In: Proceedings of JELIA'98, © Springer-Verlag

Towards State Update Axioms: Reifying Successor State Axioms

Michael Thielscher[⋆]

Dresden University of Technology
mit@pikas.inf.tu-dresden.de

Abstract. Successor state axioms are an optimal solution to the famous Frame Problem in reasoning about actions—but only as far as its representational aspect is concerned. We show how by gradually applying the principle of reification to these axioms, one can achieve gradual improvement regarding the inferential aspect without losing the representational merits. The resulting concept of *state update axioms* constitutes a novel version of what is known as the Fluent Calculus. We illustrate that under the provision that actions have no so-called open effects, any Situation Calculus specification can be transformed into an essentially equivalent Fluent Calculus specification, in which at the same time the representational and the inferential aspect of the Frame Problem are addressed. This alternative access to the Fluent Calculus both clarifies its role in relation to the most popular axiomatization paradigm and should help to enhance its acceptance.

1 Introduction

For a long time, the Fluent Calculus, introduced in [7] and so christened in [3], has been viewed exclusively as a close relative of approaches to the Frame Problem [12] which appeal to non-classical logics, namely, linearized versions of, respectively, the connection method [1, 2] and Gentzen's sequent calculus [11]. The affinity of the Fluent Calculus and these two formalisms has been emphasized by several formal comparison results. In [5], for example, the three approaches have been proved to deliver equivalent solutions to a resource-sensitive variant of STRIPS planning [4].

Yet the Fluent Calculus possesses a feature by which it stands out against the two other frameworks: It stays entirely within classical logic. In this setting the Fluent Calculus constitutes a successful attempt to address the Frame Problem as regards both the representational aspect (since no effect axiom or any other axiom needs to mention non-effects) and, at the same time, the inferential aspect (since carrying over persistent fluents from one situation to the next does not require separate deduction steps for each). Contrary to popular opinion, all this is achieved without relying on complete knowledge of the initial or any other situation. Nonetheless the Fluent Calculus has not yet received as much attention

[⋆] on leave from Darmstadt University of Technology in the scientific community as, say, the Situation Calculus. One reason might be that, due to its heritage, the relation to the mainstream calculi, and in particular to the Situation Calculus, has not yet been convincingly elaborated.

The purpose of this paper is to present an alternative approach to the Fluent Calculus, where we start off from the Situation Calculus in the version where successor state axioms are used as means to solve the representational aspect of the Frame Problem [13]. We illustrate how the Fluent Calculus can be viewed as the result of gradually improving this approach in view of the inferential aspect but without losing its representational merits. The key is to gradually apply the principle of reification, which means to use terms instead of atoms as the formal denotation of statements. Along the path leading from successor state axioms to the Fluent Calculus lies an intermediate approach, namely, the alternative formulation of successor state axioms described by [9], in which atomic fluent formulas are reified. This alternative design inherits the representational advantages and additionally addresses the inferential Frame Problem. Yet it does so only under the severe restriction that complete knowledge of the values of the relevant fluents in the initial situation is available. The Fluent Calculus can then be viewed as a further improvement in that it overcomes this restriction by carrying farther the principle of reification to conjunctions of fluents. In the following section we illustrate by means of examples how successor state axioms can thus be reified to what we call *state update axioms*. In Section 3 we then present a fully mechanic method to derive state update axioms from effect specifications with arbitrary first-order condition. One restriction turns out necessary for this method to be correct, namely, that actions do not have so-called *open* effects.[2] In Section 4, we will briefly show how to design state update axioms for actions with such effects.

Viewed in the way we pursue in this paper, the Fluent Calculus presents itself as the result of a successful attempt to cope with the inferential Frame Problem, starting off from successor state axioms as a solution to the representational aspect. Our hope is that this alternative access clarifies the role of this axiomatization paradigm in relation to the most popular approach and helps enhancing its acceptance. Following the new motivation it should become clearer that the Fluent Calculus provides an expressive axiomatization technique, in the setting of classical logic, which altogether avoids non-effect axioms and at the same time successfully copes with the inferential aspect of the Frame Problem.

2 From Situation Calculus to Fluent Calculus

2.1 From Successor State Axioms (I) ...

Reasoning about actions is inherently concerned with change: Properties expire, objects come into being and cease to exist, true statements about the state

[2] This concept is best explained by an example. This axiom specifies an open effect: $\forall x, y, s.\ \mathit{Bomb}(x) \land \mathit{Nearby}(x, y, s) \supset \mathit{Destroyed}(y, \mathit{Do}(\mathit{Explodes}(x), s))$. Even after instantiating the action expression $\mathit{Explodes}(x)$ and the situation term s, the effect literal still carries a variable, y, so that the action may have infinitely many effects.

of affairs at some time may be entirely wrong at another time. The first and fundamental challenge of formalizing reasoning about actions is therefore to account for the fact that most properties in the real world possess just a limited period of validity. This unstable nature of properties which vary in the course of time has led to calling them "fluents." In order to account for fluents changing their truth values in the course of time as consequences of actions, the Situation Calculus paradigm [12] is to attach a situation argument to each fluent, thus limiting its range of validity to a specific situation. The performance of an action then brings about a new situation in which certain fluents may no longer hold.

As an example which will be used throughout the paper, we will formalize the reasoning that led to the resolution of the following little mystery:

> A reliable witness reported that the murderer poured some milk into a cup of tea before offering it to his aunt. The old lady took a drink or two and then she suddenly fell into the armchair and died an instant later, by poisoning as has been diagnosed afterwards. According to the witness, the nephew had no opportunity to poison the tea beforehand. This proves that it was the milk which was poisoned and by which the victim was murdered.

