Reasoning About Actions: Steady Versus Stabilizing State Constraints

MICHAEL THIELSCHER[1]
Dept. of Computer Science, Dresden University of Technology, 01062 Dresden (Germany)
e-mail: mit@pikas.inf.tu-dresden.de

Abstract. In formal approaches to commonsense reasoning about actions, the Ramification Problem denotes the problem of handling indirect effects which implicitly derive from so-called state constraints. We pursue a new distinction between two kinds of state constraints which will be proved crucially important for solving the general Ramification Problem. *Steady* constraints never, not even for an instant, cease being in force. As such they give rise to truly instantaneous indirect effects of actions. *Stabilizing* state constraints, on the other hand, may be suspended for a short period of time after an action has occurred. Indirect effects deriving from these constraints materialize with a short lag. This hitherto neglected distinction is shown to have essential impact on the Ramification Problem: If stabilizing state constraints interact, then approaches not based on so-called causal propagation prove defective. But causal propagation, too, is shown to risk producing anomalous models, in case steady and stabilizing indirect effects are propagated indiscriminately. Motivated by these two observations, we improve the theory of causal relationships [21] and its Fluent Calculus axiomatization, which both are methods of causal propagation, so as to properly handle the distinction between steady and stabilizing constraints.

Keywords. Temporal Reasoning, Ramification Problem, Causality.

1 Introduction

In formal systems for reasoning about actions, the Ramification Problem denotes the problem of handling indirect effects [4]. These effects are not explicitly represented in action specifications but follow from laws, *state constraints*, which formalize general dependencies among components of the world state. State constraints are static by nature; they constrain the space of world states to those which obey the laws of physics. But common sense also gains insights about dynamics, said indirect effects, from these constraints. If we learn, for example, that some light bulb is on if and only if the adjacent switch is closed (static knowledge), then we expect that light turns on and off as a side effect of toggling the switch (dynamic knowledge). This seems a straightforward conclusion, but a decade of research devoted to the Ramification Problem revealed how difficult it can be to determine the extent to which state constraints give rise to indirect effects of actions. The difficulty arises from the qualitative gap between evidential knowledge, which state constraints provide, and causal knowledge, which state constraints do not include *per se*.

An example for a basic insight along this line is that certain state constraints give rise to additional, implicit preconditions of actions rather than to indirect effects [5, 13]. This led to the distinction between *ramification* and *qualification* constraints. To which of these two categories a particular constraint belongs, is part of the domain knowledge and cannot be guessed from its

[1]On leave from Darmstadt University of Technology

mere syntactic structure. Any specification should therefore provide this information for each constraint in order to prevent wrong conclusions about indirect effects.

In this paper we pursue yet another distinction between two kinds of state constraints, which so far have received uniform treatment in literature. Namely, our concern is to separate what we call *steady* from *stabilizing* constraints. The former formalize dependencies which cannot possibly cease to hold, not even for the tiniest fraction of time. An example for this type of constraints is the fact that a physical object never occupies two distinct locations. It is impossible to bring about a situation where for an instant this is not true. Stabilizing constraints, on the other hand, generally hold in all states, too, but may be suspended for a moment immediately after the performance of an action, in which case they soon get reinstalled as the new state stabilizes (hence the name). An example is the fact that a windowless room is stuffy iff all its ventilation ducts are blocked. Here it is possible to bring about a situation where the state constraint does not hold, at least for an instant: If the room is full of fresh air and we block the last remaining free duct, then for a short period of time all ducts are blocked with the room still not being stuffy. So for a moment the constraint ceases being in force.

Steady and stabilizing state constraints have coexisted ever since the first recognition of the Ramification Problem.[2] Both kinds may give rise to indirect effects: If we move some object to a new location, then we expect that it no longer occupies the old one. This is a consequence of the aforementioned law that two distinct locations is one too many for an object. Likewise, if we block all the ventilation ducts, then we expect that as a side effect the room gets stuffy. However, there is obviously a qualitative difference between the indirect effects triggered by steady and those triggered by stabilizing state constraints. Namely, the latter materialize not without delay, tiny and imperceptible as it might be, as opposed to indirect effects deriving from steady constraints, which occur truly instantaneously. This raises the question whether the difference in nature between steady and stabilizing constraints is not relevant to the Ramification Problem.

Indeed it is crucially important, for two reasons. First, the correct modeling of stabilizing indirect effects requires so-called causal propagation, as pursued in [21, 19];[3] standard minimization-based approaches to the Ramification Problem, such as [2, 14, 12, 20, 15, 9], are insufficient, even though some of them are based on an explicit notion of causality. This follows from an observation we already made in [21] and which we repeat here because it is much better understood in the light of the distinction between steady and stabilizing effects, and also because it implies that the results in this paper matter. Consider the electric circuit depicted in Figure 1, which is an elaboration of a well-known benchmark [11] for solutions to the Ramification Problem. The six relevant components are represented by six time-dependent atomic propositions, or *fluents*, whose initial states each shall be as indicated in Figure 1 by the corresponding fluent literal. Four state constraints formalize the various physical relations among the components:

$$\begin{aligned}
&\texttt{Switch1} \wedge \texttt{Switch2} \equiv \texttt{Light} \\
&\texttt{Switch1} \wedge \texttt{Switch3} \equiv \texttt{Relay} \\
&\texttt{Relay} \supset \neg\texttt{Switch2} \\
&\texttt{Light} \supset \texttt{Detect}
\end{aligned} \tag{1}$$

All of these constraints are stabilizing and thus give rise to stabilizing indirect effects: If, for instance, `Switch1` gets closed in the state depicted, then light will turn on with a tiny time

[2] In fact, the two example constraints just mentioned were taken from [4].

[3] The term "causal propagation" is due to [19].

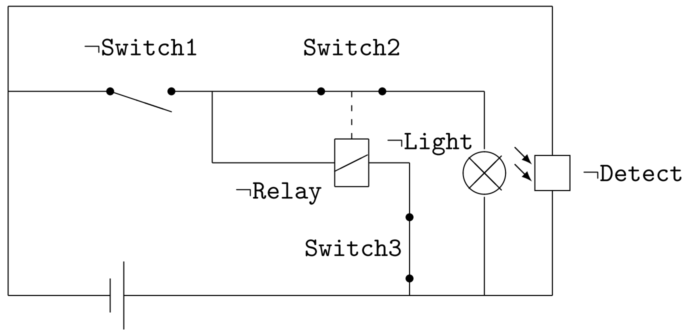

Figure 1: An electric circuit consisting of three binary switches; a light bulb; a relay, which, if activated, attracts Switch2; and a device which registers an activation of the light bulb (this device combines a phototransistor and flipflop). It is assumed that the detector stays activated forever once it was triggered. Its current state shall be ¬Detect (that is, no action of light has occurred yet).

delay. Likewise, the relay will need a short time to get activated. Once it will be active, it is going to attract Switch2, also not without a short lag. Finally, if the bulb turns on, then for an instant the detector will still be off before it registers the light.

