S_α substituted with S_0) is finitely axiomatizable and Δ–decomposable, but the decomposition components are no longer Δ–inseparable, unless we allow them to be infinite.

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

- $\mathcal{D}_{ss} = \{F(x, do(a, s)) \equiv (a = A(x)) \wedge P(x) \wedge Q(d) \ \vee \ F(x, s)\}$;
- $\mathcal{D}_{S_0} = \mathcal{D}_1 \cup \mathcal{D}_2$, with
  - $\mathcal{D}_1 = \{Taut(F, R, S_0, b),\ \neg F(x, S_0)\}$, where $Taut(F, R, S_0, b)$ is a tautological formula in the signature $\{F, R, S_0, b\}$ 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 Δ–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 $\mathtt{sig}(\mathcal{D}_2) \setminus \Delta = \{P\}$. It is also easy to check that $\mathcal{D}_1$ and $\mathcal{D}_2$ are Δ–inseparable.

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 situation constant S_0 into SSA. As $\mathcal{D}_{S_0} \models \neg F(x, S_0)$, 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 φ.

By Proposition 8 it is easy to verify that the union of $\{Taut(F, R, S_0, b)\}$ and $\mathcal{D}_2' = (\mathcal{D}_2 \setminus \{\neg F(x, S_0)\}) \cup \{\varphi,\ x \neq c \to \neg F(x, do(A(c), S_0))\}$ is a progression $(\mathcal{D}_{S_\alpha})$ of $\mathcal{D}_{S_0}$ wrt $A(c)$.

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

Note that in the example above, the initial theory $\mathcal{D}_{S_0}$ is in fact $\varnothing$–decomposable, with one signature component being tautological in signature $\{b\}$ and the other component containing the rest of the symbols. Clearly, the 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, S_0, b in the theory $\mathcal{D}_1$. We appeal to a similar observation in Example 6.

Example 6 (Split of a component and loss Δ–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 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:
  - $D(x) \vee R(x, y) \to F_1(c, S_0)$

- $D(x) \rightarrow P(x)$
- $P(x) \rightarrow \exists y(R(x,y) \wedge P(y))$

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

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

By definition, $\mathcal{D}_{S_0}$ is Δ–decomposable into Δ–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_α

$\mathcal{D}''_1$ 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_α

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

Clearly, $\mathcal{D}_{S_\alpha}(S_0/S_\alpha)$ is Δ–decomposable. Note that after progression the component $\mathcal{D}_1$ is "splitted" into $\mathcal{D}'_1(S_0/S_\alpha)$ and $\mathcal{D}''_1(S_0/S_\alpha)$ and these theories are not Δ–inseparable (similarly, $\mathcal{D}'_1(S_0/S_\alpha)$ and $\mathcal{D}'_2(S_0/S_\alpha)$). By Fact 2, it can be shown that they can not be made Δ–inseparable while remaining finitely axiomatizable.

To formulate the theorems below, we let $\mathcal{D}$ denote a $\mathcal{BAT}$ with the initial theory $\mathcal{D}_{S_0}$, the set of successor state axioms $\mathcal{D}_{ss}$, and the unique name assumption theory $\mathcal{D}_{una}$. Example 5 motivates the following definition.

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

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 Δ–inseparability can be guaranteed only for fluent-free signatures Δ. 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 Δ–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 Δ–consequences. Thereby, we formulate the idea that components of progression do not split Δ–consequences of the components of the initial theory (cf. Example 6).

Theorem 1 (Preservation of Δ-insep. for fluent-free Δ) *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 σ–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 Δ–inseparable;*
- *$\mathcal{D}_{S_\alpha}(S_0/S_\alpha)$ 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 Δ–inseparable.

Proof. Let us demonstrate that for all $j \in J$ we have $\mathtt{Cons}\,(\mathcal{D}_{una} \cup D'_j, \Delta) = \mathtt{Cons}\,(\mathcal{D}_{una} \cup \mathcal{D}_{S_0}, \Delta)$, from which the statement of the theorem obviously, follows. Essentially, we prove the following inclusions (the corresponding points of the proof are marked with circles):

$$
\begin{array}{ccccc}
 & 3 & & 2 & \\
\mathtt{Cons}\,(\mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha}(S_0/S_\alpha), \Delta) & \subseteq & \mathtt{Cons}\,(\mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha}, \Delta) & \subseteq & \mathtt{Cons}\,(\mathcal{D}_{una} \cup \mathcal{D}_{S_0}, \Delta) \\
4 \ \cup\!| \ \boxed{\text{Theorem conditions}} & & \boxed{\text{Theorem conditions}} & & |\!\cap \ 1 \\
\mathtt{Cons}\,(\mathcal{D}_{una} \cup \mathcal{D}'_j, \Delta) & & \supseteq & & \mathtt{Cons}\,(\mathcal{D}_{una} \cup \mathcal{D}_i, \Delta) \\
 & & 4 & &
\end{array}
$$