The conclusion in this story is obviously based on some general commonsense knowledge of poisoned substances and the way they may affect people's health. To begin with, let us formalize by means of the Situation Calculus the relevant piece of knowledge that mixing a poisoned substance into another one causes the latter to be poisoned as well. To this end, we use the binary predicate $\mathit{Poisoned}(x,s)$ representing the fact that x is poisoned in situation s, the action term $\mathit{Mix}(p,x,y)$ denoting the action carried out by agent p of mixing x into y, and the binary function $\mathit{Do}(a,s)$ which denotes the situation to which leads the performance of action a in situation s.[3] With this signature and its semantics the following axiom formalizes the fact that if x is poisoned in situation s then y, too, is poisoned in the situation that obtains when someone mixes x into y:

$$\mathit{Poisoned}(x,s) \supset \mathit{Poisoned}(y, \mathit{Do}(\mathit{Mix}(p,x,y),s)) \tag{1}$$

The second piece of commonsense knowledge relevant to our example concerns the effect of drinking poisoned liquids. Let $\mathit{Alive}(x,s)$ represent the property of x being alive in situation s, and let the action term $\mathit{Drink}(p,x)$ denote that p drinks x. Then the following axiom encodes the fact that if x is poisoned then person p ceases to being among the livings after she had drunk x:

$$\mathit{Alive}(p,s) \wedge \mathit{Poisoned}(x,s) \supset \neg \mathit{Alive}(p, \mathit{Do}(\mathit{Drink}(p,x),s)) \tag{2}$$

These two effect axioms, however, do not suffice to solve the mystery due to the Frame Problem, which has been uncovered as early as in [12]. To see why,

[3] A word on the notation: Predicate and function symbols, including constants, start with a capital letter whereas variables are in lower case, sometimes with sub- or superscripts. Free variables in formulas are assumed universally quantified.

let S_0 be a constant by which we denote the initial situation, and consider the assertion,

$$\neg Poisoned(Tea, S_0) \wedge Alive(Nephew, S_0) \wedge Alive(Aunt, S_0) \qquad (3)$$

Even with $Poisoned(Milk, S_0)$ added, $\neg Alive(Aunt, S_2)$ does not yet follow (where $S_2 = Do(Drink(Aunt, Tea), Do(Mix(Nephew, Milk, Tea), S_0)))$, because $Alive(Aunt, Do(Mix(Nephew, Milk, Tea), S_0))$ is needed for axiom (2) to apply but cannot be concluded. In order to obtain this and other intuitively expected conclusions, a number of non-effect axioms (or "frame axioms") need to be supplied, like the following, which says that people survive the mixing of substances:

$$Alive(x, s) \supset Alive(x, Do(Mix(p, y, z), s)) \qquad (4)$$

Now, the Frame Problem is concerned with the problems that arise from the apparent need for non-effect axioms like (4). Actually there are two aspects of this famous problem: The *representational* Frame Problem is concerned with the proliferation of all the many frame axioms. The *inferential* Frame Problem describes the computational difficulties raised by the presence of many non-effect axioms when it comes to making inferences on the basis of an axiomatization: To derive the consequences of a sequence of actions it is necessary to carry, one-by-one and almost all the time using non-effect axioms, each property through each intermediate situation.

With regard to the representational aspect of the Frame Problem, successor state axioms [13] provide a solution which is optimal in a certain sense, namely, in that it requires no extra frame axioms at all. The key idea is to combine, in a clear elaborated fashion, several effect axioms into a single one. The result, more complex than simple effect axioms like (1) and (2) but still mentioning solely effects, is designed in such a clever way that it implicitly contains sufficient information also about non-changes of fluents.

The procedure by which these axioms are set up is the following. Suppose $F(\vec{x})$ is among the fluents one is interested in. On the assumption that a fixed, finite set of actions is considered relevant, it should be possible to specify with a single formula $\gamma_F^+(\vec{x}, a, s)$ all circumstances by which $F(\vec{x})$ would be caused to become true. That is to say, $\gamma_F^+(\vec{x}, a, s)$ describes all actions a and conditions relative to situation s so that $F(\vec{x})$ is a positive effect of performing a in s. For example, among the actions we considered above there is one, and only one, by which the fluent $Poisoned(x)$ is made true, namely, mixing some poisonous y into x. Hence an adequate definition of $\gamma_{Poisoned}^+(x, a, s)$ is the formula $\exists p, y[a = Mix(p, y, x) \wedge Poisoned(y, s)]$.

A dual formula, $\gamma_F^-(\vec{x}, a, s)$, defines the circumstances by which fluent $F(\vec{x})$ is caused to become false. In our example we consider no way to 'decontaminate' a substance, which is why $\gamma_{Poisoned}^-(x, a, s)$ should be equated with a logical contradiction. For our second fluent, $Alive(x)$, the situation is just the other way round: While $\gamma_{Alive}^+(x, a, s)$ is false for any instance, the appropriate definition of $\gamma_{Alive}^-(x, a, s)$ is $\exists y[a = Drink(x, y) \wedge Alive(x, s) \wedge Poisoned(y, s)]$.

On the basis of suitable definitions for both γ_F^+ and γ_F^-, a complete account can be given of how the truth value of fluent F in a new situation depends on the old one, namely,

$$F(\vec{x}, Do(a,s)) \;\equiv\; \gamma_F^+(\vec{x},a,s) \;\vee\; [\, F(\vec{x},s) \wedge \neg\, \gamma_F^-(\vec{x},a,s)\,] \tag{5}$$

This is the general form of *successor state axioms*.[4] It says that the fluent F holds in a new situation if, and only if, it is either a positive effect of the action being performed, or it was already true and the circumstances were not such that the fluent had to become false. Notice that both γ^+ and γ^- talk exclusively about effects (positive and negative), not at all about non-effects. Nonetheless, by virtue of being bi-conditional, a successor state axiom implicitly contains all the information needed to entail any non-change of the fluent in question. For whenever neither $\gamma_F^+(\vec{x},a,s)$ nor $\gamma_F^-(\vec{x},a,s)$ is true, then (5) rewrites to the simple equivalence $F(\vec{x}, Do(a,s)) \equiv F(\vec{x},s)$.

The two successor state axioms for our example domain, given the respective formulas γ from above, are

$$\begin{aligned} \mathit{Poisoned}(x, Do(a,s)) \;\equiv\; & \exists p, y\, [\, a = \mathit{Mix}(p,y,x) \wedge \mathit{Poisoned}(y,s)\,] \\ & \vee\; \mathit{Poisoned}(x,s) \end{aligned} \tag{6}$$

and

$$\begin{aligned} & \mathit{Alive}(x, Do(a,s)) \;\equiv \\ & \quad \mathit{Alive}(x,s) \;\wedge\; \neg\exists y\, [\, a = \mathit{Drink}(x,y) \wedge \mathit{Alive}(x,s) \wedge \mathit{Poisoned}(y,s)\,] \end{aligned} \tag{7}$$

The latter, for instance, suffices to conclude that $\mathit{Alive}(\mathit{Aunt}, S_0)$ is not affected by the action $\mathit{Mix}(\mathit{Nephew}, \mathit{Milk}, \mathit{Tea})$—assuming "unique names" for actions, i.e., $\mathit{Mix}(p', x', y') \neq \mathit{Drink}(x,y)$. Thus we can spare the frame axiom (4).

