Steady Versus Stabilizing State Constraints

MICHAEL THIELSCHER
Darmstadt University of Technology, Germany
e-mail: mit@informatik.tu-darmstadt.de

Abstract. A new distinction between two kinds of state constraints is introduced and proved important for solving the Ramification Problem in Reasoning about Actions. *Steady* constraints never, not even for an instant, cease to 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 causal lag. Carelessly mixing these two types of indirect effects is shown to producing anomalous models. On the basis of the existing theory of causal relationships we illustrate how to deal with the distinction appropriately.

1 Introduction

In formal systems for reasoning about actions, the Ramification Problem denotes the problem of handling indirect effects [Ginsberg and Smith, 1988a]. These effects are not explicitly represented in action specifications but follow from general laws, so-called *state constraints*, which formalize 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 apparently 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 [Ginsberg and Smith, 1988b, Lin and Reiter, 1994]. This led to a distinction between so-called *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 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 stop 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 ducks are blocked. Here it is possible to bring about a situation where the state constraint does

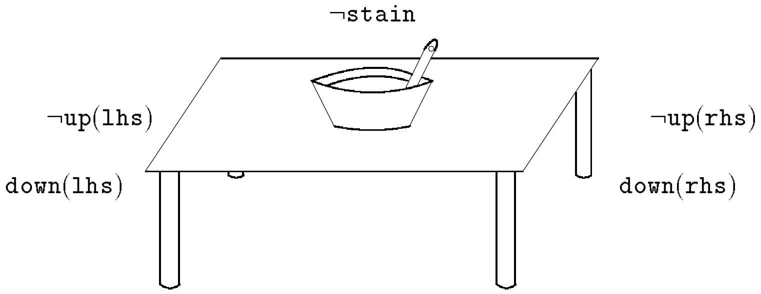

Figure 1: A bowl filled with soup is standing on a table. The soup spills out and produces a stain if the table is lifted on one hand side but not the other. Nothing of this sort is expected when lifting up the table on both sides simultaneously.

---

not hold, at least for an instant: If the room is full of fresh air and we block the last remaining free duck, then for a short period of time all ducks are blocked with the room still not being stuffy. So for a moment the constraint ceases to being in force.

Steady and stabilizing state constraints coexist ever since the introduction of the Ramification Problem.[1] 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 ducks, then we expect that as a side effect the room gets stuffy. However, it seems that there is a qualitative difference between the indirect effects triggered by steady and those triggered by stabilizing state constraints. Namely, the latter materialize not without a causal lag, tiny and imperceptible as it might be, as opposed to indirect effects deriving from steady constraints, which occur instantaneously. This raises the question whether their uniform treatment may be problematic if both give rise to side effects of the same action.

Indeed a simple scenario shows that unexpected conclusions may arise from ignoring the distinction. Suppose a bowl well filled with soup is standing on a rectangular table; see Figure 1. 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. Let us use the following five atomic propositions, or *fluents*, to describe the various states in this domain:

$$\mathtt{down(lhs)},\ \mathtt{down(rhs)},\ \mathtt{up(lhs)},\ \mathtt{up(rhs)},\ \mathtt{stain}$$

Propositions $\mathtt{down}(x)$ and $\mathtt{up}(x)$, respectively, shall be true if side x of the table is currently down (resp. up), and $\mathtt{stain}$ shall be true if the tablecloth is currently stained. These fluents depend on each other as described by these two state constraints:

$$\forall x.\ \mathtt{down}(x) \equiv \neg\mathtt{up}(x) \tag{1}$$

$$(\mathtt{up(lhs)} \wedge \neg\mathtt{up(rhs)}) \vee (\mathtt{up(rhs)} \wedge \neg\mathtt{up(lhs)}) \supset \mathtt{stain} \tag{2}$$

The first one says that at any time each side of the table is either down or else up. The second one formalizes the existence of a stain whenever one but not the opposite side of the table is

[1] In fact, the two example constraints just mentioned were taken from [Ginsberg and Smith, 1988a].

