Compiling Ramification Constraints into Effect Axioms

Javier Pinto
Depto. de Ciencia de la Computación
Pontificia Universidad Católica de Chile
Casilla 306, Santiago 22, Chile
phone: +(56-2) 686-4440
fax: +(56-2) 686-4444
URL: http://lyrcc.ing.puc.cl/jpinto/
jpinto@ing.puc.cl

Subject Category: Knowledge Representation, Action Theories.

Compiling Ramification Constraints into Effect Axioms

Javier Pinto

Abstract

We present a method to derive a solution to the combined frame and ramification problems for certain classes of theories of action written in the situation calculus. The theories of action considered include the causal laws of the domain, in the form of a set of effect axioms, as well as a set of ramification state constraints. The causal laws state the direct effects that actions have on the world, while ramification state constraints allow one to derive indirect effects of actions on the domain.

To solve the combined frame and ramification problems, the causal laws and ramification state constraints are replaced by a set of *successor state axioms*. Given a state of the world, these axioms uniquely determine the truth value of dynamic properties after an action is performed. In this article, we extend previous work by formulating an approach for the mechanical generation of these successor state axioms. We make use of the notions of *implicate* and *support* that have been developed in the context of propositional theories. The approach works for classes of syntactically restricted sets of ramification state constraints.

Keywords: Theories of Action, Ramification Problem, Integrity Constraints, Situation Calculus.

1 Introduction and Motivation

Consider a logical theory of action and change in which two sets of axioms are used to describe the dynamics of the world. First, a set of *effect axioms*, which consists of axioms stating the direct effects that actions have on the world. Second, a set of axioms stating *ramification constraints*; that is, state constraints used to derive indirect effects of actions. For example, an effect axiom may state that after *Chris* puts her glasses on, *Chris* has the glasses on. This may be expressed with an axiom like:

$$(\forall s).\ Poss(putOn(Glasses), s) \supset on(Chris, Glasses, do(putOn(Glasses), s)).$$

This axioms is read as follows: If the action $putOn(Glasses)$ is possible in situation s, then they will be on *Chris* in the situation $do(putOn(Glasses), s))$, that results from performing the action. Furthermore, we may have a ramification constraint that says that, at any time, *Chris*'s glasses are on the night table or she has them on. This may be expressed with the axiom:

$$(\forall s).\ on(NightTable, Glasses, s) \vee on(Chris, Glasses, s). \tag{1}$$

Another ramification constraint may add that objects may not be in two places at the same time:

$$(\forall x, y, z, s).\ \ x \neq z \supset \neg on(x, y, s) \vee \neg on(z, y, s). \tag{2}$$

Thus, if *Chris* puts her glasses on, then we know that, as result, the glasses will no longer be on the night table. This is an indirect effect of the action $putOn(Glasses)$.

We are concerned with the problem of representing change due to the direct or indirect effect of actions on the world. We deal with the *frame problem*, which is the problem of succinctly representing the effects of actions in the domain, avoiding the use of *frame axioms* (McCarthy and Hayes 1969; Reiter 1991). These axioms describe the properties of the world that remain unchanged after an action is performed. We also deal with the ramification problem, which is the problem of representing the indirect effects of actions due to state constraints (Finger 1986; Winslett 1988; Ginsberg and Smith 1988a).

In this work, we extend our previous work in which we derive a solution to the ramification problem for binary ramification state constraints. The starting point of our research is Lin and Reiter's work on the characterization of what can be considered a solution to the combined frame and ramification problems (Lin and Reiter 1994), in the context of Reiter's monotonic solution to the frame problem (Reiter 1991). In Reiter's approach, the main elements of a theory of action are *effect axioms* and the *action precondition axioms*. In this context, he proposed a mechanism to derive a monotonic solution to the frame problem using Schubert's *explanation closure axioms*. The solution is a theory in which the *effect axioms* are replaced with a set of *successor state axioms*. These latter axioms specify necessary and sufficient conditions for change in each of the fluents in the situation calculus language. These conditions are only applicable if the action that motivates them is possible in the situation in which they are performed.

Later, Lin and Reiter (1994) extended this work to theories that not only include the *effect axioms* and *action precondition axioms*, but also a set of *state constraints*. They considered two types of state constraints: *ramification constraints* and *qualification constraints*. With regards to *ramification constraints*, Lin and Reiter's key result is a theorem that establishes conditions under which the set of *effect axioms* along with the *ramification constraints* can be correctly replaced by a set of *successor state axioms*. However, Lin and Reiter do not provide a mechanism to derive this set of *successor state axioms*.