By specifying the effects of actions in form of successor state axioms it is possible to avoid frame axioms altogether. These axioms thus provide us with an in a certain sense optimal solution to the Frame Problem, as far as the representational aspect is concerned.

2.2 ... via Successor State Axioms (II) ...

While successor state axioms are a good way to overcome the representational Frame Problem since no frame axioms at all are required, the inferential aspect is fully present. In order to derive which fluents hold and which do not after a sequence of actions, it is still necessary to carry, one-by-one, each fluent through each intermediate situation by separate instances of successor state axioms. In this respect nothing seems gained by incorporating knowledge of non-effects in complex effect axioms instead of using explicit frame axioms.

However, it has been shown in [9] that by formulating successor state axioms in a way that is somehow dual to the scheme (5), the inferential aspect can be

[4] For the sake of clarity we ignore the concept of action precondition in this paper, as it is irrelevant for our discussion (see Section 4).

addressed at least to a certain extent. Central to this alternative is the representation technique of reification. It means that properties like $\mathit{Poisoned}(x)$ are formally modeled as terms, in other words as objects, in logical axiomatizations. This allows for a more flexible handling of these properties within first-order logic. Let, to this end, $\mathit{Holds}(f,s)$ be a binary predicate representing the fact that in situation s holds the fluent f, now formally a term but still meaning a proposition.

The key to the alternative form of successor state axioms is to devise one for each action, and not for each fluent, which gives a complete account of the positive and negative effects of that action. Suppose $A(\vec{x})$ is an action, then it should be possible to specify with a single formula $\delta_A^+(\vec{x},f,s)$ the necessary and sufficient conditions on f and s so that f is a positive effect of performing $A(\vec{x})$ in s. In our running example, the appropriate definition of $\delta_{\mathit{Mix}}^+(p,x,y,f,s)$, say, is $[f = \mathit{Poisoned}(y,s)] \wedge \mathit{Holds}(\mathit{Poisoned}(x),s)$, while $\delta_{\mathit{Drink}}^+(p,x,f,s)$ should be equated with a logical contradiction since $\mathit{Drink}(p,x)$ has no relevant positive effect. A dual formula, $\delta_A^-(\vec{x},f,s)$, defines the necessary and sufficient conditions on f and s so that f is a negative effect of performing $A(\vec{x})$ in s. For instance, $\delta_{\mathit{Mix}}^-(p,x,y,f,s)$ should be false in any case, while $\delta_{\mathit{Drink}}^-(p,x,f,s)$ is suitably described by $[f = \mathit{Alive}(p)] \wedge \mathit{Holds}(\mathit{Alive}(p),s) \wedge \mathit{Holds}(\mathit{Poisoned}(x),s)$.

On the basis of δ_A^+ and δ_A^-, a complete account can be given of which fluents hold in situations reached by performing $A(\vec{x})$, namely,

$$\mathit{Holds}(f,\mathit{Do}(A(\vec{x}),s)) \;\equiv\; \delta_A^+(\vec{x},f,s) \,\vee\, [\,\mathit{Holds}(f,s) \wedge \neg\, \delta_A^-(\vec{x},f,s)\,] \qquad (8)$$

That is to say, the fluents which hold after performing the action $A(\vec{x})$ are exactly those which are among the positive effects or which held before and are not among the negative effects. The reader may contrast this scheme with (5) and in particular observe the reversed roles of fluents and actions.[5]

Given the formulas $\delta_{\mathit{Mix}}^+(p,x,y,f,s)$, $\delta_{\mathit{Mix}}^-(p,x,y,f,s)$, $\delta_{\mathit{Drink}}^+(p,x,f,s)$, and $\delta_{\mathit{Drink}}^+(p,x,f,s)$, respectively, from above, we thus obtain these two successor state axioms of type (II):

$$\begin{aligned}\mathit{Holds}(f,\mathit{Do}(\mathit{Mix}(p,x,y),s)) \;\equiv\;& f = \mathit{Poisoned}(y) \wedge \mathit{Holds}(\mathit{Poisoned}(x),s)\\ &\vee\, \mathit{Holds}(f,s)\end{aligned} \qquad (9)$$

and

$$\begin{aligned}&\mathit{Holds}(f,\mathit{Do}(\mathit{Drink}(p,x),s)) \;\equiv\\ &\qquad \mathit{Holds}(f,s) \,\wedge\, \neg\,[\,f = \mathit{Alive}(p) \wedge \mathit{Holds}(\mathit{Alive}(p),s)\\ &\qquad\qquad\qquad\qquad\qquad \wedge \mathit{Holds}(\mathit{Poisoned}(x),s)\,]\end{aligned} \qquad (10)$$

Notice that as before non-effects are not explicitly mentioned and no additional frame axioms are required, so the representational aspect of the Frame Problem is addressed with the alternative notion of successor state axioms just as well. The inferential advantage of the alternative design shows if we represent the

[5] Much like [13] roots in the axiomatization technique of [6], the foundations for the alternative form of successor state axioms were laid in [10].

collection of fluents that are true in a situation s by equating the atomic formula $\mathit{Holds}(f,s)$ with the conditions on f to hold in s. The following formula, for instance, constitutes a suitable description of the initial situation in our example:

$$\begin{aligned} &\mathit{Holds}(f,S_0) \;\equiv \\ &\qquad f = \mathit{Alive}(\mathit{Nephew}) \vee f = \mathit{Alive}(\mathit{Aunt}) \vee f = \mathit{Poisoned}(\mathit{Milk}) \end{aligned} \tag{11}$$