up. To which of our two categories do these constraints belong? The first one is steady, for we can think of it as a definition of $\texttt{down}$ as not being $\texttt{up}$ (or vice versa)—and definitions cannot possibly, not even for an instant, be violated. The second one is stabilizing, for the stain is not produced without a causal lag 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. Propositions $\texttt{down}$ and $\texttt{up}$ 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 either of the two disjuncts in the left hand side of the implication becoming true. Throughout this paper, we will formalize such causal knowledge of indirect effects as so-called *causal relationships* [Thielscher, 1997]. Their general format is ε $\underline{\text{causes}}$ ϱ $\underline{\text{if}}$ Φ, which should be read as: If ε is a direct or indirect effect of an action, then ϱ occurs as indirect effect, provided context Φ holds. The various causal relationships that hold in our example thus are the following:

$$\begin{array}{rcrcl@{\qquad}rcrcl}
\neg\texttt{down}(x) & \underline{\text{causes}} & \texttt{up}(x) & \underline{\text{if}} & \top & \texttt{up(lhs)} & \underline{\text{causes}} & \texttt{stain} & \underline{\text{if}} & \neg\texttt{up(rhs)} \\
\texttt{down}(x) & \underline{\text{causes}} & \neg\texttt{up}(x) & \underline{\text{if}} & \top & \texttt{up(rhs)} & \underline{\text{causes}} & \texttt{stain} & \underline{\text{if}} & \neg\texttt{up(lhs)} \\
\texttt{up}(x) & \underline{\text{causes}} & \neg\texttt{down}(x) & \underline{\text{if}} & \top & \neg\texttt{up(rhs)} & \underline{\text{causes}} & \texttt{stain} & \underline{\text{if}} & \texttt{up(lhs)} \\
\neg\texttt{up}(x) & \underline{\text{causes}} & \texttt{down}(x) & \underline{\text{if}} & \top & \neg\texttt{up(lhs)} & \underline{\text{causes}} & \texttt{stain} & \underline{\text{if}} & \texttt{up(rhs)}
\end{array} \tag{3}$$

(where formula $\top$ denotes a tautology). Relationship $\texttt{up(lhs)}$ $\underline{\text{causes}}$ $\texttt{stain}$ $\underline{\text{if}}$ $\neg\texttt{up(rhs)}$, for instance, indicates that any action with effect $\texttt{up(lhs)}$ while $\neg\texttt{up(rhs)}$ holds, has $\texttt{stain}$ as indirect effect. Notice that the causal relationships in the first column describe indirect effects which occur without a causal lag since they derive from a steady constraint. Deriving from a stabilizing constraint, the relationships in the second column each describe an indirect effect with causal lag.

Now, suppose the current state be that the table is standing firmly on the floor and there is no stain, i.e.,

$$\texttt{down(lhs)} \wedge \neg\texttt{up(lhs)} \wedge \texttt{down(rhs)} \wedge \neg\texttt{up(rhs)} \wedge \neg\texttt{stain}$$

as depicted in Figure 1. 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 $\neg\texttt{down(lhs)}$ and $\neg\texttt{down(rhs)}$. So the preliminary result of our action, where no indirect effects have yet been generated, is

$$\neg\texttt{down(lhs)} \wedge \neg\texttt{up(lhs)} \wedge \neg\texttt{down(rhs)} \wedge \neg\texttt{up(rhs)} \wedge \neg\texttt{stain}$$

Proceeding with adjusting according to possible indirect effects of our action, we see that $\texttt{up(lhs)}$ should become true as a side effect of $\neg\texttt{down(lhs)}$. Formally this follows from the topmost causal relationship in the first column of equation (3). Accommodating this effect results in

$$\neg\texttt{down(lhs)} \wedge \texttt{up(lhs)} \wedge \neg\texttt{down(rhs)} \wedge \neg\texttt{up(rhs)} \wedge \neg\texttt{stain}$$

As for the next step, the most reasonable thing to do would be to likewise change $\neg\texttt{up(rhs)}$ to $\texttt{up(rhs)}$ as a side effect of $\neg\texttt{down(rhs)}$. The result would be that the two sides of the table are both up, not down, and that no stain has been produced—the natural 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 topmost causal relationship in the second column: $\texttt{up(lhs)}$ occurred as (indirect) effect while $\neg$up(rhs) still holds. If afterwards we resolve the conflict of having both $\neg$down(rhs) and $\neg$up(rhs), then the final result is a state which satisfies

$$\neg\texttt{down(lhs)} \wedge \texttt{up(lhs)} \wedge \neg\texttt{down(rhs)} \wedge \texttt{up(rhs)} \wedge \texttt{stain}$$

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 stain, which occurs with causal lag, *before* accounting for the instantaneous effect up(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, 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 with causal 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 our abstract theory of causal relationships as described in [Thielscher, 1997]. In Section 3, we then sketch corresponding modifications of an existing axiomatization in the Fluent Calculus. In Section 4, we give reasons why the insight gained in this paper is of general importance for action formalisms dealing with ramifications. The paper ends with a discussion on ramifications vs. delayed effects, a topic which naturally arises when considering stabilizing state constraints.