Now, suppose we close Switch1 in the state shown in Figure 1. What is the expected outcome? Obviously, the relay gets activated and, thus, attracts Switch2. Hence, the latter is open in the finally resulting state. Notice, however, that as soon as the first switch is closed, the sub-circuit involving the light bulb gets closed, too. This may activate the light bulb for an instant, that is, before Switch2 jumps its position as a consequence of activating the relay. If this is indeed the case, then this short-time activation might be registered by the photo device, in which case the latter would be activated forever. Hence, while it is clear that the relay is activated, Switch2 is open, and the light bulb is off in the resulting state, it may or may not be the case that Detect becomes true. Therefore our circuit may end up in either of two possible resulting states, viz.

(a) $\mathtt{Switch1} \wedge \neg\mathtt{Switch2} \wedge \mathtt{Switch3} \wedge \mathtt{Relay} \wedge \neg\mathtt{Light} \wedge \neg\mathtt{Detect}$

(b) $\mathtt{Switch1} \wedge \neg\mathtt{Switch2} \wedge \mathtt{Switch3} \wedge \mathtt{Relay} \wedge \neg\mathtt{Light} \wedge \mathtt{Detect}$

The circuit thus exhibits a non-deterministic behavior. In particular, no conclusion can be made concerning the resulting truth-value of fluent Detect. Notice that the first one of the possible successor states is strictly closer to the initial state of the circuit than the second one (that is, the first one can be obtained by a strict subset of fluent changes). The significance of this observation lies in the fact that non-propagation approaches to the Ramification Problem, e.g., [2, 14, 12, 20, 15, 9], entail the overly credulous conclusion that ¬Detect holds after closing Switch1. The reason is that in all of the cited methods all obtained indirect effects need to be justified, on the basis of an action's direct effects, wrt. either the initial or the final state. But fluent Detect possibly becoming true cannot be gathered from the initial state (in which Light is false) nor from the overall resulting state (in which Light is false again). Rather the non-minimal possible successor state, (b), is obtained by a sequence of (stabilizing) effects $\dots\varepsilon_i\dots\varepsilon_j\dots\varepsilon_k\dots$ in which ε_j 'exploits' the temporary violation of some constraint after ε_i

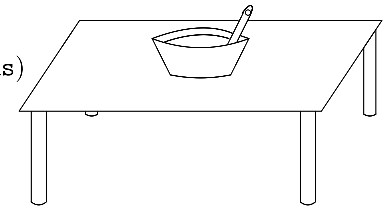

Figure 2: A bowl filled with soup is standing on a table. The soup spills out and produces a stain if the table is lifted on one hand side but not the other. Nothing of this sort is expected when lifting up the table on both sides simultaneously.

and prior to ε_k.[4] The aforementioned approaches ignore the possibility of ε_j being 'inserted,' and so entail erroneous conclusions to the effect that $\varepsilon_j = \texttt{Detect}$ cannot possibly occur.

Our example scenario reveals a general deficiency of most existing approaches to the Ramification Problem when it comes to modeling stabilizing indirect effects: During the process of stabilizing, many interesting situations may temporarily arise, which all are necessarily missed if the mere initial and finally resulting states are used for reference. One might argue that the light detector coming on in our example would be an effect without cause since neither the initial nor the final state contains a justification for this. However, it lies in the nature of stabilizing constraints that they can be momentarily violated in actual states—states which are unstable but may occur in reality for a very short period of time. Thus an effect is well justified which materializes in the course of state stabilization due to the temporary appearance of a cause. Getting our example right, and in general handling nested stabilizing effects appropriately, therefore requires to somehow compute ramifications step-by-step by establishing intermediate states.[5] This is the paradigm of *causal propagation* [21, 19], by which ramifications are computed separately, one after another, and so causal chains are suitably accounted for. In this way it is possible to formally mimic the above reasoning process that led to the two possible resulting successor states [21].

While existing methods of causal propagation thus handle interacting stabilizing state constraints correctly, the qualitative difference between steady and stabilizing indirect effects raises a new challenge for these approaches. The following simple scenario shows that anomalous models may be produced if our distinction between the two kinds of state constraints is not respected: Suppose a bowl well filled with soup is standing on a rectangular table; see Figure 2. Whenever the left hand side of the table is lifted up but not the right hand side (or vice versa), then the spilling soup stains the tablecloth. If, on the other hand, both sides are lifted up simultaneously, then the soup stays in the bowl.[6] Suppose given the following five fluents to describe the various states in this domain:

$$\texttt{OnTheGround(Lhs)},\ \texttt{OnTheGround(Rhs)},\ \texttt{InTheAir(Lhs)},\ \texttt{InTheAir(Rhs)},\ \texttt{Stain}$$

[4] The three effect ε_i, ε_j, and ε_k are, respectively, light turns on, detector is activated, light turns off again.

[5] A radical alternative consequence that could be drawn from our observation is to ban stabilizing state constraints from the Ramification Problem altogether. This, however, seems not to be in accordance with many existing works on this research topic. Here is a (surely incomplete) list of related publications in which at least one constraint occurs which had to be categorized as stabilizing: [4, 2, 13, 6, 9].

[6] This scenario originates in an example from [17]. In the following we use a novel version thereof, for the sake of argument.

Propositions $\texttt{OnTheGround}(x)$ and $\texttt{InTheAir}(x)$, respectively, shall be true if side x of the table is currently down (resp. up), and $\texttt{Stain}$ shall be true if the tablecloth is currently stained. For the sake of simplicity, we assume that at any time each side of the table is either on the ground or in the air. We then have these two state constraints:

$$\forall x.\ \texttt{OnTheGround}(x) \equiv \neg\texttt{InTheAir}(x) \tag{2}$$

$$(\texttt{InTheAir}(\texttt{Lhs}) \not\equiv \texttt{InTheAir}(\texttt{Rhs})) \supset \texttt{Stain} \tag{3}$$

The second one says that there is a stain whenever one but not the opposite side of the table is in the air. To which of our two categories do these constraints belong? The first one is steady, for it is impossible, even for an instant, that a side of the table is both on the ground and in the air. The second one is stabilizing, for the stain is produced only after a short delay once the two sides of the table are brought into different positions.