The crucial feature of this formula is that the situation argument, S_0, occurs only once. With this representational trick it becomes possible to obtain a complete description of a successor situation in one go, that is, by singular application of a successor state axiom. To see why, consider the axiom which specifies the effects of mixing, (9). If we substitute p, x, and y by Nephew, Milk, and Tea, respectively, and s by S_0, then we can replace the sub-formula $\mathit{Holds}(f,S_0)$ of the resulting instance by the equivalent disjunction as given in axiom (11). So doing yields the formula,

$$\begin{aligned} &\mathit{Holds}(f,\mathit{Do}(\mathit{Mix}(\mathit{Nephew},\mathit{Milk},\mathit{Tea}),S_0)) \;\equiv \\ &\qquad f = \mathit{Poisoned}(\mathit{Tea}) \wedge \mathit{Holds}(\mathit{Poisoned}(\mathit{Milk}),S_0) \\ &\qquad \vee f = \mathit{Alive}(\mathit{Nephew}) \vee f = \mathit{Alive}(\mathit{Aunt}) \vee f = \mathit{Poisoned}(\mathit{Milk}) \end{aligned}$$

which all at once provides a complete description of the successor situation. Given suitable axioms for equality, the above can be simplified, with the aid of (11), to

$$\begin{aligned} &\mathit{Holds}(f,\mathit{Do}(\mathit{Mix}(\mathit{Nephew},\mathit{Milk},\mathit{Tea}),S_0)) \;\equiv \\ &\qquad\qquad f = \mathit{Poisoned}(\mathit{Tea}) \vee f = \mathit{Alive}(\mathit{Nephew}) \\ &\qquad\qquad \vee f = \mathit{Alive}(\mathit{Aunt}) \vee f = \mathit{Poisoned}(\mathit{Milk}) \end{aligned}$$

The reader may verify that we can likewise infer the result of $\mathit{Drink}(\mathit{Aunt},\mathit{Tea})$ in the new situation by applying the appropriate instance of successor state axiom (10), which, after simplification, yields

$$\begin{aligned} &\mathit{Holds}(f,\mathit{Do}(\mathit{Drink}(\mathit{Aunt},\mathit{Tea}),\mathit{Do}(\mathit{Mix}(\mathit{Nephew},\mathit{Milk},\mathit{Tea}),S_0))) \;\equiv \\ &\qquad f = \mathit{Poisoned}(\mathit{Tea}) \vee f = \mathit{Alive}(\mathit{Nephew}) \vee f = \mathit{Poisoned}(\mathit{Milk}) \end{aligned}$$

At first glance it seems that the alternative design of successor state axioms provides an overall satisfactory solution to both aspects of the Frame Problem. No frame axioms at all are needed, and one instance of a single successor state axiom suffices to carry over to the next situation all unchanged fluents. However, the proposed method of inference relies on the very strong assumption that we can supply a complete account of what does and what does not hold in the initial situation. Formula (11) provides such a complete specification, because it says that any fluent is necessarily false in S_0 which does not occur to the right of the equivalence symbol. Unfortunately it is impossible to formulate partial knowledge of the initial state of affairs in a similarly advantageous fashion. Of course one can start with an incomplete specification like, for instance,

$$\mathit{Holds}(f,S_0) \;\subset\; [f = \mathit{Alive}(\mathit{Nephew}) \vee f = \mathit{Alive}(\mathit{Aunt})] \wedge f \neq \mathit{Poisoned}(\mathit{Tea})$$

which mirrors the incomplete description we used earlier (c.f. formula (3)). But then the elegant inference step from above, where we have simply replaced a sub-formula by an equivalent, is no longer feasible. In this case one is in no way better off with the alternative notion of successor state axioms; again separate instances need to be applied, one for each fluent, in order to deduce what holds in a successor situation.

2.3 ... to State Update Axioms

So far we have used reification to denote single properties by terms. The 'meta'-predicate Holds has been introduced which relates a reified fluent to a situation term, thus indicating whether the corresponding property is true in the associated situation. When formalizing collected information about a particular situation S as to which fluents are known to hold in it, the various corresponding atoms $\mathit{Holds}(f_i, S)$ are conjuncted using the standard logical connectives. We have seen how the inferential aspect of the Frame Problem is addressed if this is carried out in a certain way, namely, by equating $\mathit{Holds}(f,s)$ with some suitable formula Ψ. The effects of an action a can then be specified in terms of how Ψ modifies to some formula Ψ' such that $\mathit{Holds}(f, \mathit{Do}(a,s)) \equiv \Psi'$. We have also seen, however, that this representation technique is still not sufficiently flexible in that it is impossible to construct a first-order formula Ψ so that $\mathit{Holds}(f, S_0) \equiv \Psi$ provides a correct incomplete specification of S_0. Yet it is possible to circumvent this drawback by carrying farther the principle of reification, to the extent that not only single fluents but also their conjunctions are formally treated as terms. Required to this end is a binary function which to a certain extent reifies the logical conjunction. This function shall be denoted by the symbol " $\circ$ " and written in infix notation, so that, for instance, the term $\mathit{Alive}(\mathit{Nephew}) \circ \mathit{Poisoned}(\mathit{Milk})$ is the reified version of $\mathit{Alive}(\mathit{Nephew}) \wedge \mathit{Poisoned}(\mathit{Milk})$. The use of the function " $\circ$ " is the characteristic feature of axiomatizations which follow the paradigm of Fluent Calculus.

The union of all relevant fluents that hold in a situation is called the *state* (of the world) in that situation. Recall that a situation is characterized by the sequence of actions that led to it. While the world possibly exhibits the very same state in different situations,[6] the world is in a unique state in each situation. A function denoted by $\mathit{State}(s)$ shall relate situations s to the corresponding states, which are reified collections of fluents.

Modeling entire states as terms allows the use of variables to express mere partial information about a situation. The following, for instance, is a correct incomplete account of the initial situation S_0 in our mystery story (c.f. (3)):

$$\begin{aligned} \exists z\ [\ & \mathit{State}(S_0) = \mathit{Alive}(\mathit{Nephew}) \circ \mathit{Alive}(\mathit{Aunt}) \circ z \\ & \wedge \forall z'.\ z \neq \mathit{Poisoned}(\mathit{Tea}) \circ z'\ ] \end{aligned} \tag{12}$$

[6] If, for example, the tea was already poisoned initially, then the state of the world prior to and after $\mathit{Mix}(\mathit{Nephew}, \mathit{Milk}, \mathit{Tea})$ would have been the same—in terms of which of the two liquids are poisoned and who of our protagonists is alive.

That is to say, of the initial state it is known that both $\mathit{Alive}(\mathit{Nephew})$ and $\mathit{Alive}(\mathit{Aunt})$ are true and that possibly some other facts z hold, too—with the restriction that z must not include $\mathit{Poisoned}(\mathit{Tea})$, of which we know it is false.