1) Note that for any $i \in I$, $\mathcal{D}_{S_0}$ is σ–decomposable with components D_i and $\bigcup_{k \in I \setminus \{i\}} D_k$. We claim that $\mathcal{D}_{una} \cup D_i$ and $\mathcal{D}_{una} \cup \bigcup_{k \in I \setminus \{i\}} D_k$ are Δ–inseparable. Let φ be a formula in signature Δ. If $\mathcal{D}_{una} \cup D_i \models \varphi$ then clearly, $\mathcal{D}_{una} \cup \bigcup_{k \in I \setminus \{i\}} D_k \models \varphi$ by Δ–inseparability from the condition of the theorem. On the other hand, if $\mathcal{D}_{una} \cup \bigcup_{k \in I \setminus \{i\}} D_k \models \varphi$ then by PIP we have $\mathcal{T}_{una} \cup \bigcup_{k \in I \setminus \{i\}} \mathcal{T}_k \models \varphi$, where $\mathcal{D}_{una} \models \mathcal{T}_{una}$, $\mathtt{sig}\,(\mathcal{T}_{una}) \subseteq \mathtt{sig}\,(\mathcal{D}_{una})$ and $D_k \models \mathcal{T}_k$ for $k \in I \setminus \{i\}$, $\mathtt{sig}\,(\mathcal{T}_k) \subseteq \Delta$. Again, by Δ–inseparability, for each $k \in I \setminus \{i\}$ we have $\mathcal{D}_{una} \cup D_i \models \mathcal{T}_k$ and thus, $\mathcal{D}_{una} \cup D_i \models \varphi$.

Therefore, if $\varphi \in \mathtt{Cons}\,(\mathcal{D}_{una} \cup \mathcal{D}_{S_0}, \Delta)$, then for every $i \in I$, $[\mathcal{D}_{una} \cup \bigcup_{k \in I \setminus \{i\}} D_k] \cup [\mathcal{D}_{una} \cup D_i] \models \varphi$ and then by PIP and inseparability shown above, $\mathcal{D}_{una} \cup D_i \models \varphi$. Since $\mathcal{D}_{S_0} \models \bigcup_{i \in I} D_i$ by decomposability, we obtain $\mathtt{Cons}\,(\mathcal{D}_{una} \cup \mathcal{D}_{S_0}, \Delta) = \mathtt{Cons}\,(\mathcal{D}_{una} \cup D_i, \Delta)$ for all $i \in I$.

2) Let us show that $\mathtt{Cons}\,(\mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha}, \varDelta) \subseteq \mathtt{Cons}\,(\mathcal{D}_{una} \cup \mathcal{D}_{S_0}, \varDelta)$. First, take a formula $\psi \in \mathtt{Cons}\,(\mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha}, \varDelta)$ which does not contain situation terms. From the definition of progression, every model of $\mathcal{D}$ is a model of $\mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha}$, so $\mathcal{D} \models \mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha}$ and hence, $\mathcal{D} \models \psi$. Assume $\mathcal{D}_{una} \cup \mathcal{D}_{S_0} \not\models \psi$, then $\mathcal{D}_{una} \cup \mathcal{D}_{S_0} \cup \{\neg\psi\}$ is satisfiable and since ψ is a uniform formula, by Proposition 1, $\mathcal{D} \cup \{\neg\psi\}$ is satisfiable, which contradicts $\mathcal{D} \models \psi$. Therefore, $\mathcal{D}_{una} \cup \mathcal{D}_{S_0} \models \psi$.

It remains to verify that the set $\mathtt{Cons}\,(\mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha}, \varDelta)$ is axiomatized by sentences which do not contain situation terms. We have $\varDelta = \mathtt{sig}\,(\mathcal{D}_{una}) \cup \sigma \subseteq \mathtt{sig}\,(\mathcal{D}_{una}) \cup \mathtt{sig}\,(\mathcal{D}_{S_0})$, so $\{do, \preceq, Poss\} \cap \varDelta = \varnothing$, by definition of $\mathcal{D}_{una}$ and $\mathcal{D}_{S_0}$. As σ if fluent-free by the condition of the theorem (and $\mathtt{sig}\,(\mathcal{D}_{una})$ is fluent-free by definition of $\mathcal{BAT}$), $\varDelta$ may contain only situation–independent predicates and functions. Thus, any formula $\varphi \in \mathtt{Cons}\,(\mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha}, \varDelta)$ may contain situation terms only in equalities, where each term is either the constant S_0 (in case $S_0 \in \sigma$) or a bound variable of sort situation. Suppose that this is the case and there is no $\psi \in \mathtt{Cons}\,(\mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha}, \varDelta)$ such that $\psi \models \varphi$ and ψ does not contain situation terms. By the syntax of $\mathcal{L}_{sc}$ and the choice of $\varDelta$, then φ is a boolean combination of formulas without situation terms and sentences over signature $\{S_0\}$ stating that φ has a model with cardinality $|Sit|$ of sort *situation* lying in the interval $[n, m]$ for $n \in \omega$ and $m \in \omega \cup \{\infty\}$. Denote sentences of this form by $\exists^{[n,m]}\theta_=$. We may assume that φ is in conjunctive normal form and there is a formula ξ, a boolean combination of $\exists^{[n,m]}\theta_=$ such that $\not\models \xi$, $\not\models \neg\xi$, either ξ or $\xi \vee \eta$ is a conjunct of φ, and $\eta \notin \mathtt{Cons}\,(\mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha}, \varDelta)$, $\mathtt{sig}\,(\eta) \subseteq \varDelta$, is a formula without situation terms. As $\not\models \xi$ and $\not\models \neg\xi$, there are $n, m \in \omega$ such that ξ does not have a model with $|Sit| = n$ and $\neg\xi$ does not have a model with $|Sit| = m$. Then by Lemma 1, we conclude that $\mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha} \not\models \xi$ and $\mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha} \not\models \neg\xi$. In particular, ξ can not be a conjunct of φ. If $\xi \vee \eta$ is a conjunct, then there exists a model $\mathcal{M}$ of $\mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha}$ such that $\mathcal{M} \models \xi$ and $\mathcal{M} \not\models \eta$. Then, by applying Lemma 1 again, there must be a model $\mathcal{M}'$ of $\mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha}$ with $|Sit| = n$ where the interpretation of situation–independent predicates and functions is the same as in $\mathcal{M}$. Thus, $\mathcal{M}' \not\models \xi$ and since η does not contain situation terms, $\mathcal{M}' \not\models \eta$, which contradicts $\mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha} \models \varphi$.