2 Steady vs. Stabilizing Causal Relationships

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 properties, sometimes of entities, such as stain or down(lhs) etc. A *fluent literal* is a fluent or its negation. We say that a set of fluent literals is *inconsistent* if it contains a fluent along with its negation. A *state* is a maximal consistent set of 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. The notion of fluent formulas being *true* in a state S is based on defining a fluent literal ℓ to be true if and only if $\ell \in S$. E.g., the two fluent formulas of equations (1) and (2) are true in the state depicted in Figure 1 but false in, say, the state $\{\neg\texttt{down(lhs)}, \texttt{up(lhs)}, \neg\texttt{down(rhs)}, \neg\texttt{up(rhs)}, \neg\texttt{stain}\}$. *State constraints* are fluent formulas which constrain the set of states to those which can possibly occur in the domain.

The second fundamental notion are the *actions*. Actions cause state transitions. The (direct) effect of an action is specified by saying which fluents change their truth-value when the action is being performed. *Action laws* serves this purpose: They are of the form $a(\widetilde{x})\ \underline{\text{transforms}}\ C\ \underline{\text{into}}\ E$ where

- $\widetilde{x}$ is a sequence of pairwise distinct variables;

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

An example for an action law is

$$\texttt{lift\_up\_both}(x, y) \ \underline{\text{transforms}} \ \{\texttt{down}(x), \texttt{down}(y)\} \ \underline{\text{into}} \ \{\neg\texttt{down}(x), \neg\texttt{down}(y)\} \tag{4}$$

If S is a state, then an instance $\alpha\{\widetilde{x} \mapsto \widetilde{e}\}$ of an action law $\alpha = a(\widetilde{x})$ $\underline{\text{transforms}}$ C $\underline{\text{into}}$ E is *applicable* to S iff $C\{\widetilde{x} \mapsto \widetilde{e}\} \subseteq S$; the *application* yields the state $(S \setminus C\{\widetilde{x} \mapsto \widetilde{e}\}) \cup E\{\widetilde{x} \mapsto \widetilde{e}\}$.[2]

The states resulting from applying action laws, which concentrate on the direct effects, may violate the underlying state constraints: For instance, if we apply the aforementioned law for $\texttt{lift\_up\_both}(\texttt{lhs}, \texttt{rhs})$ to the state depicted in Figure 1, then $\forall x.\, \texttt{down}(x) \equiv \neg\texttt{up}(x)$ 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 $\underline{\text{causes}}$ ϱ $\underline{\text{if}}$ 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. 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 $r\{\widetilde{x} \mapsto \widetilde{e}\}$ of a causal relationship $r = \varepsilon$ $\underline{\text{causes}}$ ϱ $\underline{\text{if}}$ Φ with free variables $\widetilde{x}$ *applicable* to (S, E) iff $\varepsilon\{\widetilde{x} \mapsto \widetilde{e}\} \in E$ and $\Phi\{\widetilde{x} \mapsto \widetilde{e}\} \wedge \neg\varrho\{\widetilde{x} \mapsto \widetilde{e}\}$ is true in S. The *application* yields the pair (S', E') where $S' = (S \setminus \{\neg\varrho\{\widetilde{x} \mapsto \widetilde{e}\}\}) \cup \{\varrho\{\widetilde{x} \mapsto \widetilde{e}\}\}$ and $E' = (E \setminus \{\neg\varrho\{\widetilde{x} \mapsto \widetilde{e}\}\}) \cup \{\varrho\{\widetilde{x} \mapsto \widetilde{e}\}\}$. In words, a causal relationship is applicable if the associated context Φ holds, the particular indirect effect ϱ is currently false, and 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 $(S, E) \overset{*}{\leadsto}_{\mathcal{R}} (S', E')$ to indicate that there are causal relationships in $\mathcal{R}$ whose successive application to (S, E) yields (S', E').

We have somewhat loosely said that indirect effects are consequences of state constraints. Having the formal definition of causal relationships, this correspondence can be stated more precisely. A causal relationship ε $\underline{\text{causes}}$ ϱ $\underline{\text{if}}$ Φ is consequence of some state constraint if the latter implies $\Phi \wedge \varepsilon \supset \varrho$. However, not all such purely logical consequences of state constraints correspond to indirect effects, as has first been observed in [Lifschitz, 1990]. Causal relationships thus contain more information than the mere state constraints. It is, however, 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. For details see [Thielscher, 1997].[3]

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

[3] For example, the causal relationships of equation (3) can be automatically obtained from the underlying state constraints of equation (1) and (2), respectively, on the basis of the influence information that `down` may affect `up` and vice versa, and that `up` may affect `stain`.