In this paper, we consider the task of mechanically generating the set of *successor state axioms* from a set of *effect axioms* and a set of *ramification state constraints*. This approach works in an interesting class of state constraints. In particular, we make use of the notions of *implicate* and *support* developed in the context of propositional theories, and show how to derive successor state axioms when the dynamic properties of the domain are described using predicates that have an argument of type situation and no other arguments. Furthermore, we argue that the successor state axioms are a solution to the frame and ramification problems, according to Lin and Reiter's criterion, if and only if such a solution exists. The technique is later extended to more general, non propositional, classes of theories.

The problem addressed in this paper is restricted to formalizations of theories of action in which the specifications of direct effects and state constraints, along with preference for minimal change, are sufficient in order to characterize what changes (and what does not change). As discussed later in the paper (section 5), there are problems in which extra information is necessary in order to characterize change; for instance, in the form of causal rules, or influence information. Although, this contribution may be regarded as less general than those that use the notion of causality (discussed in section 5), it deals with the problem of characterizing actions and change without having to appeal to the use of explicit *causal* relations between actions and/or fluents. As discussed in (Pinto 1998a; Pinto 1998b), we are developing an approach in which we do not use explicit causal information in order to derive the indirect effects of an action. Rather, all change can be derived by the use of *effect axioms*, and the use of *triggers* (actions and or the state of the world triggering the occurrence of other action(s)). The work presented in this paper aims at deriving *all* the effect axioms, which express all the immediate effects of an action. Finally, another important aspect of this contribution, is that it provides a characterization that allows for the identification of those axiomatizations that are deterministic, in the sense that the information given is sufficient to completely characterize change due to the execution of actions.

The outline of the paper is as follows. In section 2 we present the version of the situation calculus used in this article, along with basic background axioms that describe the universe of situations. Also, we present Reiter's solution to the frame problem, which specifies a successor state axiom for each dynamic property of the domain. In section 3 we present Lin and Reiter's result regarding the combined solution to the frame and ramification problems using successor state axioms. In section 4, we present a method to derive successor state axioms that would solve the frame and ramification problems according to the criterion presented in section 3. In section 5, we discuss related work. Finally, in section 6 we present our conclusions.

2 The Situation Calculus and Successor State Axioms

In this section we present the version of the situation calculus that will be used in the rest of the paper. This situation calculus is based on the formulation that appears in (Reiter 1993; Lin and Reiter 1994). Also, we present Reiter's original solution to the frame problem.

2.1 The Foundational Situation Calculus

The situation calculus is a logical language designed to represent theories of action and change. The language has its origins in the pioneering work of John McCarthy (1968). The intuition behind the situation calculus language is that there is an *initial situation*, called S_0, and that the world changes from one situation to another when actions are performed. In the situation calculus, the only situations that exist are those that are reached by performing some sequence of actions starting in S_0. As a result, the structure of situations can be seen as a tree rooted at S_0. In this framework, situation dependent properties are called *fluents*, and a *state* corresponds to a set of fluents. Thus, the state of a situation is the set of fluents that hold in that situation. Notice that the branching factor of the tree of situations rooted at S_0 equals the number of distinct actions that exist.

The situation calculus is a sorted second order language with equality. We use the standard operators for negation, conjunction, disjunction, implication and equivalence (*neg*, $\wedge$, $\vee$, $\supset$ and $\equiv$ respectively), and standard universal and existential quantifiers. Scoping is indicated by parentheses or by the dot notation (e.g., "$(\forall x)$." indicates that the scope of the quantifier is the rest of the formula). There are two domain independent sorts $\mathcal{A}$ and $\mathcal{S}$, for action types and situations. We also include a sort $\mathcal{D}$ of domain objects, which may be further subdivided in sub-sorts depending on the particular domain of application. Variables are denoted by lower case letters (with or without subscripts), and constants are denoted by upper case letters (with or without subscripts). Letters a, s, d (A, S, D) are used for variables (constants) of sorts $\mathcal{A}, \mathcal{S}, \mathcal{D}$ respectively. We will have a single 1-place predicate variable ϕ over situations. Unless otherwise stated, other variables and constants will be assumed to be of sort $\mathcal{D}$. Also, when free variables appear in formulas, they are assumed to be universally quantified from the outside.

Strictly speaking, the language is second order (induction is needed in order to define the set of situations). Nevertheless, a standard sorted first order sub-language (with equality) is used to express all other axioms of our theories. If we wanted to eliminate the second order nature of the axiomatization of the situation calculus, we could simply consider "standard models" of the situation calculus axioms (replacing the second order induction axiom with a first order induction schema, as done in first order number theory).