3) Now let us demonstrate that $\mathtt{Cons}\,(\mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha}(S_0/S_\alpha), \varDelta) \subseteq \mathtt{Cons}\,(\mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha}, \varDelta)$. Note that $\mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha}(S_0/S_\alpha)$ is uniform in S_0, so, following the above proved, assume there is a formula $\varphi \in \mathtt{Cons}\,(\mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha}(S_0/S_\alpha), \varDelta)$ such that φ does not contain situation terms and $\mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha} \not\models \varphi$. Take a model $\mathcal{M}$ of $\mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha}$ such that $\mathcal{M} \not\models \varphi$. Then, by Lemma 1, there exists a model $\mathcal{M}'$ of $\mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha}$ such that the domain for sort *situation* in $\mathcal{M}'$ is a singleton set (so, the interpretation of terms S_0 and S_α coincide in $\mathcal{M}'$) and the interpretation of situation–independent symbols is the same in $\mathcal{M}$ and $\mathcal{M}'$. Then $\mathcal{M}' \not\models \varphi$, but clearly, $\mathcal{M}' \models \mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha}(S_0/S_\alpha)$ which contradicts the assumption $\mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha} \not\models \varphi$.

4) Finally, by the condition of the theorem, for all $j \in J$, we have $\mathcal{D}_{una} \cup D'_j \subseteq \mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha}\ (S_0/S_\alpha)$ and from points 1–3 above we obtain $\mathtt{Cons}\,(\mathcal{D}_{una} \cup \mathcal{D}_{S_\alpha}(S_0/S_\alpha), \varDelta) \subseteq \mathtt{Cons}\,(\mathcal{D}_{una} \cup \mathcal{D}_{S_0}, \varDelta)$. Hence, for all $j \in J$ we have $\mathtt{Cons}\,(\mathcal{D}_{una} \cup D'_j, \varDelta) \subseteq \mathtt{Cons}\,(\mathcal{D}_{una} \cup \mathcal{D}_{S_0}, \varDelta)$. On the other hand, we also have $\mathtt{Cons}\,(\mathcal{D}_{una} \cup$

$D_i, \Delta) \subseteq \mathtt{Cons}\,(\mathcal{D}_{una} \cup D'_j, \Delta)$ from the condition of the theorem. Therefore from the inclusion $\forall\ i \in I\ \mathtt{Cons}\,(\mathcal{D}_{una} \cup \mathcal{D}_{S_0}, \Delta) \subseteq \mathtt{Cons}\,(\mathcal{D}_{una} \cup D_i, \Delta)$ of point 1 we conclude that $\mathtt{Cons}\,(\mathcal{D}_{una} \cup D'_j, \Delta) = \mathtt{Cons}\,(\mathcal{D}_{una} \cup \mathcal{D}_{S_0}, \Delta)$ for all $j \in J$. □

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. A finite set $\mathcal{D}_{ss}$ of the SSAs is considered to be syntactically divided into the union of $|I|$ subtheories sharing some fluent-free signature Δ_1 (that may include actions, static predicates, and object constants), the initial theory $\mathcal{D}_{S_0}$ is Δ_2–decomposable, for a fluent-free signature Δ_2, into $|J|$ components, and the subtheories of $\mathcal{D}_{ss}$ are aligned with the components of $\mathcal{D}_{S_0}$ via syntactic occurrences of fluents. 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 Δ_1 and Δ_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 the situation calculus.