The binary function " $\circ$ " needs to inherit from the logical conjunction an important property. Namely, the order is irrelevant in which conjuncts are given. Formally, order ignorance is ensured by stipulating associativity and commutativity, that is, $\forall x, y, z. (x \circ y) \circ z = x \circ (y \circ z)$ and $\forall x, y. x \circ y = y \circ x$. It is convenient to also reify the empty conjunction, a logical tautology, by a constant usually denoted $\emptyset$ and which satisfies $\forall x. x \circ \emptyset = x$. The three equational axioms, jointly abbreviated AC1, in conjunction with the standard axioms of equality entail the equivalence of two state terms whenever they are built up from an identical collection of reified fluents.[7] In addition, denials of equalities, such as in the second part of formula (12), need to be derivable. This requires an extension of the standard assumption of "unique names" for fluents to uniqueness of states, denoted by EUNA (see, e.g., [8, 14]).

The assertion that some fluent f holds (resp. does not hold) in some situation s can now be formalized by $\exists z. \mathit{State}(s) = f \circ z$ (resp. $\forall z. \mathit{State}(s) \neq f \circ z$). This allows to reintroduce the Holds predicate, now, however, not as a primitive notion but as a derived concept:

$$\mathit{Holds}(f, s) \;\equiv\; \exists z.\, \mathit{State}(s) = f \circ z \tag{13}$$

In this way, any Situation Calculus assertion about situations can be directly transferred to a formula of the Fluent Calculus. For instance, the (quite arbitrary) Situation Calculus formula $\exists x. \mathit{Poisoned}(x, S_0) \vee \neg \mathit{Alive}(\mathit{Aunt}, S_0)$ reads $\exists x. \mathit{Holds}(\mathit{Poisoned}(x), S_0) \vee \neg \mathit{Holds}(\mathit{Alive}(\mathit{Aunt}), S_0)$ in the Fluent Calculus. We will use the notation $\mathit{HOLDS}(\Psi)$ to denote the formula that results from transforming a Situation Calculus formula Ψ into the reified version using the Holds predicate.

Knowledge of effects of actions is formalized in terms of specifying how a current state modifies when moving on to a next situation. The universal form of what we call *state update axiom* is

$$\Delta(s) \;\supset\; \Gamma[\mathit{State}(\mathit{Do}(A, s)), \mathit{State}(s)] \tag{14}$$

where $\Delta(s)$ states conditions on s, or rather on the corresponding state, under which the successor state is obtained by modifying the current state according to Γ. Typically, condition $\Delta(s)$ is a compound formula consisting of $\mathit{Holds}(f, s)$ atoms, as defined with the foundational axiom (13). The component Γ defines the way the state in situation s modifies according to the effects of the action under consideration. Actions may initiate and terminate properties. We will discuss the designing of Γ for these two cases in turn.

[7] The reader may wonder why function " $\circ$ " is not expected to be idempotent, i.e., $\forall x.\ x \circ x = x$, which is yet another property of logical conjunction. The (subtle) reason for this is given below.

If an action has a positive effect, then the fluent which becomes true simply needs to be coupled onto the state term. An example is the following axiomatization of the (conditional) effect of mixing a liquid into a second one:

$$\begin{array}{l} \mathit{Holds}(\mathit{Poisoned}(x), s) \wedge \neg \mathit{Holds}(\mathit{Poisoned}(y), s) \supset \\ \qquad \mathit{State}(\mathit{Do}(\mathit{Mix}(p, x, y), s)) = \mathit{State}(s) \circ \mathit{Poisoned}(y) \\ \neg \mathit{Holds}(\mathit{Poisoned}(x), s) \vee \mathit{Holds}(\mathit{Poisoned}(y), s) \supset \\ \qquad \mathit{State}(\mathit{Do}(\mathit{Mix}(p, x, y), s)) = \mathit{State}(s) \end{array} \tag{15}$$

That is to say, if x is poisoned and y is not, then the new state is obtained from the predecessor just by adding the fluent $\mathit{Poisoned}(y)$, else nothing changes at all and so the two states are identical. Notice that neither of the two state update axioms mentions any non-effects.

If we substitute, in the two axioms (15), p, x, and y by Nephew, Milk, and Tea, respectively, and s by S_0, then we can replace the term $\mathit{State}(S_0)$ in both resulting instances by the equal term as given in axiom (12). So doing yields,

$$\begin{array}{l} \exists z \; [\; \mathit{Holds}(\mathit{Poisoned}(\mathit{Milk}), S_0) \wedge \neg \mathit{Holds}(\mathit{Poisoned}(\mathit{Tea}), S_0) \supset \\ \qquad \mathit{State}(\mathit{Do}(\mathit{Mix}(\mathit{Nephew}, \mathit{Milk}, \mathit{Tea}), S_0)) \\ \qquad\quad = \mathit{Alive}(\mathit{Nephew}) \circ \mathit{Alive}(\mathit{Aunt}) \circ z \circ \mathit{Poisoned}(\mathit{Tea}) \\ \quad \wedge \neg \mathit{Holds}(\mathit{Poisoned}(\mathit{Milk}), S_0) \vee \mathit{Holds}(\mathit{Poisoned}(\mathit{Tea}), S_0) \supset \\ \qquad \mathit{State}(\mathit{Do}(\mathit{Mix}(\mathit{Nephew}, \mathit{Milk}, \mathit{Tea}), S_0)) \\ \qquad\quad = \mathit{Alive}(\mathit{Nephew}) \circ \mathit{Alive}(\mathit{Aunt}) \circ z \\ \quad \wedge \forall z'.\, z \neq \mathit{Poisoned}(\mathit{Tea}) \circ z' \;] \end{array}$$

which implies, using the abbreviation $S_1 = \mathit{Do}(\mathit{Mix}(\mathit{Nephew}, \mathit{Milk}, \mathit{Tea}), S_0)$ and the correspondence (13) along with axioms for equality and assertion (12),

$$\begin{array}{l} \exists z \; [\; \mathit{Holds}(\mathit{Poisoned}(\mathit{Milk}), S_0) \supset \\ \qquad \mathit{State}(S_1) = \mathit{Alive}(\mathit{Nephew}) \circ \mathit{Alive}(\mathit{Aunt}) \\ \qquad\qquad \circ \mathit{Poisoned}(\mathit{Milk}) \circ \mathit{Poisoned}(\mathit{Tea}) \circ z \\ \quad \wedge \neg \mathit{Holds}(\mathit{Poisoned}(\mathit{Milk}), S_0) \supset \\ \qquad \mathit{State}(S_1) = \mathit{Alive}(\mathit{Nephew}) \circ \mathit{Alive}(\mathit{Aunt}) \circ z \\ \qquad \wedge \neg \mathit{Holds}(\mathit{Poisoned}(\mathit{Tea}), S_1) \;] \end{array}$$

In this way we have obtained from an incomplete initial specification a still partial description of the successor state, which includes the unaffected fluents $\mathit{Alive}(\mathit{Nephew})$ and $\mathit{Alive}(\mathit{Aunt})$. These properties thus survived the application of the effects axioms without the need to be carried over, one-by-one, by separate application of axioms.