Actions are denoted with function terms from $\mathcal{D}^n \to \mathcal{A}$. We include the special constant S_0 of sort $\mathcal{S}$, the function $do : \mathcal{A} \times \mathcal{S} \to \mathcal{S}$, the relation $< \subseteq \mathcal{S} \times \mathcal{S}$, the relation $Poss \subseteq \mathcal{A} \times \mathcal{S}$, along with other functions and relations introduced later. We use the abbreviation $do([a_1, \dots, a_n], s)$ for the term:

$$do(a_n, do(\dots, do(a_1, s) \dots)).$$

Keep in mind that $[a_1,\dots,a_n]$ is not a term in the language and that it can only appear as a first argument in a *do* function term.

We also include a finite set of predicate constants of different arities. These are the *fluent predicates* and they have a single situation argument, which is always the last argument of the predicate.

The basic axioms for situations are [1]:

$$(\forall\phi).[\phi(S_0)\wedge(\forall s,a)\ (\phi(s)\supset\phi(do(a,s)))]\supset(\forall s)\ \phi(s), \tag{3}$$
$$do(a_1,s_1)=do(a_2,s_2)\supset a_1=a_2, \tag{4}$$
$$S_0\neq do(a,s), \tag{5}$$
$$s_1<do(a,s_2)\equiv Poss(a,s_2)\wedge s_1\leq s_2, \tag{6}$$
$$\neg s<S_0, \tag{7}$$
$$s_1<s_2\supset\neg s_2<s_1. \tag{8}$$

The induction axiom (3) defines the universe of situations as the smallest set that includes S_0 and such that if s is in the set of situations and a is an element of $\mathcal{A}$, then $do(a,s)$ is also in the set of situations. The axiom is necessary to prove properties true in all situations (Reiter 1993).

Axioms (6)-(8) establish the properties of $<$. Intuitively, $s<s'$ is true if and only if s and s' fall in the same branch of the tree of situations and s' is reachable from s by performing possible actions. Also, we will say that a situation s is legal if and only if $S_0\leq s$.

2.2 The Frame Problem

One of the hardest problems encountered in reasoning about actions has been the so called *frame problem*. The problem is understood as one of finding a way to *succinctly* specify the effects, and non-effects, of actions, given the situations in which they are performed. Many solutions to the frame problem are based on non-monotonic logics, e.g., (Shoham 1988; Lifschitz 1987; Lin and Shoham 1992). Also, there has been interest in the development of solutions based on standard monotonic logics. In particular, we appeal to the solution proposed by Reiter (1991), which is based on the work of Haas (1987), Pednault (1988), and Schubert (1990). In this section, we present Reiter's monotonic solution to the frame problem.

A basic theory of action in the situation calculus consists of the following sets of axioms:

- A set Σ_{bd}, which includes
  - The set Σ_b of basic situation calculus axioms (i.e., axioms (3)-(8)).
  - A set Σ_d of situation independent axioms formalizing the static properties of the domain of interest.

[1] We use the abbreviation $s_1\leq s_2$ to mean $s_1<s_2\vee s_1=s_2$. Also, we use the notation $a<b<c$ to mean $a<b\wedge b<c$, and similarly when the expression combines $<$ and $\leq$.

- A set T_{pos} of action precondition axioms. In general, we have a set of necessary conditions for executing an action. For example, we may know that to repair an object it is necessary that the object be broken and that we have glue. This is written as an implication. For instance:

$$Poss(repair(r,x),s) \supset hasglue(r,s) \wedge broken(x,s),$$

  can be interpreted as stating that for agent r to be able to repair object x, she must have glue and x must be broken.

  In this work, we assume that all the necessary conditions for the execution of an action are known. This completeness assumption allows us to write for each action A, an axiom of the form:

$$Poss(A,s) \equiv \Pi_A(s),$$

  where $\Pi_A(s)$ is a simple formula[2]. Therefore, the set T_{pos} is a set of equivalences that establish necessary and sufficient conditions for actions to be executable. The problem of determining the conditions that must be satisfied by a situation for an action to be executable is called the *qualification problem*. This problem is also treated by Lin and Reiter (1994), in the context of Reiter's solution to the frame problem. They show how the action precondition axioms can be derived after the ramification problem has been solved.

- A set T_{S_0} of sentences about S_0. That is, a description of the initial situation. Notice that no observations are considered for situations other than S_0. This limitation can be overcome, however, as described in (Pinto 1998d).

- A set T_{una} of uniqueness of names axioms for actions. For distinct action names a and a':

$$a(x_1,\ldots,x_n) \neq a'(y_1,\ldots,y_m).$$

  Identical actions have identical arguments. Thus, for each function symbol of sort $\mathcal{A}$ in the language, we write:

$$a(x_1,\ldots,x_n) = a(y_1,\ldots,y_n) \supset x_1 = y_1 \wedge \ldots \wedge x_n = y_n.$$

- A set T_{ef} of general effect axioms. Simple effect axioms can be positive or negative. A simple positive effect axiom is of the form[3]:

$$Poss(a,s) \wedge G_f^+(\underline{\mathbf{x}},a,s) \supset f(\underline{\mathbf{x}},do(a,s)).$$

  Whereas simple negative effect axioms are of the form:

$$Poss(a,s) \wedge G_f^-(\underline{\mathbf{x}},a,s) \supset \neg f(\underline{\mathbf{x}},do(a,s)).$$

[2] A simple formula is a first order formula which does not mention the predicate symbols *Poss* or $<$, that mentions a unique situation variable (e.g. s) and no other situation terms, and which does not quantify over variables of sort *situation* ($\mathcal{S}$) (Reiter 1992).