Theorem 2 (Preservation of components in local-effect $\mathcal{BAT}$) *Let $\mathcal{D}$ be a local-effect $\mathcal{BAT}$, with $\mathcal{D}_{S_0}$ an initial theory in first-order logic. Let Δ_1, Δ_2 be fluent-free signatures, $do \notin \Delta_1$, and $\alpha = A(\bar{c})$, be a ground action term. Denote $\Delta = \Delta_1 \cup \Delta_2 \cup \{c_1, \dots, c_k\}$, if $\bar{c} = \langle c_1, \dots, c_k \rangle$, 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 $\{D_i\}_{i \in I}$, with $\mathtt{sig}\,(D_n) \cap \mathtt{sig}\,(D_m) \subseteq \Delta_1 \cup \{do\}$ for all $n, m \in I \neq \varnothing$, $n \neq m$;*
- *$\mathcal{D}_{S_0}$ is Δ_2–decomposable into finite components $\{D'_j\}_{j \in J}$ 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 $\mathcal{D}_{S_\alpha}(S_0/S_\alpha)$ is Δ–decomposable. If the components $\{D'_j\}_{j \in J}$ are pairwise Δ–inseparable, then so are the components of $\mathcal{D}_{S_\alpha}(S_0/S_\alpha)$ in the corresponding decomposition.

Proof. By definition of $\mathcal{BAT}$, for every $i \in I$, we have $\mathtt{sig}\,(D_i) \cap \mathcal{F} \neq \varnothing$ and thus, from the conditions of the theorem, $\mathtt{sig}\,(D_i) \cap \mathtt{sig}\,(\mathcal{D}_{S_0}) \neq \varnothing$, $\mathtt{sig}\,(\mathcal{D}_{ss}) \cap \mathcal{F} = \mathtt{sig}\,(\mathcal{D}_{S_0}) \cap \mathcal{F}$. Hence, for every $i \in I$ there is $j \in J$ such that $\mathtt{sig}\,(D_i) \cap \mathcal{F} \subseteq \mathtt{sig}\,(D'_j)$. Moreover, such $j \in J$ is unique for every $i \in I$, because otherwise there would exist $n, m \in J$, $n \neq m$, such that $\mathtt{sig}\,(D'_n) \cap \mathtt{sig}\,(D'_m) \cap \mathcal{F} \neq \varnothing$, which contradicts the condition that Δ_2 is fluent-free. Therefore, there is a map $f : I \to J$ such that for every $i \in I$, $\mathtt{sig}\,(D_i) \cap \mathcal{F} \subseteq \mathtt{sig}\,(D'_{f(i)})$. Note that there may exist $j \in J$ such that $\mathtt{sig}\,(D'_j) \cap \mathcal{F} = \varnothing$ and in this case j is the image of no $i \in I$. Let us denote the image of f by $\tilde{J}$ (so, $\tilde{J} \subseteq J$).

Now for every $i \in I$ consider the set of formulas $D_i[\Omega]$, the instantiation of D_i w.r.t. $\Omega(S_0)$, and for each $j \in \tilde{J}$ denote $\widetilde{D}_j = [\ \bigcup_{i \in f^{-1}(j)} (D_i[\Omega])\ ] \cup D'_j$.

Then, by Proposition 8, progression $\mathcal{D}_{S_\alpha}$ of $\mathcal{D}_{S_0}$ wrt α is logically equivalent to

$$[\ \mathtt{forget}\,(\bigcup_{j\in\tilde{J}} \widetilde{D}_j, \Omega(S_0))\ \cup \bigcup_{j\in J\setminus\tilde{J}} D'_j\ ]\ (S_0/S_\alpha).$$

As Δ_1 and Δ_2 are fluent-free, the signatures $\{\mathtt{sig}\,(\widetilde{D}_j)\}_{j\in\tilde{J}}$ do not have fluents in common and thus, by Corollary 2, $\mathcal{D}_{S_\alpha}$ is equivalent to

$$[\ \bigcup_{j\in\tilde{J}} \mathtt{forget}\,(\widetilde{D}_j, \Omega(S_0)\mid_j)\ \cup \bigcup_{j\in J\setminus\tilde{J}} D'_j\ ]\ (S_0/S_\alpha),$$

where for $j\in\tilde{J}$, $\Omega(S_0)\mid_j$ is the subset of ground atoms from $\Omega(S_0)$ with fluents from $\mathtt{sig}\,(D'_j)$. For all $j\in J\setminus\tilde{J}$, we have $\mathtt{sig}\,(D'_j)\cap\mathcal{F}=\varnothing$ and D'_j is uniform in S_0, so it follows that $S_0\notin\mathtt{sig}\,(D'_j)$ and thus, $\mathcal{D}_{S_\alpha}(S_0/S_\alpha)$ is equivalent to the union

$$[\ \bigcup_{j\in\tilde{J}} \mathtt{forget}\,(\widetilde{D}_j, \Omega(S_0)\mid_j)\ ]\ (S_0/S_\alpha)\ \cup \bigcup_{j\in J\setminus\tilde{J}} D'_j.$$