Both our two state constraints give rise to a number of indirect effects. Fluents $\texttt{OnTheGround}$ and $\texttt{InTheAir}$ being tightly coupled, whenever an action occurs that changes the truth value of an instance of one of them, then the respective instance of the other one changes accordingly as a side effect. Our second constraint, if read causally, says that any action additionally causes a stain in the tablecloth if it first causes the left hand side of the implication becoming true. Throughout this paper, we will formalize such causal knowledge of indirect effects by so-called *causal relationships* as defined in [21]. Their general format is ε <u>causes</u> ϱ <u>if</u> Φ, which should be read as: If context Φ holds after the occurrence of a direct or indirect effect ε, then the additional indirect effect ϱ is caused.[7] The various causal relationships that hold in our example thus are the following:[8]

$$\begin{array}{rlr}
\neg\texttt{OnTheGround}(x) & \underline{\text{causes}} & \texttt{InTheAir}(x)\ \underline{\text{if}}\ \top \\
\texttt{OnTheGround}(x) & \underline{\text{causes}} & \neg\texttt{InTheAir}(x)\ \underline{\text{if}}\ \top \\
\texttt{InTheAir}(x) & \underline{\text{causes}} & \neg\texttt{OnTheGround}(x)\ \underline{\text{if}}\ \top \\
\neg\texttt{InTheAir}(x) & \underline{\text{causes}} & \texttt{OnTheGround}(x)\ \underline{\text{if}}\ \top \\
\texttt{InTheAir}(\texttt{Lhs}) & \underline{\text{causes}} & \texttt{Stain}\ \underline{\text{if}}\ \neg\texttt{InTheAir}(\texttt{Rhs}) \\
\texttt{InTheAir}(\texttt{Rhs}) & \underline{\text{causes}} & \texttt{Stain}\ \underline{\text{if}}\ \neg\texttt{InTheAir}(\texttt{Lhs}) \\
\neg\texttt{InTheAir}(\texttt{Rhs}) & \underline{\text{causes}} & \texttt{Stain}\ \underline{\text{if}}\ \texttt{InTheAir}(\texttt{Lhs}) \\
\neg\texttt{InTheAir}(\texttt{Lhs}) & \underline{\text{causes}} & \texttt{Stain}\ \underline{\text{if}}\ \texttt{InTheAir}(\texttt{Rhs})
\end{array} \tag{4}$$

Relationship $\texttt{InTheAir}(\texttt{Lhs})$ <u>causes</u> $\texttt{Stain}$ <u>if</u> $\neg\texttt{InTheAir}(\texttt{Rhs})$, for instance, indicates that any action with effect $\texttt{InTheAir}(\texttt{Lhs})$ while $\neg\texttt{InTheAir}(\texttt{Rhs})$ holds, has $\texttt{Stain}$ as an indirect effect. Notice that the first four causal relationships describe indirect effects which occur without lag since they derive from a steady constraint. Deriving from a stabilizing constraint, the bottom four relationships each describe an indirect effect with a short delay.

Now, suppose the current state be that the table is standing firmly on the floor and there is

[7] Notice the distinction between triggering effect ε, which is a fluent literal that must have become true in order for the causal relationship to 'fire,' and context Φ, which merely has to hold, regardless of whether it just came about or was true all the time. Dividing the condition for the occurrence of an indirect effect into two components matches the distinction often made in philosophical accounts of causality between so-called "triggering" and "predisposal" causes.

[8] Below, $\top$ denotes a tautology.

no stain, i.e.,

$$\texttt{OnTheGround(Lhs)} \wedge \neg\texttt{InTheAir(Lhs)} \wedge \texttt{OnTheGround(Rhs)} \wedge \neg\texttt{InTheAir(Rhs)} \\ \wedge \neg\texttt{Stain}$$

as depicted in Figure 2. Suppose further that we lift up both the left hand side and the right hand side of the table simultaneously so that afterwards neither of the two sides is down any longer. This action can thus be characterized by the two direct effects $\neg\texttt{OnTheGround(Lhs)}$ and $\neg\texttt{OnTheGround(Rhs)}$. So the preliminary result of our action, where no indirect effects have yet been generated, is

$$\neg\texttt{OnTheGround(Lhs)} \wedge \neg\texttt{InTheAir(Lhs)} \wedge \neg\texttt{OnTheGround(Rhs)} \wedge \neg\texttt{InTheAir(Rhs)} \\ \wedge \neg\texttt{Stain}$$

Proceeding with adjusting according to possible indirect effects of our action, we see that $\texttt{InTheAir(Lhs)}$ should become true as a side effect of $\neg\texttt{OnTheGround(Lhs)}$. Formally this follows from the topmost causal relationship of (4). Accommodating this effect results in

$$\neg\texttt{OnTheGround(Lhs)} \wedge \texttt{InTheAir(Lhs)} \wedge \neg\texttt{OnTheGround(Rhs)} \wedge \neg\texttt{InTheAir(Rhs)} \\ \wedge \neg\texttt{Stain}$$

As for the next step, the most natural thing to do would be to likewise change $\neg\texttt{InTheAir(Rhs)}$ to $\texttt{InTheAir(Rhs)}$ as a side effect of $\neg\texttt{OnTheGround(Rhs)}$. The result would be that the two sides of the table are both in the air, no longer on the ground, and that no stain has been produced—the only reasonable conclusion in this scenario. But if we take a look at our causal relationships, then we see that we could 'squeeze in' the indirect effect that the tablecloth gets stained! This is so because all conditions are satisfied for the application of the fifth one of our causal relationships in (4): $\texttt{InTheAir(Lhs)}$ occurred as (indirect) effect while $\neg\texttt{InTheAir(Rhs)}$ still holds. So doing we obtain

$$\neg\texttt{OnTheGround(Lhs)} \wedge \texttt{InTheAir(Lhs)} \wedge \neg\texttt{OnTheGround(Rhs)} \wedge \neg\texttt{InTheAir(Rhs)} \\ \wedge \texttt{Stain}$$

If afterwards we resolve the conflict still present of having both $\neg\texttt{OnTheGround(Rhs)}$ and $\neg\texttt{InTheAir(Rhs)}$, then the final result is a state which satisfies

$$\neg\texttt{OnTheGround(Lhs)} \wedge \texttt{InTheAir(Lhs)} \wedge \neg\texttt{OnTheGround(Rhs)} \wedge \texttt{InTheAir(Rhs)} \\ \wedge \texttt{Stain}$$

We have thus found a chain of deductions which comes to the unexpected conclusion that the tablecloth become stained.

Recalling the discussion at the beginning, it is quite obvious what is responsible for the undesired conclusion. Our mistake was to mix indirect effects triggered by steady state constraints with those triggered by stabilizing ones. In particular we should not have generated the effect $\texttt{Stain}$, which occurs only after a short delay, *before* accounting for the instantaneous effect $\texttt{InTheAir(Rhs)}$.

This example, and in particular the undesired conclusion, shows that it might be vital to know the category, steady or stabilizing, a state constraint belongs to. In the following, we illustrate exemplarily, on the basis of a concrete approach to the Ramification Problem that uses causal propagation, how to exploit this information in order to avoid erroneous conclusions like the above. Basically, what needs to be guaranteed is that never *any* indirect effect that occurs with a short lag is generated until *all* effects deriving from steady state constraints have been accounted for. In the next section, we introduce the distinction between steady and stabilizing constraints into the formal theory of causal relationships as described in [21]. Afterwards we present a correspondingly elaborated strategy for axiomatizing action domains with ramifications by means of the Fluent Calculus. In the concluding discussion, we contrast ramifications with delayed effects, a topic which naturally arises when considering stabilizing state constraints.