If an action has a negative effect, then the fluent f which becomes false needs to be withdrawn from the current state $\mathit{State}(s)$. The schematic equation $\mathit{State}(\mathit{Do}(A, s)) \circ f = \mathit{State}(s)$ serves this purpose. Incidentally, this scheme is the sole reason for not stipulating that "$\circ$" be idempotent. For otherwise the equation $\mathit{State}(\mathit{Do}(A,s)) \circ f = \mathit{State}(s)$ would be satisfied if $\mathit{State}(\mathit{Do}(A,s))$ contained f. Hence this equation would not guarantee that f becomes false. Vital for our scheme is also to ensure that state terms do not contain any fluent twice or more, i.e.,

$$\forall s,x,z.\ \mathit{State}(s) = x \circ x \circ z \supset x = \emptyset \tag{16}$$

These preparatory remarks lead us to the following axiomatization of the (conditional) effect of drinking:

$$\begin{aligned}
&\mathit{Holds}(\mathit{Alive}(p) \circ \mathit{Poisoned}(x), s) \supset \\
&\qquad \mathit{State}(\mathit{Do}(\mathit{Drink}(p,x),s)) \circ \mathit{Alive}(p) = \mathit{State}(s) \\
&\neg \mathit{Holds}(\mathit{Alive}(p), s) \vee \neg \mathit{Holds}(\mathit{Poisoned}(x), s) \supset \\
&\qquad \mathit{State}(\mathit{Do}(\mathit{Drink}(p,x),s)) = \mathit{State}(s)
\end{aligned} \tag{17}$$

That is to say, if p is alive and x is poisoned, then the new state is obtained from the predecessor just by terminating $\mathit{Alive}(p)$, else nothing changes at all.[8]

Applying the two axioms (17) to what we have derived about the state in situation S_1 yields, setting $S_2 = \mathit{Do}(\mathit{Drink}(\mathit{Aunt}, \mathit{Tea}), S_1)$ and performing straightforward simplifications,

$$\begin{aligned}
\exists z\ [\ &\mathit{Holds}(\mathit{Poisoned}(\mathit{Milk}), S_0) \supset \\
&\qquad \mathit{State}(S_2) \circ \mathit{Alive}(\mathit{Aunt}) = \mathit{Alive}(\mathit{Nephew}) \circ \mathit{Alive}(\mathit{Aunt}) \\
&\qquad\qquad \circ \mathit{Poisoned}(\mathit{Milk}) \circ \mathit{Poisoned}(\mathit{Tea}) \circ z \\
&\wedge \neg \mathit{Holds}(\mathit{Poisoned}(\mathit{Milk}), S_0) \supset \\
&\qquad \mathit{State}(S_2) = \mathit{Alive}(\mathit{Nephew}) \circ \mathit{Alive}(\mathit{Aunt}) \circ z\ ]
\end{aligned}$$

This partial description[9] of the successor state again includes every persistent fluent without having applied separate deduction steps for each. The Fluent Calculus thus provides a solution to both the representational and the inferential aspect of the Frame Problem which is capable of dealing with incomplete knowledge about states.

3 The General Method

Having illustrated the design and use of state update axioms by example, in this section we will present a general, fully mechanic procedure by which is generated

[8] Actions may of course have both positive and negative effects at the same time, in which case the component Γ of a state update axiom combines the schemes for initiating and terminating fluents. This general case is dealt with in Section 3.

[9] which by the way, since $\mathit{State}(S_2) = \mathit{Alive}(\mathit{Nephew}) \circ \mathit{Alive}(\mathit{Aunt}) \circ z$ implies that $\mathit{Holds}(\mathit{Alive}(\mathit{Aunt}), S_2)$, leads directly to the resolution of the murder mystery: Along with the statement of the witness, $\neg \mathit{Holds}(\mathit{Alive}(\mathit{Aunt}), S_2)$, the formula above logically entails the explanation that $\mathit{Holds}(\mathit{Poisoned}(\mathit{Milk}), S_0)$.

a suitable set of state update axioms from a given collection of Situation Calculus effect axioms, like (1) and (2). As indicated in the introduction, we will only consider actions without open effects (c.f. Footnote 2). This is reflected in the assumption that each positive effect specification be of the following form, where A denotes an action and F a fluent:

$$\varepsilon^{+}_{A,F}(\vec{x}, s) \supset F(\vec{y}, Do(A(\vec{x}), s)) \tag{18}$$

Here, ε is a first-order formula whose free variables are among $\vec{x}, s$; and $\vec{y}$ contains only variables from $\vec{x}$. Notice that it is the very last restriction which ensures that the effect specification does not describe what is called an open effect: Except for the situation term, all arguments of the effect F are bound by the action term $A(\vec{x})$. Likewise, negative effect specifications are of the form

$$\varepsilon^{-}_{A,F}(\vec{x}, s) \supset \neg F(\vec{y}, Do(A(\vec{x}), s)) \tag{19}$$

where again ε is a first-order formula whose free variables are among $\vec{x}, s$ and where $\vec{y}$ contains only variables from $\vec{x}$.[10] We assume that a given set $\mathcal{E}$ of effect axioms is consistent in that for all A and F the unique names assumption entails $\neg\exists\vec{x}, s\,[\,\varepsilon^{+}_{A,F}(\vec{x}, s) \wedge \varepsilon^{-}_{A,F}(\vec{x}, s)\,]$.