For every $j\in J$, let D''_j be the set of formulas $(\mathtt{forget}\,(\widetilde{D}_j,\Omega(S_0)\mid_j))(S_0/S_alpha)$ (in case $j\in\tilde{J}$) or the set of formulas D'_j (if $j\in J\setminus\tilde{J}$). So $\mathcal{D}_{S_\alpha}(S_0/S_\alpha)$ is equivalent to $\bigcup_{j\in J} D''_j$. From the syntactic definition of forgetting a set of ground atoms and the substitution of S_α with S_0 it follows that the pairwise intersection of any signatures from $\{\mathtt{sig}\,(D''_j)\}_{j\in J}$ is a subset of Δ. Then $\{D''_j\cup Taut(\Delta,j)\}_{j\in J}$ is Δ–decomposition of $\mathcal{D}_{S_\alpha}(S_0/S_\alpha)$, where for each $j\in J$, $Taut(\Delta,j)$ is a set of tautologies in signature $\Delta\setminus\mathtt{sig}\,(D''_j)$ which are uniform in S_0.

It remains to verify that the sets of formulas from $\{D''_j\}_{j\in J}$ are pairwise Δ–inseparable, if so are the components of $\mathcal{D}_{S_0}$.

1) First, consider the sets from the union

$$\bigcup_{j\in\tilde{J}} \widetilde{D}_j\ \cup \bigcup_{j\in J\setminus\tilde{J}} D'_j. \qquad (\dagger)$$

The pairwise intersection of their signatures is contained in $\Delta\cup\mathtt{sig}\,(S_\alpha)$. We claim that the sets from this union are pairwise Δ–inseparable.

By our definition, for all $j\in\tilde{J}$ we have $D'_j\subseteq\widetilde{D}_j$ and hence, $\mathtt{Cons}\,(D'_j,\Delta)\subseteq\mathtt{Cons}\,(\widetilde{D}_j,\Delta)$, so let us check that $\mathtt{Cons}\,(\widetilde{D}_j,\Delta)\subseteq\mathtt{Cons}\,(D'_j,\Delta)$ for every $j\in\tilde{J}$. Each formula in $D_i[\Omega]$, $i\in f^{-1}(j)$, $j\in\tilde{J}$, has the form

$$F(\bar{c}, do(A(c_1,\ldots,c_k),S_0))\equiv(\varepsilon_1\wedge\phi^+)\ \vee\ (F(\bar{c},S_0)\wedge\varepsilon_2\wedge\phi^-), \qquad (*)$$

where F is a fluent from $\mathtt{sig}\,(D'_j)$, $\bar{c}$ is a vector of constants from $\{c_1,\ldots,c_k\}$, ϕ^+, ϕ^- are sentences uniform in S_0, and each ε_1, ε_2 equals *true* or *false* (the parameters to summarize different cases of this formula). This is a definition of ground atom $F(\bar{c}, do(A(c_1,\ldots,c_k),S_0))$ via fluents at situation S_0 and situation–independent predicates and functions. Therefore, since Δ is fluent-free and for all $j\in\tilde{J}$, D'_j is uniform in S_0, every model $\mathcal{M}$ of D'_j can be transformed into a model $\mathcal{M}'$ of $\widetilde{D}_j$ which agrees with $\mathcal{M}$ on Δ. The model $\mathcal{M}'$ is obtained in two steps. First, we expand $\mathcal{M}$ with an arbitrary interpretation of function *do* and situation-independent predicates and functions from $\mathtt{sig}\,(D_i[\Omega]) \setminus \mathtt{sig}\,(D'_j)$ for every $i \in f^{-1}(j)$. Then we continue with this expanded model and modify the truth value of each fluent F at the interpretation of the tuple $\langle \bar{c}, do(A(c_1, \ldots, c_k), S_0)\rangle$ according to the obtained truth value of the formula in the definition of $F(\bar{c}, do(A(c_1, \ldots, c_k), S_0)$ above. This gives us the model $\mathcal{M}'$. Hence, if $\varphi \in \mathtt{Cons}\,(\widetilde{D}_j, \Delta)$ and $\varphi \notin \mathtt{Cons}\,(D'_j, \Delta)$, then there is a model $\mathcal{M}$ of D'_j such that $\mathcal{M} \not\models \varphi$, but then $\mathcal{M}' \models \widetilde{D}_j$ and $\mathcal{M}' \not\models \varphi$, a contradiction. Therefore, we conclude that for all $j \in \tilde{J}$, $\mathtt{Cons}\,(\widetilde{D}_j, \Delta) = \mathtt{Cons}\,(D'_j, \Delta)$ and, by pairwise Δ–inseparability of the components of $\mathcal{D}_{S_0}$, the sets from the union (†) are Δ–inseparable.