2 Steady vs. Stabilizing Causal Relations

The theory of causal relationship has been developed to address the Ramification Problem in a causality-oriented way. In the following we integrate the distinction between steady and stabilizing state constraints. This distinction passes on to the various causal relationships, and we extend the existing theory so as to suitably reflect this distinction.

A basic ingredient of the theory is the concept of a *fluent*, which describes time-dependent properties, sometimes of *entities*; e.g., `Stain` or `OnTheGround(Lhs)` etc. A *ground fluent literal* is a fluent or its negation. We say that a set of ground fluent literals is *inconsistent* if it contains a fluent along with its negation. A *state* is a maximal consistent set of ground fluent literals. The elements of an underlying set of fluents can be considered atoms for constructing formulas using the standard connectives of classical first-order logic, including quantifiers, where the variables range over the underlying set of entities. The notion of fluent formulas being *true* in a state S is based on defining a ground fluent literal L to be true if and only if $L \in S$. E.g., the two fluent formulas (2) and (3) are true in the state depicted in Figure 2 but false in, say, the state $\{\texttt{OnTheGround(Lhs)}, \neg\texttt{InTheAir(Lhs)}, \texttt{OnTheGround(Rhs)}, \texttt{InTheAir(Rhs)}, \neg\texttt{Stain}\}$. *State constraints* are fluent formulas which constrain the set of all formally possible states.

The second fundamental notion is that of an *action*. Actions cause state transitions. Since the focus of the paper is on indirect effects, we consider a basic, STRIPS style [3] way of specifying the direct effect of an action, namely, by saying which fluents change their truth-value when the action is being performed. *Action laws* serve this purpose: They are of the form $a(\vec{x})$ $\underline{\text{transforms}}$ C $\underline{\text{into}}$ E where

- $\vec{x}$ is a (possibly empty) sequence of pairwise distinct variables;
- a is an action name of arity equal to the length of $\vec{x}$;
- C (the *condition*) and E (the *effect*) are sets of fluent literals (possibly with variables chosen from $\vec{x}$);
- for any sequence of entities $\vec{e}$ of the same length as $\vec{x}$, both $C\{\vec{x} \mapsto \vec{e}\}$ and $E\{\vec{x} \mapsto \vec{e}\}$ contain the same fluents (but usually with different polarity).[9]

An example for an action law is

$$\begin{aligned} \texttt{LiftBoth}(x,y)\ \underline{\text{transforms}}\ & \{\texttt{OnTheGround}(x), \texttt{OnTheGround}(y)\} \\ \underline{\text{into}}\ & \{\neg\texttt{OnTheGround}(x), \neg\texttt{OnTheGround}(y)\} \end{aligned} \tag{5}$$

[9]By $\{\vec{x} \mapsto \vec{e}\}$ we mean the simultaneous replacement of each variable in $\vec{x}$ by the respective entity in $\vec{e}$.

If S is a state, then an instance $\alpha\{\vec{x} \mapsto \vec{e}\}$ of an action law $\alpha = a(\vec{x})$ transforms C into E is *applicable* to S iff $C\{\vec{x} \mapsto \vec{e}\} \subseteq S$; the *application* of this instance to S yields the state $(S \setminus C\{\vec{x} \mapsto \vec{e}\}) \cup E\{\vec{x} \mapsto \vec{e}\}$.[10]

States resulting from the application of an action law, which concentrates on the direct effects, may violate the underlying state constraints. If, for instance, we apply the aforementioned law for $\texttt{LiftBoth}(\texttt{Lhs},\texttt{Rhs})$ to the state depicted in Figure 2, then our constraint $\forall x.\, \texttt{OnTheGround}(x) \equiv \neg\texttt{InTheAir}(x)$ no longer holds. This calls for the additional generation of indirect effects. Each single indirect effect is obtained according to a *causal relationship*, which is of the form ε causes ϱ if Φ where Φ is a fluent formula and both ε and ϱ are fluent literals (possibly containing variables). The process of generating indirect effects is initialized with the state resulting from the direct effects of an action. Additional, indirect effects are then computed by (non-deterministically) selecting and (serially) applying causal relationships, until eventually a state obtains which satisfies all state constraints. In this way indirect effects are *causally propagated*, in the terminology of [19]. Notice that some of the 'intermediate' states may violate one or more steady constraints and thus do not necessarily correspond to states which are possible in reality.

Formally, causal relationships manipulate state-effect pairs (S, E). State S is an intermediate result where some but not yet all indirect effects have been accounted for, and E contains all direct and indirect effects computed so far. We define an instance $r\{\vec{x} \mapsto \vec{e}\}$ of a causal relationship $r = \varepsilon$ causes ϱ if Φ (with free variables $\vec{x}$) *applicable* to (S, E) iff $\varepsilon\{\vec{x} \mapsto \vec{e}\} \in E$ and $\Phi\{\vec{x} \mapsto \vec{e}\} \wedge \neg\varrho\{\vec{x} \mapsto \vec{e}\}$ is true in S. The *application* of this instance to (S, E) yields the pair (S', E') where $S' = (S \setminus \{\neg\varrho\{\vec{x} \mapsto \vec{e}\}\}) \cup \{\varrho\{\vec{x} \mapsto \vec{e}\}\}$ and $E' = (E \setminus \{\neg\varrho\{\vec{x} \mapsto \vec{e}\}\}) \cup \{\varrho\{\vec{x} \mapsto \vec{e}\}\}$. Put in words, a causal relationship is applicable to an intermediate state if the associated context Φ holds in that state, if the particular indirect effect ϱ is currently false, and if the cause ε is among the current effects. As the result of the application the indirect effect ϱ becomes true in S and is added to E. If $\mathcal{R}$ is a set of causal relationships, then by $(S, E) \leadsto_{\mathcal{R}} (S', E')$ we denote the existence of an element in $\mathcal{R}$ whose application to (S, E) yields (S', E'). We adopt a standard notation in writing $(S, E) \stackrel{*}{\leadsto}_{\mathcal{R}} (S', E')$ to indicate that there is a (possibly empty) sequence of causal relationships in $\mathcal{R}$ whose successive application to (S, E) yields (S', E').

We have somewhat loosely said that indirect effects follow from state constraints. Having the formal definition of causal relationships, this correspondence can be stated more precisely. A causal relationship ε causes ϱ if Φ *originates* in some state constraint if the latter implies $\Phi \wedge \varepsilon \supset \varrho$. However, fundamental to the Ramification Problem is the fact that an implication which is a purely logical consequence of a state constraint does not necessarily give rise to an indirect effects. Causal relationships thus contain more information than the mere state constraints. Yet it is not necessary to draw up the valid causal relationships all by hand. They can rather be generated automatically given additional domain-specific knowledge—called *influence information*—of how fluents may generally affect each other (see [21] for details).[11]

State constraints are either steady or stabilizing. To which category a constraint belongs is

[10] The definition allows two or more simultaneously applicable laws for one and the same action, so that non-deterministic actions can be specified.