State constraints are either steady or stabilizing. To which category a constraint belongs is 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 introduction showed that during the application of causal relationships the insertion of an effect with real causal lag 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. 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 $S_0 = (S \setminus C) \cup E$, $E_0 = E$,

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

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 proves 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 with causal lag as discussed in Section 1.

Proposition 2 *Let given a domain specification consisting of*

- *the entities* `lhs` *and* `rhs`*, the unary fluent names* `down` *and* `up`*, and the nullary fluent name* `stain`*;*
- *the unary action* `lift_up_both` *in conjunction with the action law of equation (4);*
- *the steady state constraint of equation (1) and the stabilizing state constraint of equation (2); and*
- *the causal relationships of equation (3), of which the ones in the first column are steady while the others are stabilizing.*

Then there is a unique successor of $S = \{\texttt{down(lhs)}, \neg\texttt{up(lhs)}, \texttt{down(rhs)}, \neg\texttt{up(rhs)}, \neg\texttt{stain}\}$ *and* $a = \texttt{lift\_up\_both(lhs,rhs)}$*, in which* $\neg\texttt{stain}$ *holds.*

Proof: Let $\mathcal{C}_{std} = \{(1)\}$, $\mathcal{C}_{stb} = \{(2)\}$, and let $\mathcal{R}_{std}$ and $\mathcal{R}_{stb}$ consist of the first and second, respectively, column of equation (3). 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 $E = \{\neg\texttt{down(lhs)}, \neg\texttt{down(rhs)}\}$. So there is a unique pair (S_0, E_0) to start off, viz.

$$\begin{aligned}(\, S_0 &= \{\neg\texttt{down(lhs)}, \neg\texttt{up(lhs)}, \neg\texttt{down(rhs)}, \neg\texttt{up(rhs)}, \neg\texttt{stain}\}, \\ E_0 &= \{\neg\texttt{down(lhs)}, \neg\texttt{down(rhs)}\} \,)\end{aligned}$$

The first component violates $\mathcal{C}_{std}$. Two instances of causal relationships in $\mathcal{R}_{std}$ are applicable, namely,

$$\begin{array}{c}\neg\texttt{down(lhs)} \ \underline{\text{causes}} \ \texttt{up(lhs)} \ \underline{\text{if}} \ \top \\ \neg\texttt{down(rhs)} \ \underline{\text{causes}} \ \texttt{up(rhs)} \ \underline{\text{if}} \ \top\end{array}$$

If either of them is applied to (S_0, E_0), then the state component of the resulting pair still does not satisfy $\forall x.\, \texttt{down}(x) \equiv \neg\texttt{up}(x)$. 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.

$$\begin{aligned}(\, S_0' &= \{\neg\texttt{down(lhs)}, \texttt{up(lhs)}, \neg\texttt{down(rhs)}, \texttt{up(rhs)}, \neg\texttt{stain}\}, \\ E_0' &= \{\neg\texttt{down(lhs)}, \neg\texttt{down(rhs)}, \texttt{up(lhs)}, \texttt{up(rhs)}\} \,)\end{aligned}$$

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 Fluent Calculus-based axiomatization of the theory of causal relationships [Thielscher, 1997] in order to cope with the distinction between steady and stabilizing ramifications. As opposed to the Situation Calculus [McCarthy and Hayes, 1969], the Fluent Calculus [Hölldobler and Schneeberger, 1990] employs structured state terms which each consists in a collection of all 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 following term:

$$\texttt{down(lhs)} \circ \neg\texttt{up(lhs)} \circ \texttt{down(rhs)} \circ \neg\texttt{up(rhs)} \circ \neg\texttt{stain}$$

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 [Hölldobler and Schneeberger, 1990] 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 the following, we concentrate on the part of the axiomatization of [Thielscher, 1997] 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 $Possible(s)$ 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 $Possible_{std}(s)$, which shall be true iff its argument satisfies just all steady constraints.

Predicate $Causes(s, e, s', e')$ has been defined as the existence of a causal relationship that maps (s, e) into (s', e') (or rather the states and sets of effects being represented). 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. More precisely, $Ramify(s, e, s')$ has been, and shall still be, defined in such a way that an instance is 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 $Ramify_{std}(s, e, s', e')$, an 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 $Possible_{std}(s')$ holds. In essence the definitions of *Ramify* and $Ramify_{std}$ are transitive closures. As this mathematical concept cannot be expressed in first-order logic, we employ the standard way of encoding transitive closure by means of second-order formulas:

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

$$
\wedge\ Possible_{std}(s')
$$

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

Analogously, we define

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

$$
\wedge\ Possible(s')
$$

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

This completes the modifications 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 $Ramify$, respectively, and from the main correctness result given in [Thielscher, 1997].