2) Since Δ is fluent-free and $\Omega(S_0)$ consists only of ground atoms with fluents, from Corollary 2 we conclude that the sets from the following union are Δ–inseparable:

$$\bigcup_{j \in \tilde{J}} \mathtt{forget}\,(\widetilde{D}_j, \Omega(S_0) \mid_j) \;\cup \bigcup_{j \in J \setminus \tilde{J}} D'_j.$$

Now we are ready to prove that the sets from $\{D''_j\}_{j \in J}$ are pairwise Δ–inseparable. For every $j \in \tilde{J}$, let us denote $G_j = \mathtt{forget}\,(\widetilde{D}_j, \Omega(S_0) \mid_j)$. We will demonstrate that for every $j \in \tilde{J}$ it holds $\mathtt{Cons}\,(G_j(S_0/S_\alpha), \Delta) = \mathtt{Cons}\,(G_j,\ \Delta)$, from which the statement follows. First, let us verify that $\mathtt{Cons}\,(G_j\ (S_0/S_\alpha), \Delta) \subseteq \mathtt{Cons}\,(G_j,\ \Delta)$. Assume that for some $j \in \tilde{J}$ (we fix this j for the following) there is a formula $\varphi \in \mathtt{Cons}\,(G_j(S_0/S_\alpha), \Delta)$ and a model $\mathcal{M}$ of G_j such that $\mathcal{M} \not\models \varphi$, and arrive at contradiction.

By the syntactic definition of forgetting a ground atom, the term S_α occurs in G_j only in subformulas obtained from the definitions $(*)$, so let us consider such a definition for a ground atom $F(\bar{c}, S_\alpha)$ with some fluent F. Let us recall that G_j is the result of forgetting a set of ground atoms with fluents having S_0 as situation argument. Since $\bar{c}$ is the vector of object arguments in the definition of $F(\bar{c}, S_\alpha)$ in $(*)$, we have $F(\bar{c}, S_0) \in \Omega(S_0) \mid_j$. Therefore, if $\mathcal{M} \models\ {}^{\varepsilon}F(\bar{c}, S_0)$ (ε denotes the possible negation in front of atom), then there is a model $\mathcal{M}' \models \neg^{\varepsilon}F(\bar{c}, S_0)$ such that $\mathcal{M}' \sim_\sigma \mathcal{M}$, with $\sigma = F(\bar{c}, S_0)$, and hence, $\mathcal{M}' \not\models \varphi$ (since Δ is fluent–free) and the truth value of $F(\bar{c}, S_\alpha)$ in $\mathcal{M}$ and $\mathcal{M}'$ is the same. Hence, either in $\mathcal{M}$ or $\mathcal{M}'$ the truth values of $F(\bar{c}, S_\alpha)$ and $F(\bar{c}, S_0)$ coincide. The similar argument applies to the whole set of definitions $(*)$ from $\widetilde{D}_j$ under forgetting the set $\Omega(S_0) \mid_j$. Therefore we may assume that in $\mathcal{M}$ or $\mathcal{M}'$, for each fluent $F \in \mathtt{sig}\,(G_j)$ the values of $F(\bar{c}, S_\alpha)$ and $F(\bar{c}, S_0)$ coincide. So $\mathcal{M} \models G_j(S_0/S_\alpha)$ or $\mathcal{M}' \models G_j(S_0/S_\alpha)$ which is a contradiction, because φ holds in neither of these models.

To prove the reverse inclusion $\mathtt{Cons}\,(G_j,\ \Delta) \subseteq \mathtt{Cons}\,(G_j(S_0/S_\alpha), \Delta)$, observe that $G_j(S_0/S_\alpha)$ is uniform in S_0. Hence, by an observation similar to Lemma 1, every model $\mathcal{M}$ of $G_j(S_0/S_\alpha)$ can be expanded to a model $\mathcal{M}'$, where the interpretation of function *do* is such that the values of terms S_α and S_0 in $\mathcal{M}'$

coincide. Then $\mathcal{M}' \models G_j$ and thus, there is no formula $\varphi \in \mathtt{Cons}\,(G_j,\ \Delta)$ such that $\varphi \notin \mathtt{Cons}\,(G_j(S_0/S_\alpha),\ \Delta)$. $\square$

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_α.

The proof of the theorem uses Proposition 8 and the component properties of forgetting from Section 3. The important observation behind this result is that in order to compute progression of an initial theory wrt an action having effects only on fluents from one decomposition component, it suffices to compute forgetting only in this component. Given a decomposition of the initial theory into inseparable components, the rest of the conditions in the theorem are purely syntactical, easy to check, and natural to hold, judging from experience of formalizing composite domains in situation calculus. SSAs can be grouped into $|I|$ components by drawing a graph with fluent names as vertices, and an edge from the fluent on the left-hand-side of each SSA going to each fluent occurring on the right-hand-side of the same SSA. Similarly, it is easy to check the last condition of the Theorem that guarantees alignment of groups of axioms in SSAs with decomposition components of $\mathcal{D}_{S_0}$.

In the above conditions, observe that if an action A occurs in active position of SSAs from two different sub-theories of $\mathcal{D}_{ss}$, then computing progression may involve forgetting in two corresponding components of $\mathcal{D}_{S_0}$ and potentially cause occurrence of common Δ_1–symbols in the components of progression. A practically important class of $\mathcal{BAT}$s for which this interference can be avoided is described in the corollary below. Note the first condition in the corollary which yields that every action mentioned in $\mathcal{BAT}$ can have effect on fluents only from one component of $\mathcal{D}_{ss}$.