[11] For example, the causal relationships (4) can be automatically obtained from the underlying state constraints (2) and (3), respectively, on the basis of the influence information that `OnTheGround` may affect `InTheAir` and vice versa, and that `InTheAir` may affect `Stain`. A critical property of the method described in [21] is that it may yield different sets of causal relationships for semantically equivalent state constraints. Following a suggestion by Javier Pinto, independence of syntax is achieved by processing the prime implicants of a set of state constraints.

domain knowledge and so needs to be part of the specification. The criterion for characterizing a state constraint as steady is that not even for an instant a situation is imaginable where this constraint is violated. This information passes over to the corresponding causal relationships. To summarize, a domain specification consists of

- sets of entities and fluents,
- sets of actions and action laws,
- sets of steady and stabilizing state constraints, and
- sets of steady and stabilizing causal relationships.

The scenario discussed in the second part of the introduction taught us that during the application of causal relationships the insertion of an effect with real delay in between the generation of steady indirect effects needs to be prohibited. This we can achieve by first applying only causal relationships stemming from steady state constraints, until none of these constraints is violated any longer. Only thereafter a stabilizing effect may be generated, again followed by accounting for all steady effects necessary to satisfy the steady constraints, and so on until an overall acceptable state obtains. This strategy is formalized in the following definition of a successor state.

Definition 1 Let $\mathcal{C}_{std}$ and $\mathcal{C}_{stb}$ be sets of steady and stabilizing, respectively, state constraints, and $\mathcal{R}_{std}$ and $\mathcal{R}_{stb}$ be sets of steady and stabilizing, respectively, causal relationships. If a is an action and S a state in which all elements of $\mathcal{C}_{std} \cup \mathcal{C}_{stb}$ are true, then a state S' is a *successor* of S and a iff the following holds: There is an applicable action law instance a <u>transforms</u> C <u>into</u> E and there exist states $S_0, S'_0, \ldots, S_n, S'_n$ and sets of fluent literals $E_0, E'_0, \ldots, E_n, E'_n$ $(n \geq 0)$ such that $S_0 = (S \setminus C) \cup E$, $E_0 = E$,

$$
\begin{aligned}
(S_0, E_0) \overset{*}{\leadsto}_{\mathcal{R}_{std}} & (S'_0, E'_0) \\
& \leadsto_{\mathcal{R}_{stb}} (S_1, E_1) \overset{*}{\leadsto}_{\mathcal{R}_{std}} (S'_1, E'_1) \\
& \ldots \\
& \leadsto_{\mathcal{R}_{stb}} (S_n, E_n) \overset{*}{\leadsto}_{\mathcal{R}_{std}} (S'_n, E'_n)
\end{aligned}
$$

and, for each $0 \leq i \leq n$, all elements of $\mathcal{C}_{std}$ are true in S'_i, and in S'_n also all elements of $\mathcal{C}_{stb}$ are true. ∎

The following proposition shows that by integrating the distinction between steady and stabilizing state constraints and by its formal treatment according to Definition 1, we have solved the problem of undesired 'squeezing-ins' of indirect effects that occur with a lag as discussed in the introduction.

Proposition 2 *Consider the domain specification consisting of*

- *the entities* `Lhs` *and* `Rhs`*, the unary fluent names* `OnTheGround` *and* `InTheAir`*, and the nullary fluent name* `Stain`*;*
- *the binary action* `LiftBoth` *in conjunction with action law (5);*
- *steady state constraint (2) and stabilizing state constraint (3); and*

- *the causal relationships of (4), of which the top four are steady while the others are stabilizing.*

Then $\neg\texttt{Stain}$ *holds in the unique successor of*

$$S = \{\texttt{OnTheGround(Lhs)}, \neg\texttt{InTheAir(Lhs)}, \texttt{OnTheGround(Rhs)}, \neg\texttt{InTheAir(Rhs)}, \neg\texttt{Stain}\}$$

and $a = \texttt{LiftBoth(Lhs,Rhs)}$.

> **Proof:** Let $\mathcal{C}_{std} = \{(2)\}$, $\mathcal{C}_{stb} = \{(3)\}$, and let $\mathcal{R}_{std}$ and $\mathcal{R}_{stb}$ consist of the first and second half, respectively, of list (4). We first note that state S satisfies $\mathcal{C}_{std} \cup \mathcal{C}_{stb}$ and that the only applicable action law instance has the effect $\{\neg\texttt{OnTheGround(Lhs)}, \neg\texttt{OnTheGround(Rhs)}\}$. So there is a unique pair (S_0, E_0) to start off, viz.
>
> $$\begin{aligned} S_0 &= \{\neg\texttt{OnTheGround(Lhs)}, \neg\texttt{InTheAir(Lhs)}, \\ &\quad\; \neg\texttt{OnTheGround(Rhs)}, \neg\texttt{InTheAir(Rhs)}, \neg\texttt{Stain}\}, \\ E_0 &= \{\neg\texttt{OnTheGround(Lhs)}, \neg\texttt{OnTheGround(Rhs)}\} \end{aligned}$$
>
> The first component violates $\mathcal{C}_{std}$. Two instances of elements of $\mathcal{R}_{std}$ are applicable, namely,
>
> $$\begin{array}{l} \neg\texttt{OnTheGround(Lhs)}\ \ \underline{\text{causes}}\ \ \texttt{InTheAir(Lhs)}\ \ \underline{\text{if}}\ \ \top \\ \neg\texttt{OnTheGround(Rhs)}\ \ \underline{\text{causes}}\ \ \texttt{InTheAir(Rhs)}\ \ \underline{\text{if}}\ \ \top \end{array}$$
>
> If either of them is applied to (S_0, E_0), then the state component of the resulting pair still does not satisfy $\forall x. \texttt{OnTheGround}(x) \equiv \neg\texttt{InTheAir}(x)$. The other one of the two causal relationships, however, remains applicable—and is then the only one among those which are steady. We can thus find a unique (S_0', E_0') such that $(S_0, E_0) \overset{*}{\leadsto}_{\mathcal{R}_{std}} (S_0', E_0')$, viz.
>
> $$\begin{aligned} S_0' &= \{\neg\texttt{OnTheGround(Lhs)}, \texttt{InTheAir(Lhs)}, \\ &\quad\; \neg\texttt{OnTheGround(Rhs)}, \texttt{InTheAir(Rhs)}, \neg\texttt{Stain}\}, \\ E_0' &= \{\neg\texttt{OnTheGround(Lhs)}, \neg\texttt{OnTheGround(Rhs)}, \\ &\quad\; \texttt{InTheAir(Lhs)}, \texttt{InTheAir(Rhs)}\} \end{aligned}$$
>
> All formulas in $\mathcal{C}_{std} \cup \mathcal{C}_{stb}$ are true in state S_0', hence the latter is a successor state of S and a—in which $\neg\texttt{Stain}$ holds. Moreover, no further causal relationships, be they steady or stabilizing, are applicable to (S_0', E_0'), which is why S_0' is the unique successor.