4 Assessment

We have seen how ignoring the distinction between steady and stabilizing state constraints can lead to anomalous models if indirect effects are accommodated by the theory of causal relationships. How does this insight affect action formalisms in general? Our answer comes in three steps. First of all, the encountered problem is certainly not a particularity of causal relationships themselves. Rather it would arise in any formalism by which ramifications are calculated step-wise by constructing a sequence of intermediate states.

Secondly, however, the problem does not occur in approaches where all obtained indirect effects need to be 'justified,' on the basis of the direct effects, wrt. the initial or the resulting state.[4] The reason for this, informally speaking, is the following. The anomalous model in our key example obtains by a sequence of effects $\ldots\varepsilon_i\ldots\varepsilon_j\ldots\varepsilon_k\ldots$ so that after ε_i some steady and also some stabilizing constraint is violated, which both get re-established by ε_k, but beforehand the violation of the steady constraint is, unacceptably, 'exploited' to insert the stabilizing effect ε_j. Now, the temporary justification for this effect cannot be gathered from any state prior to ε_i or after ε_k, in particular not from the initial nor the overall resulting successor state. So if the latter are the only points of reference, then the anomalous indirect effect ε_j is unjustified, hence denied. At first glance it therefore seems that the distinction between steady and stabilizing constraints is irrelevant for this type of approaches.

Yet, and now we come to the third part of our answer, referring solely to the initial and/or final state, and thus ignoring intermediate states, not only avoids anomalous models but also misses intended ones. This can be seen by a scenario discussed in Section 5.2 of [Thielscher, 1997]. There, a particular model also obtains via a chain of effects $\ldots\varepsilon_i\ldots\varepsilon_j\ldots\varepsilon_k\ldots$ in which ε_j 'exploits' the temporary violation of some constraint after ε_i and prior to ε_k. But this time both ε_j and ε_k are stabilizing indirect effects, so that ε_j may indeed precede ε_k. The aforementioned approaches ignore the now perfectly reasonable possibility of ε_j being 'squeezed in,' and so entail erroneous conclusions to the effect that ε_j never occurs. Getting this effect right requires to somehow compute ramifications step-by-step by establishing intermediate states. This shows that the notion of steady vs. stabilizing state constraints has general significance: It should first be used to check whether the domain at hand includes stabilizing constraints, in which case intermediate states need to be considered when calculating indirect effects; and if so, then we had better take into account the different nature of the two types of constraints.

5 Ramifications vs. Delayed Effects

We have started off with the question whether 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 causal lag between some particular indirect effect and its triggering cause may generally be shorter than the causal

[4] Most of today's approaches are of this kind, e.g., [Elkan, 1992, McCain and Turner, 1995, Lin, 1995, Shanahan, 1997, Kakas and Miller, 1997].

lag between another particular effect and its cause. One approach to this problem is to introduce an explicit notion of time, namely, in specifying the exact delay between the occurrence of an effect and its cause. This, however, is not in the spirit of the Ramification Problem. For the latter is concerned with accounting for those indirect effects which so rapidly follow the performance of an action that common sense considers them virtually instantaneous. Had we assumed precise knowledge as to the delay of the soup's spilling, then the latter fell into the category of so-called *delayed* effects. Though certainly of importance, too, these effects need to be meticulously distinguished from those indirect effects investigated in the context of the Ramification Problem. For delayed effects deserve a separate state transition, that is, they should not occur during the very same transition step as its triggering effect. In contrast, solving the Ramification Problem means to account for all indirect effects that are summarized in a single state transition. This amounts to performing qualitative reasoning about causal lags, 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 causal lags—with the exception that real causal lags do need to be distinguished from those which are zero, as we have argued in this paper.

References

[Elkan, 1992] 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.

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

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

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

[Kakas and Miller, 1997] Antonis Kakas and Rob Miller. Reasoning about actions, narratives, and ramifications. *Linköping Electronic Articles in Computer and Information Science*, 2(no. 012), 1997. (URL: `http://www.ep.liu.se/ea/cis/1997/012/`).

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

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

[Lin, 1995] 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.

[McCain and Turner, 1995] 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.

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

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

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