Corollary 3 (Strong preservation of components in local-effect $\mathcal{BAT}$s) *For every ground action term $\alpha = A(\bar{c})$, in the conditions and notations of Theorem 2, if:*

- *no action function is in Δ_1,*
- *whenever A is in active position in an SSA for a fluent F and $F \in \mathtt{sig}\,(D'_j)$ for some $j \in J$, we have $\{c_1, \ldots, c_k\} \subseteq \mathtt{sig}\,(D'_j)$,*

then $\mathcal{D}_{S_\alpha}(S_0/S_\alpha)$ is Δ_2–decomposable into Δ_2–inseparable components.

Proof. By the first condition, action A can be in active position of SSAs of a single subtheory D_i of $\mathcal{D}_{ss}$. Then, due to the componentwise computation of progression shown in the proof of Theorem 2, progression can affect the single corresponding component $D'_{f(i)}$ of $\mathcal{D}_{S_0}$. The second condition of the corollary guarantees that $\{c_1, \ldots, c_k\} \subseteq \mathtt{sig}\,(D'_{f(i)})$. Thus, computing the progression of $\mathcal{D}_{S_0}$ wrt α is essentially a modification of $D'_{f(i)}$ which may introduce new signature symbols from context conditions of D_i only into $D'_{f(i)}$ under forgetting and into no other components of $\mathcal{D}_{S_0}$. Hence, $\mathcal{D}_{S_\alpha}(S_0/S_\alpha)$ is Δ_2–decomposable into Δ_2–inseparable components, just like $\mathcal{D}_{S_0}$ is. $\square$

We note that the corollary obviously remains true if the first condition is replaced with the simple requirement: $\Delta_1 \subseteq \Delta_2$.

Example 1 (continuation). Note that the $\mathcal{BAT}$ considered in the example satisfies the conditions of the corollary with signatures $\Delta_1 = \varnothing$ and $\Delta_2 = \{Block, S_0\}$. The theory $\mathcal{D}_{ss}$ is a union of two theories, with the intersection of signatures equal to $\{do\}$. As already noted in the example, the initial theory $\mathcal{D}_{S_0}$ is Δ_2–decomposable into Δ_2–inseparable components. Now, consider the ground action $\alpha = move(A, B, C)$. By Corollary 2 and Proposition 8, in order to compute the theory $\mathcal{D}_{S_\alpha}(S_0/S_\alpha)$ (the progression of $\mathcal{D}_{S_0}$ wrt α, with the term S_α substituted with S_0), it suffices to forget the ground atoms $On(A, B, S_0)$ and $Clear(C, S_0)$ in the first decomposition component of $\mathcal{D}_{S_0}$ and update it with the ground atoms $On(A, C, S_0)$ and $Clear(B, S_0)$. The second component of $\mathcal{D}_{S_0}$ remains unchanged. One can check that $\mathcal{D}_{S_\alpha}(S_0/S_\alpha)$ is the union of the following theories:

$$\varphi \wedge \psi \wedge (x \neq C) \rightarrow Clear(x, S_0)$$
$$\psi \rightarrow Block(x)$$
$$Block(B) \wedge Block(C) \wedge On(A,C,S_0) \wedge \neg On(A,B,S_0)$$
$$Clear(A, S_0) \wedge Clear(B, S_0) \wedge \neg Clear(C, S_0)$$

and

$$(Top(x, S_0) \vee Inheap(x, S_0)) \rightarrow \neg Block(x)$$
$$\exists x \; Block(x),$$

where φ and ψ, respectively, stand for

$$(x \neq B) \wedge \neg\exists y \; ((y \neq A \vee x \neq B) \wedge On(y, x, S_0)),$$
$$(x = A) \vee \exists y \; ((x \neq A \vee B \neq y) \wedge On(x, y, S_0)).$$

The theory $\mathcal{D}_{S_\alpha}(S_0/S_\alpha)$ is Δ_2–decomposable by the syntactic form and there is no need to compute a decomposition again after progression. Corollary 3 guarantees that the obtained components are Δ_2–inseparable and that we can compute progression for arbitrary long sequences of actions while preserving decomposability of $\mathcal{D}_{S_\alpha}(S_0/S_\alpha)$ and inseparability of its components.

5 Summary

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 σ to denote a signature or a ground atom. We slightly abuse notation and consider σ as a set of symbols even in the case of a ground atom implying that in the latter case σ 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 $\mathtt{sig}(\mathcal{T}_1) \cap \mathtt{sig}(\mathcal{T}_2) = \varDelta$, for a signature $\varDelta$.

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

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

Acknowledgements. The first author was supported by the German Research Foundation within the Transregional Collaborative Research Centre SFB/ TRR 62 "Companion-Technology for Cognitive Technical Systems", the Russian Academy of Sciences (Grant No. 15/10), and the Siberian Division of the Russian Academy of Sciences (Integration Project No. 3). Both authors would like to thank the Natural Sciences and Engineering Research Council of Canada and the Dept. of Computer Science of the Ryerson University for providing partial financial support.