■

3 Steady vs. Stabilizing Ramifications in the Fluent Calculus

We proceed with adapting the axiomatization strategy for action domains with ramifications of [21], which is based on the Fluent Calculus, so as to cope with the distinction between steady and stabilizing ramifications. As opposed to the Situation Calculus [16], the Fluent

Calculus [7, 1] employs structured state terms which each consists in a collection of the fluent literals that are true in the state being represented. To this end, fluent literals are reified, i.e., formally represented as terms. The initial state of our example scenario, for instance, could be represented by the term $\mathtt{OnTheGround}(\mathtt{Lhs}) \circ \neg\mathtt{InTheAir}(\mathtt{Lhs}) \circ \mathtt{OnTheGround}(\mathtt{Rhs}) \circ \neg\mathtt{InTheAir}(\mathtt{Rhs}) \circ \neg\mathtt{Stain}$, where the negation sign denotes a special unary function and $\circ$ a special binary function which obeys the laws of associativity and commutativity. It has first been argued in [7] that this representation technique, which appeals exclusively to classical, i.e., monotonic logic, avoids extra axioms to encode the general commonsense law of persistence. The effects of actions are modeled by manipulating state terms through removal and addition of sub-terms. Then all sub-terms which are not affected by these operations automatically remain in the state term, hence continue to being true. In [22] we have presented a novel version of the Fluent Calculus as the result of gradually improving the concept of successor state axioms [18] in view of the inferential aspect of the Frame Problem [16] but without losing its representational merits.

In the following, we concentrate on the part of the axiomatization strategy of [21] which requires refinement in order to cope with the subject of the present paper. The original axiomatization uses three predicates called, respectively, *Possible*, *Causes*, and *Ramify*. Their definitions need to be extended or modified.

An instance $\mathit{Possible}(s)$ is defined to being true iff s is a term which represents a state that satisfies all underlying constraints. To this we add an identical definition of a predicate $\mathit{Possible}_{std}(s)$, which shall be true iff its argument satisfies just all steady constraints. Informally, then, a state which is not $\mathit{Possible}_{std}$ is truly impossible, while a state which is $\mathit{Possible}_{std}$ but not *Possible* may occur for an instant but not as a final, stable result of an action. Clearly, a correct axiomatization always entails $\forall s.\, \mathit{Possible}(s) \supset \mathit{Possible}_{std}(s)$.

Predicate $\mathit{Causes}(s, e, s', e')$ has been defined as the existence of a causal relationship that maps pair (s, e) into pair (s', e') (or rather the states and sets of effects being represented by s, e, s', e'). This we replace by identical definitions of two predicates named Causes_{std} and Causes_{stb}, in order to distinguish between steady and stabilizing causal relationships.

The only less straightforward modification concerns the predicate *Ramify*, which models the repeated application of causal relationships, i.e., the process of causal propagation. More precisely, $\mathit{Ramify}(s, e, s')$ has been, and shall still be, defined true iff s' is a successor state which can be obtained from the initial state-effect pair (s, e). To reflect the interim stages in the new ramification procedure pursued in this paper, we introduce a second predicate named $\mathit{Ramify}_{std}(s, e, s', e')$, a valid instance of which shall indicate the existence of a (possibly empty) sequence of steady causal relationships whose application to the state-effect pair (s, e) yields (s', e') such that $\mathit{Possible}_{std}(s')$ holds. In essence the definitions of *Ramify* and Ramify_{std} are reflexive, transitive closures. As this mathematical concept cannot be expressed in first-order logic, we employ the standard way of encoding closure by means of second-order formulas,[12] just

[12] See, for example, Section 2 in [10].

like in [21]:

$$Ramify_{std}(s,e,s',e') \;\equiv\; Possible_{std}(s') \wedge \forall\Pi \left\{ \begin{array}{c} \forall s_1,e_1.\ \Pi(s_1,e_1,s_1,e_1) \\ \wedge \\ \left[\begin{array}{l} \forall s_1,e_1,s_2,e_2,s_3,e_3. \\ \quad \Pi(s_1,e_1,s_2,e_2) \wedge Causes_{std}(s_2,e_2,s_3,e_3) \\ \qquad\qquad \supset \Pi(s_1,e_1,s_3,e_3) \end{array} \right] \\ \supset \\ \Pi(s,e,s',e') \end{array} \right\} \tag{6}$$

That is, $Ramify_{std}(s,e,s',e')$ is true iff (s,e,s',e') belongs to the reflexive and transitive closure of $Causes_{std}$ and if s' satisfies the underlying steady state constraints.

Analogously, we define

$$Ramify(s,e,s') \;\equiv\; Possible(s') \wedge \forall\Pi \left\{ \begin{array}{c} \forall s_1,e_1,s_2,e_2.\ Ramify_{std}(s_1,e_1,s_2,e_2) \supset \Pi(s_1,e_1,s_2,e_2) \\ \wedge \\ \left[\begin{array}{l} \forall s_1,e_1,s_2,e_2,s_3,e_3,s_4,e_4. \\ {[}\,\Pi(s_1,e_1,s_2,e_2) \wedge Causes_{stb}(s_2,e_2,s_3,e_3) \\ \quad \wedge Ramify_{std}(s_3,e_3,s_4,e_4)\,] \supset \Pi(s_1,e_1,s_4,e_4) \end{array} \right] \\ \supset \\ \exists e'.\ \Pi(s,e,s',e') \end{array} \right\} \tag{7}$$

That is, an instance $Ramify(s,e,s')$ holds iff there exists some e' such that (s,e,s',e') belongs to the reflexive and transitive closure of joining $Ramify_{std}$ to $Causes_{stb}$ and if s' satisfies the entire state constraints.

This completes the improvement of the original axiomatization needed to reflect the new distinction between steady and stabilizing ramifications. Correctness of the resulting encoding wrt. the refined concept of successor state as given in Definition 1 follows from the relative correctness of our definitions of $Ramify_{std}$ and $Ramify$ (for a proof see the appendix) and from the main correctness result given in [21].

4 Summary and Discussion

We have shown how ignoring the distinction between steady and stabilizing state constraints can lead to anomalous models if indirect effects are accommodated by causal propagation. While our key scenario would be treated correctly by existing non-propagation approaches to the Ramification Problem, these are defective if just stabilizing constraints interact. In summary, as soon as a domain specification includes stabilizing state constraints which give rise to indirect effects, then the latter need to be accounted for by causal propagation, and in so doing one had better take into account the different nature of the two types of constraints. We have accordingly refined both the theory of causal relationships and a suitable Fluent Calculus axiomatization so as to properly deal with this distinction.