Fundamental for any attempt to solve the Frame Problem is the assumption that a given set of effect axioms is *complete* in the sense that it specifies all relevant effects of all involved actions.[11] Our concern, therefore, is to design state update axioms for a given set of effect specifications which suitably reflect the completeness assumption. The following instance of scheme (14) is the general form of state update axioms for deterministic actions with only direct effects:

$$\mathit{\Delta}(s) \supset \mathit{State}(Do(A, s)) \circ \vartheta^{-} = \mathit{State}(s) \circ \vartheta^{+}$$

where ϑ^{-} are the negative effects and ϑ^{+} the positive effects, respectively, of action A under condition $\mathit{\Delta}(s)$. The main challenge for the design of these state update axioms is to make sure that condition $\mathit{\Delta}$ is strong enough for the equation in the consequent to be sound. Neither must ϑ^{+} include a fluent that already holds in situation s (for this would contradict the foundational axiom about multiple occurrences, (16)), nor should ϑ^{-} specify a negative effect that is already false in s (for then *EUNA* implies that the equation be false). This is the motivation behind step 1 and 2 of the procedure below. The final and main step 3 reflects the fact that actions with conditional effects require more than one state update axiom, each applying in different contexts:

1. Rewrite to $\varepsilon^{+}_{A,F}(\vec{x}, s) \wedge \neg F(\vec{y}, s) \supset F(\vec{y}, Do(A(\vec{x}), s))$ each positive effect axiom of the form (18).

[10] Our two effect axioms at the beginning of Section 2.1 fit this scheme, namely, by equating $\varepsilon^{+}_{\mathit{Mix},\mathit{Poisoned}}(p, x, y, s)$ with $\mathit{Poisoned}(x, s)$ and $\varepsilon^{-}_{\mathit{Drink},\mathit{Alive}}(p, x, s)$ with $\mathit{Alive}(p, s) \wedge \mathit{Poisoned}(x, s)$.

[11] If actions have additional, indirect effects, then this gives rise to the so-called Ramification Problem; see Section 4.

2. Similarly, rewrite to $\varepsilon^-_{A,F}(\vec{x},s) \wedge F(\vec{y},s) \supset \neg F(\vec{y}, Do(A(\vec{x}),s))$ each negative effect axiom of the form (19).
3. For each action A, let the following $n \geq 0$ axioms be all effect axioms thus rewritten (positive and negative) concerning A:

$$\begin{array}{l} \varepsilon_1(\vec{x},s) \supset F_1(\vec{y}_1, Do(A(\vec{x}),s)),\ \ldots,\ \varepsilon_m(\vec{x},s) \supset F_m(\vec{y}_m, Do(A(\vec{x}),s)) \\ \varepsilon_{m+1}(\vec{x},s) \supset \neg F_{m+1}(\vec{y}_{m+1}, Do(A(\vec{x}),s)),\ \ldots, \\ \qquad\qquad\qquad\qquad \varepsilon_n(\vec{x},s) \supset \neg F_n(\vec{y}_n, Do(A(\vec{x}),s)) \end{array}$$

   Then, for any pair of subsets $\mathcal{I}_+ \subseteq \{1,\ldots,m\}$, $\mathcal{I}_- \subseteq \{m+1,\ldots,n\}$ (including the empty ones) introduce the following state update axiom:

$$\begin{array}{l} \bigwedge_{i \in \mathcal{I}^+ \cup \mathcal{I}^-} HOLDS(\varepsilon_i(\vec{x},s)) \wedge \bigwedge_{j \notin \mathcal{I}^+ \cup \mathcal{I}^-} HOLDS(\neg\varepsilon_j(\vec{x},s)) \\ \qquad \supset State(Do(A(\vec{x}),s)) \circ \vartheta^{\mathcal{I}_-} = State(s) \circ \vartheta^{\mathcal{I}_+} \end{array} \tag{20}$$

   where $\vartheta^{\mathcal{I}_-}$ is the term $F_1 \circ \ldots \circ F_k$ if $\{F_1,\ldots,F_k\} = \{F_i(\vec{y}_i) : i \in \mathcal{I}^-\}$ and, similarly, $\vartheta^{\mathcal{I}_+}$ is the term $F_1 \circ \ldots \circ F_k$ if $\{F_1,\ldots,F_k\} = \{F_i(\vec{y}_i) : i \in \mathcal{I}^+\}$.[12]

Step 3 blindly considers all combinations of positive and negative effects. Some of the state update axiom thus obtained may have inconsistent antecedent, in which case they can be removed. To illustrate the interaction of context-dependent positive and negative effects, let us apply our procedure to these two effect axioms:

$$\begin{array}{r} Loaded(s) \supset Dead(Do(Shoot,s)) \\ true \supset \neg Loaded(Do(Shoot,s)) \end{array}$$

After rewriting according to steps 1 and 2, step 3 produces four state update axioms, viz.

$$\begin{array}{l} \neg\,[\,Holds(Loaded,s) \wedge \neg Holds(Dead,s)\,] \wedge \neg\,[\,true \wedge Holds(Loaded,s)\,] \\ \quad \supset State(Do(Shoot,s)) \circ \emptyset = State(s) \circ \emptyset \\ \neg\,[\,Holds(Loaded,s) \wedge \neg Holds(Dead,s)\,] \wedge true \wedge Holds(Loaded,s) \\ \quad \supset State(Do(Shoot,s)) \circ Loaded = State(s) \circ \emptyset \\ Holds(Loaded,s) \wedge \neg Holds(Dead,s) \wedge \neg\,[\,true \wedge Holds(Loaded,s)\,] \\ \quad \supset State(Do(Shoot,s)) \circ \emptyset = State(s) \circ Dead \\ Holds(Loaded,s) \wedge \neg Holds(Dead,s) \wedge true \wedge Holds(Loaded,s) \\ \quad \supset State(Do(Shoot,s)) \circ Loaded = State(s) \circ Dead \end{array}$$

Logical simplification of the premises of the topmost two axioms yields

$$\begin{array}{r} \neg Holds(Loaded,s) \supset State(Do(Shoot,s)) = State(s) \\ Holds(Dead,s) \wedge Holds(Loaded,s) \supset State(Do(Shoot,s)) \circ Loaded = State(s) \end{array}$$

The third axiom can be abandoned because of an inconsistent antecedent, while the fourth axiom simplifies to

$$\begin{array}{l} Holds(Loaded,s) \wedge \neg Holds(Dead,s) \supset \\ \qquad\qquad State(Do(Shoot,s)) \circ Loaded = State(s) \circ Dead \end{array}$$

[12] Thus $\vartheta^{\mathcal{I}_-}$ contains the negative effects and $\vartheta^{\mathcal{I}_+}$ the positive effects specified in the update axiom. If either set is empty then the respective term is the unit element, $\emptyset$.