References

1. Eyal Amir. (De)composition of situation calculus theories. In Henry A. Kautz and Bruce W. Porter, editors, *AAAI/IAAI*, pages 456–463. AAAI Press / The MIT Press, 2000.
2. Eyal Amir. Projection in decomposed situation calculus. In Dieter Fensel, Fausto Giunchiglia, Deborah L. McGuinness, and Mary-Anne Williams, editors, *KR*, pages 315–326. Morgan Kaufmann, 2002.
3. Stephen A. Cook and Yongmei Liu. A complete axiomatization for blocks world. *J. Log. Comput.*, 13(4):581–594, 2003.
4. William Craig. Three uses of the herbrand-gentzen theorem in relating model theory and proof theory. *J. Symb. Log.*, 22(3):269–285, 1957.
5. William Craig. The road to two theorems of logic. *Synthese*, 164(3):333–339, 2008.
6. C. Cordell Green. Application of theorem proving to problem solving. In Donald E. Walker and Lewis M. Norton, editors, *IJCAI*, pages 219–240. William Kaufmann, 1969.
7. Michael Grüninger, Torsten Hahmann, Ali Hashemi, Darren Ong, and Atalay Özgövde. Modular first-order ontologies via repositories. *Applied Ontology*, 7(2):169–209, 2012.
8. Yilan Gu and Mikhail Soutchanski. A description logic based situation calculus. *Ann. Math. Artif. Intell.*, 58(1-2):3–83, 2010.
9. Boris Konev, Carsten Lutz, Denis Ponomaryov, and Frank Wolter. Decomposing description logic ontologies. In Fangzhen Lin, Ulrike Sattler, and Miroslaw Truszczynski, editors, *KR*. AAAI Press, 2010.
10. Boris Konev, Carsten Lutz, Dirk Walther, and Frank Wolter. Formal properties of modularisation. In Heiner Stuckenschmidt, Christine Parent, and Stefano Spaccapietra, editors, *Modular Ontologies*, volume 5445 of *Lecture Notes in Computer Science*, pages 25–66. Springer, 2009.
11. George Kourousias and David Makinson. Parallel interpolation, splitting, and relevance in belief change. *J. Symb. Log.*, 72(3):994–1002, 2007.
12. Hector Levesque and Gerhard Lakemeyer. Cognitive robotics (chapter 24). In Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter, editors, *Handbook of Knowledge Representation*. Elsevier, 2007.
13. Fangzhen Lin. On strongest necessary and weakest sufficient conditions. *Artif. Intell.*, 128(1-2):143–159, 2001.
14. Fangzhen Lin and Ray Reiter. Forget it! In *Proceedings of the AAAI Fall Symposium on Relevance*, pages 154–159, 1994.
15. Fangzhen Lin and Raymond Reiter. How to progress a database. *Artificial Intelligence*, 92:131–167, 1997.

16. Yongmei Liu and Gerhard Lakemeyer. On first-order definability and computability of progression for local-effect actions and beyond. In Craig Boutilier, editor, *IJCAI*, pages 860–866, 2009.
17. Carsten Lutz, Dirk Walther, and Frank Wolter. Conservative extensions in expressive description logics. In Manuela M. Veloso, editor, *IJCAI*, pages 453–458, 2007.
18. Carsten Lutz and Frank Wolter. Mathematical logic for life science ontologies. In Hiroakira Ono, Makoto Kanazawa, and Ruy J. G. B. de Queiroz, editors, *WoLLIC*, volume 5514 of *Lecture Notes in Computer Science*, pages 37–47. Springer, 2009.
19. Carsten Lutz and Frank Wolter. Deciding inseparability and conservative extensions in the description logic el. *J. Symb. Comput.*, 45(2):194–228, 2010.
20. John McCarthy. 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, pages 410-417.
21. John McCarthy. *Formalization of common sense: papers by John McCarthy edited by V. Lifschitz*. Ablex, Norwood, N.J., 1990.
22. John McCarthy and Patrick Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence*, volume 4, pages 463–502. Edinburgh University Press, Reprinted in [21], 1969.
23. Fiora Pirri and Ray Reiter. Some contributions to the metatheory of the situation calculus. *Journal of the ACM*, 46(3):325–364, 1999.
24. Denis Ponomaryov. On decomposibility in logical calculi. *Bulletin of the Novosibirsk Computing Center*, 28:111–120, 2008.
25. Raymond Reiter. *Knowledge in Action: Logical Foundations for Describing and Implementing Dynamical Systems*. The MIT Press, 2001.
26. Stavros Vassos and Hector J. Levesque. On the progression of situation calculus basic action theories: Resolving a 10-year-old conjecture. In Dieter Fox and Carla P. Gomes, editors, *AAAI*, pages 1004–1009. AAAI Press, 2008.
27. Wael Yehia, Hongkai Liu, Marcel Lippmann, Franz Baader, and Mikhail Soutchanski. Experimental results on solving the projection problem in action formalisms based on description logics. In Yevgeny Kazakov, Domenico Lembo, and Frank Wolter, editors, *Description Logics*, volume 846 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2012.