The motivation for distinguishing steady and stabilizing indirect effects is the observation that it might be overly credulous to consider possible any order in which additional, indirect effects of actions are generated. Not always are all computed chains of indirect effects equally likely to happen in reality. The lag between some particular indirect effect and its triggering cause may generally be shorter than between another particular effect and its cause. An approach to this problem different to the one taken in this paper, is to introduce an explicit notion of time, namely, in specifying the exact delay between the occurrence of an effect and its cause. This would make an indirect effect a so-called *delayed* effect. Yet by introducing explicit time one lowers the level of abstraction, which is not necessary, and hence unwanted, in many instances of commonsense reasoning about actions and change. Even worse, if precise knowledge as to the delays of certain effects is just not available, then one needs to introduce some symbolic delay and, more troublesome, to disallow the occurrence of intervening actions or events. The Ramification Problem calls for performing qualitative reasoning about indirect effects, as opposed to quantitative reasoning, which would require precise knowledge of virtually indistinguishable time intervals. Qualitative reasoning, which acknowledges the fact that common sense often lacks precise knowledge, considers equal all temporal delays between cause and indirect effect—with the exception that real delays do need to be distinguished from those which are zero, as we have argued in this paper.

Acknowledgments. The author wants to thank the two anonymous reviewers for their suggestions and comments, and the participants of the CommonSense'98 Workshop for the discussion on this paper, which helped improving it in several aspects.

A Proof for Section 3

We prove that the elaboration described in Section 3 of the axiomatization technique of [21] is correct wrt. the elaboration of the theory of causal relationships as given by Definition 1.

Fluent Calculus encodings are characterized by their using a binary function " $\circ$ " which connects reified fluent literals. Below we employ a mapping, denoted by τ, which assigns to a set of fluent literals $S = \{L_1, \ldots, L_n\}$ a certain term, the so-called *collection* $\tau_S = L_1 \circ \cdots \circ L_n$. (This mapping includes the special case $\tau_{\{\}} = \emptyset$.) Using the function $\circ$ to reify conjunctions of fluents requires a set of foundational axioms, namely,

- three equality axioms (abbreviated AC1) formalizing the laws of associativity, commutativity, and unit element (denoted by $\emptyset$), that is,

$$\begin{array}{lrcl} \forall x, y, z. & (x \circ y) \circ z & = & x \circ (y \circ z) \\ \forall x, y. & x \circ y & = & y \circ x \\ \forall x. & x \circ \emptyset & = & x \end{array}$$

- the standard equality axioms (i.e., reflexivity, symmetry, transitivity, and substitutivity);
- equational formulas making the whole theory AC1-unification complete (see [8]).

This theory we abbreviate by EUNA, which stands for *extended unique name assumption*. This emphasizes the fact that in being AC1-unification complete, this theory generalizes the standard unique name assumption so that two collections are provably unequal whenever they are built up from different fluent literals.

The foregoing preparatory remarks are needed to formally express what it means for our axioms which define Ramify_{std} and Ramify, to be correct. We assume given correct axiomatizations of state constraints and causal relationships in terms of the predicates $\mathit{Possible}_{std}$, $\mathit{Possible}$, Causes_{std}, and Causes_{stb} along the line of [21]. More precisely, suppose given two sets of steady and stabilizing state constraints, $\mathcal{C}_{std}$ and $\mathcal{C}_{stb}$, and two sets of steady and stabilizing causal relationships, $\mathcal{R}_{std}$ and $\mathcal{R}_{stb}$. Let then $\Sigma(\mathcal{C}_{std}, \mathcal{C}_{stb}, \mathcal{R}_{std}, \mathcal{R}_{stb})$ be a theory consisting of

1. the theory *EUNA*;
2. definitions of $\mathit{Possible}_{std}$ and $\mathit{Possible}$ such that if S is a state, then
   (a) S satisfies $\mathcal{C}_{std}$ iff $\Sigma \models \mathit{Possible}_{std}(\tau_S)$, and
   (b) S satisfies $\mathcal{C}_{std} \cup \mathcal{C}_{stb}$ iff $\Sigma \models \mathit{Possible}(\tau_S)$;
3. definitions of Causes_{std} and Causes_{stb} such that if S is a state, E a set of fluent literals, and s', e' two collections of fluent literals, then
   (a) $\Sigma \models \mathit{Causes}_{std}(\tau_S, \tau_E, s', e')$ iff there exist two sets of fluent literals S', E' such that $\mathit{EUNA} \models s' = \tau_{S'} \wedge e' = \tau_{E'}$ and $(S, E) \rightsquigarrow_{\mathcal{R}_{std}} (S', E')$, and
   (b) $\Sigma \models \mathit{Causes}_{stb}(\tau_S, \tau_E, s', e')$ iff there exist two sets of fluent literals S', E' such that $\mathit{EUNA} \models s' = \tau_{S'} \wedge e' = \tau_{E'}$ and $(S, E) \rightsquigarrow_{\mathcal{R}_{stb}} (S', E')$.

On this basis we can prove correctness of our axioms of Section 3.

Theorem 3 *Let $\mathcal{C}_{std}$ and $\mathcal{C}_{stb}$ be sets of steady and stabilizing, respectively, state constraints, and $\mathcal{R}_{std}$ and $\mathcal{R}_{stb}$ be sets of steady and stabilizing, respectively, causal relationships. Let Σ^* be $\Sigma(\mathcal{C}_{std}, \mathcal{C}_{stb}, \mathcal{R}_{std}, \mathcal{R}_{stb})$ augmented by the axioms (6) and (7).*

Consider a state S, a set of fluent literals E, and a collection of fluent literals $\hat{s}$. Then,

$$\Sigma^* \models \mathit{Ramify}(\tau_S, \tau_E, \hat{s})$$

iff there exist states $S_0, S'_0, \ldots, S_n, S'_n$ and sets of fluent literals $E_0, E'_0, \ldots, E_n, E'_n$ $(n \geq 0)$ such that $S_0 = S$, $E_0 = E$, $\mathit{EUNA} \models \hat{s} = \tau_{S'_n}$,

$$\begin{aligned} (S_0, E_0) &\overset{*}{\rightsquigarrow}_{\mathcal{R}_{std}} (S'_0, E'_0) \\ &\rightsquigarrow_{\mathcal{R}_{stb}} (S_1, E_1) \overset{*}{\rightsquigarrow}_{\mathcal{R}_{std}} (S'_1, E'_1) \\ &\ldots \\ &\rightsquigarrow_{\mathcal{R}_{stb}} (S_n, E_n) \overset{*}{\rightsquigarrow}_{\mathcal{R}_{std}} (S'_n, E'_n) \end{aligned} \tag{8}$$

and, for each $0 \leq i \leq n$, all elements of $\mathcal{C}_{std}$ are true in S'_i, and in S'_n also all elements of $\mathcal{C}_{stb}$ are true.