(The interested reader may verify that applying the general procedure to our effect axioms (1) and (2) yields four axioms which, after straightforward simplification, turn out to be (15) and (17), respectively.)

The following primary theorem for the Fluent Calculus shows that the resulting set of state update axioms correctly reflects the effect axioms if the fundamental completeness assumption is made.

Theorem 1. *Consider a finite set $\mathcal{E}$ of effect axioms which complies with the assumption of consistency, and let SUA be the set of state update axioms generated from $\mathcal{E}$. Suppose $\mathcal{M}$ is a model of $SUA \cup \{(13),(16)\} \cup EUNA$,*[13] *and consider a fluent term $F(\vec{\tau})$, an action term $A(\vec{\rho})$, and a situation term σ. Then $\mathcal{M} \models Holds(F(\vec{\tau}), Do(A(\vec{\rho}), \sigma))$ iff*

1. *$\mathcal{M} \models \varepsilon^{+}_{A,F}(\vec{\rho}, \sigma)$, for the instance $\varepsilon^{+}_{A,F}(\vec{\rho}, \sigma) \supset F(\vec{\tau}, Do(A(\vec{\rho}), \sigma))$ of some axiom in $\mathcal{E}$;*
2. *or $\mathcal{M} \models Holds(F(\vec{\tau}), \sigma)$ and no instance $\varepsilon^{-}_{A,F}(\vec{\rho}, \sigma) \supset \neg F(\vec{\tau}, Do(A(\vec{\rho}), \sigma))$ of an axiom in $\mathcal{E}$ exists such that $\mathcal{M} \models \varepsilon^{-}_{A,F}(\vec{\rho}, \sigma)$.*

4 Conclusion

We have pursued a new motivation for the Fluent Calculus, namely, as the outcome of applying the principle of reification to successor state axioms. The resulting concept of state update axioms copes with the inferential Frame Problem without losing the solution to the representational aspect. We have shown how, much like in [13], a suitable collection of these axioms can be automatically derived from a complete (wrt. the relevant fluents and actions) set of single effect axioms, provided actions have no open effects. Since state update axioms cover the entire change an action causes in order to solve the inferential aspect of the Frame Problem, their number is, in the worst case, exponentially larger than the number of single effect axioms. This is perfectly acceptable since actions are viewed as having very few effects compared to the overall number of fluents.

Open effects can only be implicitly described in state update axioms. An example is the following axiom (c.f. Footnote 2):

$$
\begin{array}{l}
Bomb(x) \supset \\
\quad \left[\forall f, y \left[\begin{array}{c} f = Destroyed(y) \wedge Holds(Nearby(x,y), s) \wedge \neg Holds(f, s) \\ \equiv \ \exists z.\, w = f \circ z \end{array} \right] \right. \\
\qquad\qquad \left. \supset \ State(Do(Explodes(x), s)) = z \circ w \right]
\end{array}
$$

in which w, the positive effects of the action, is defined rather than explicitly given. It lies in the nature of open effects that a suitable state update axiom can only implicitly describe the required update and so does no longer solve the inferential Frame Problem (though it still covers the representational aspect).

[13] Recall that $EUNA$, the extended unique names assumption, axiomatizes equality and inequality of terms with the function "$\circ$".

The problem of action preconditions has been ignored for the sake of clarity. Their dealing with requires no special treatment in the Fluent Calculus since each Situation Calculus assertion about what holds in a situation corresponds directly to a Fluent Calculus assertion via the fundamental relation (13).

The basic Fluent Calculus as investigated in this paper assumes state update axioms to describe all effects of an action. The solution to the Ramification Problem of [14], and in particular its axiomatization in the Fluent Calculus, furnishes a ready approach for elaborating the ideas developed in the present paper so as to deal with additional, indirect effects of actions.

The version of the Fluent Calculus we arrived at in this paper differs considerably from its roots [7], e.g. in that it exploits the full expressive power of first-order logic. In so doing it is much closer to the variant introduced in [14], but still novel is the notion of state update axioms. In particular the new function $State(s)$ seems to lend more elegance to effect specifications and at the same time emphasizes the relation to the Situation Calculus.

References

1. W. Bibel. A deductive solution for plan generation. *New Generation Computing*, 4:115–132, 1986.
2. W. Bibel. Let's plan it deductively! In M. Pollack, editor, *Proceedings of IJCAI*, pages 1549–1562, Nagoya, Japan, August 1997. Also to appear in *Artif. Intell.*
3. S.-E. Bornscheuer and M. Thielscher. Explicit and implicit indeterminism: Reasoning about uncertain and contradictory specifications of dynamic systems. *J. of Logic Programming*, 31(1–3):119–155, 1997.
4. R. E. Fikes and N. J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artif. Intell.*, 2:189–208, 1971.
5. G. Große, S. Hölldobler, and J. Schneeberger. Linear Deductive Planning. *J. of Logic and Computation*, 6(2):233–262, 1996.
6. A. R. Haas. The case for domain-specific frame axioms. In F. M. Brown, editor, *The Frame Problem in Artificial Intelligence*, pages 343–348, Los Altos, CA, 1987.
7. S. Hölldobler and J. Schneeberger. A new deductive approach to planning. *New Generation Computing*, 8:225–244, 1990.
8. S. Hölldobler and M. Thielscher. Computing change and specificity with equational logic programs. *Annals of Mathematics and Artif. Intell.*, 14(1):99–133, 1995.
9. H. Khalil. Formalizing the effects of actions in first order logic. Master's thesis, Dept. of Computer Science, Darmstadt University of Technology, 1996.
10. R. Kowalski. *Logic for Problem Solving.* Elsevier, 1979.
11. M. Masseron, C. Tollu, and J. Vauzielles. Generating plans in linear logic I. Actions as proofs. *J. of Theoretical Computer Science*, 113:349–370, 1993.
12. J. McCarthy and P. J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intell.*, 4:463–502, 1969.
13. R. 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, 1991.
14. M. Thielscher. Ramification and causality. *Artif. Intell.*, 89(1–2):317–364, 1997.