Proof: Let $n \geq 0$, fix some $i = 0, \ldots, n$, and let s'_i, e'_i be two collections of fluent literals and S_i a state and E_i a set of fluent literals. Following the standard semantics of second-order logic (see, e.g., [10]), the second conjunct in the right hand side of the equivalence in (6) is true under Σ^* for $\{s \mapsto \tau_{S_i}, e \mapsto \tau_{E_i}, s' \mapsto s'_i, e \mapsto e'_i\}$ iff $(\tau_{S_i}, \tau_{E_i}), (s'_i, e'_i)$ belongs to the reflexive and transitive closure of Causes_{std}, that is, iff there are terms $\zeta_0, \eta_0, \ldots, \zeta_k, \eta_k$ $(k \geq 0)$ such that $\zeta_0 = \tau_{S_i}$, $\eta_0 = \tau_{E_i}$, $\zeta_k = s'_i$, $\eta_k = e'_i$, and $\Sigma^* \models \mathit{Causes}_{std}(\zeta_j, \eta_j, \zeta_{j+1}, \eta_{j+1})$ for all $0 \leq j < k$. According to the assumption about the axiomatization of Causes_{std} in Σ, the latter holds iff there exist two sets of fluent literals S'_i, E'_i such that $\mathit{EUNA} \models s'_i = \tau_{S'_i} \wedge e'_i = \tau_{E'_i}$ and $(S_i, E_i) \overset{*}{\leadsto}_{\mathcal{R}_{std}} (S'_i, E'_i)$. The first conjunct in the right hand side of (6) additionally ensures that state S'_i satisfies $\mathcal{C}_{std}$, again according to the assumption about Σ. To summarize, $\Sigma^* \models \mathit{Ramify}_{std}(\tau_{S_i}, \tau_{E_i}, s'_i, e'_i)$ iff $(S_i, E_i) \overset{*}{\leadsto}_{\mathcal{R}_{std}} (S'_i, E'_i)$ for some S'_i, E'_i such that $\mathit{EUNA} \models s'_i = \tau_{S'_i} \wedge e'_i = \tau_{E'_i}$ and all constraints in $\mathcal{C}_{std}$ are true in S'_i.

Now, let $s = \tau_S$ and $e = \tau_E$. The second conjunct in the right hand side of the equivalence in (7) is true under Σ^* iff there exists some $\hat{e}$ such that $(s, e), (\hat{s}, \hat{e})$ belongs to the reflexive and transitive closure of joining Ramify_{std} and Causes_{stb}. According to the assumption about Σ and the correctness of Σ^* wrt. Ramify_{std} as just proved, this is equivalent to the existence of two sets of fluent literals $\hat{S}, \hat{E}$ such that

- $\mathit{EUNA} \models \hat{s} = \tau_{\hat{S}} \wedge \hat{e} = \tau_{\hat{E}}$;
- derivation (8) holds for $S_0 = S$, $E_0 = E$, $S'_n = \hat{S}$, and $E'_n = \hat{E}$; and
- for each $0 \leq i \leq n$, all elements of $\mathcal{C}_{std}$ are true in S'_i.

Again according to the assumption about Σ, the first conjunct in the right hand side of (7) additionally ensures that $\hat{S}$ satisfies both $\mathcal{C}_{std}$ and $\mathcal{C}_{stb}$, which completes the proof.

■

References

[1] Sven-Erik Bornscheuer and Michael Thielscher. Explicit and implicit indeterminism: Reasoning about uncertain and contradictory specifications of dynamic systems. *Journal of Logic Programming*, 31(1–3):119–155, 1997.

[2] Charles Elkan. Reasoning about action in first-order logic. In *Proceedings of the Conference of the Canadian Society for Computational Studies of Intelligence (CSCSI)*, pages 221–227, Vancouver, Canada, May 1992. Morgan Kaufmann.

[3] Richard E. Fikes and Nils J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208, 1971.

[4] Matthew L. Ginsberg and David E. Smith. Reasoning about action I: A possible worlds approach. *Artificial Intelligence*, 35:165–195, 1988.

[5] Matthew L. Ginsberg and David E. Smith. Reasoning about action II: The qualification problem. *Artificial Intelligence*, 35:311–342, 1988.

[6] Enrico Giunchiglia, G. Neelakantan Kartha, and Vladimir Lifschitz. Representing action: Indeterminacy and ramifications. *Artificial Intelligence*, 95:409–443, 1997.

[7] Steffen Hölldobler and Josef Schneeberger. A new deductive approach to planning. *New Generation Computing*, 8:225–244, 1990.

[8] Joxan Jaffar, Jean-Louis Lassez, and Michael J. Maher. A theory of complete logic programs with equality. *Journal of Logic Programming*, 1(3):211–223, 1984.

[9] Antonis Kakas and Rob Miller. Reasoning about actions, narratives, and ramifications. *Electronic Transactions on Artificial Intelligence*, 1(4):39–72, 1997.

[10] Daniel Leivant. Higher order logic. In D. M. Gabbay, C. J. Hogger, and J. A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 2, pages 229–321. Oxford University Press, 1994.

[11] Vladimir Lifschitz. Frames in the space of situations. *Artificial Intelligence*, 46:365–376, 1990.

[12] Fangzhen Lin. Embracing causality in specifying the indirect effects of actions. In C. S. Mellish, editor, *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1985–1991, Montreal, Canada, August 1995. Morgan Kaufmann.

[13] Fangzhen Lin and Ray Reiter. State constraints revisited. *Journal of Logic and Computation*, 4(5):655–678, 1994.

[14] Norman McCain and Hudson Turner. A causal theory of ramifications and qalifications. In C. S. Mellish, editor, *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1978–1984, Montreal, Canada, August 1995. Morgan Kaufmann.

[15] Norman McCain and Hudson Turner. Causal theories of action and change. In B. Kuipers and B. Webber, editors, *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 460–465, Providence, RI, July 1997. MIT Press.

[16] John McCarthy and Patrick J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence*, 4:463–502, 1969.

[17] Edwin Pednault. Formulating multi-agent, dynamic world problems in the classical planning framework. In M. P. Georgeff and A. L. Lansky, editors, *Proceedings of the Workshop on Reasoning about Actions & Plans*, pages 47–82, San Mateo, CA, 1986. Morgan Kaufmann.

[18] Ray Reiter. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In V. Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation*, pages 359–380. Academic Press, 1991.

[19] Erik Sandewall. Assessments of ramification methods that use static domain constraints. In L. C. Aiello, J. Doyle, and S. Shapiro, editors, *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 99–110, Cambridge, MA, November 1996. Morgan Kaufmann.

[20] Murray Shanahan. *Solving the Frame Problem: A Mathematical Investigation of the Common Sense Law of Inertia.* MIT Press, 1997.

[21] Michael Thielscher. Ramification and causality. *Artificial Intelligence*, 89(1–2):317–364, 1997.

[22] Michael Thielscher. Towards state update axioms: Reifying successor state axioms. Accepted for the *European Workshop on Logics in AI (JELIA)* 1998. (Electronically available at `www.intellektik.informatik.tu-darmstadt.de/~mit`).