[3] We write $\underline{\mathbf{x}}$ to denote a tuple of domain variables (i.e., variables of sort $\mathcal{D}$).

The formulas G^+ and G^- are simple on the situation variable s.

Now, all the simple positive (negative) effect axioms for each fluent predicate can be put together in a single general positive (negative) effect axiom. A general positive effect axiom for f would be of the form:

$$Poss(a,s) \wedge \gamma_f^+(\underline{\mathbf{x}},a,s) \supset f(\underline{\mathbf{x}},do(a,s)). \tag{9}$$

The corresponding general negative effect axiom is of the form:

$$Poss(a,s) \wedge \gamma_f^-(\underline{\mathbf{x}},a,s) \supset \neg f(\underline{\mathbf{x}},do(a,s)). \tag{10}$$

Thus, $T_{ef} = T_{ef}^+ \cup T_{ef}^-$, where T_{ef}^+ and T_{ef}^- are the general positive and negative effect axioms for each fluent. If there were no positive effect axioms for fluent f, then the formula $\gamma_f^+(\underline{\mathbf{x}},a,s)$ is the atom *False*. Also, if there were no negative effect axioms, then $\gamma_f^-(\underline{\mathbf{x}},a,s)$ would be *False* as well.

In these axioms $\gamma_f^+(\underline{\mathbf{x}},a,s)$ and $\gamma_f^-(\underline{\mathbf{x}},a,s)$ are simple formulas which are used to provide conditions under which an action a produces an effect on a fluent f.

Following Schubert (1990), the assumption that the general (positive or negative) effect axioms characterize all the conditions for a state s under which action a makes $f(\underline{\mathbf{x}},do(a,s))$ true can be formalized with an *explanation closure axiom* of the form:

$$Poss(a,s) \wedge \neg f(\underline{\mathbf{x}},s) \wedge f(\underline{\mathbf{x}},do(a,s)) \supset \gamma_f^+(\underline{\mathbf{x}},a,s).$$

Where $\gamma_f^+(\underline{\mathbf{x}},a,s)$ can be understood as an explanation why f changed from not holding in s to holding in $do(a,s)$, as result of performing action a. Similarly, the assumption that the general effect axioms characterize all the conditions for a state s under which action a makes $f(\underline{\mathbf{x}},do(a,s))$ false is formalized with:

$$Poss(a,s) \wedge f(\underline{\mathbf{x}},s) \wedge \neg f(\underline{\mathbf{x}},do(a,s)) \supset \gamma_f^-(\underline{\mathbf{x}},a,s).$$

Reiter provides an approach to mechanically integrate Schubert's explanation closure axioms with the general effect axioms to derive a set of *successor state axioms*. The resulting successor state axiom for fluent f would be:

$$Poss(a,s) \supset [f(\underline{\mathbf{x}},do(a,s)) \equiv \gamma_f^+(\underline{\mathbf{x}},a,s) \vee f(\underline{\mathbf{x}},s) \wedge \neg\gamma_f^-(\underline{\mathbf{x}},a,s)]. \tag{11}$$

That is, given a situation, if an action is possible we can determine whether the fluent f will hold in the subsequent situation. For this approach to work, it is necessary to have a finite set of effect axioms. Reiter's proposal is to replace the effect axioms with the successor state axioms, yielding a new theory $\Sigma_{bd} \cup T_{pos} \cup T_{S_0} \cup T_{una} \cup T_{ss}(T_{ef})$. Where we write $T_{ss}(T_{ef})$ to denote the successor state axioms that result from utilizing explanation closure axioms and the effect axioms T_{ef}. So, for each fluent predicate in the language, we have a successor state axiom. Notice that if for some fluent f there were no effect axioms, the resulting successor state axioms would state that:

$$Poss(a,s) \supset [f(\underline{\mathbf{x}},do(a,s)) \equiv f(\underline{\mathbf{x}},s)].$$

Thus, f does not change due to the execution of any action.

Interestingly, Reiter's monotonic solution to the frame problem is equivalent to the solution of Lin and Shoham's (1991) in the absence of state constraints. More precisely, as shown in (Lin and Reiter 1994) and in (Shapiro 1993), the models of $\Sigma_{bd} \cup T_{pos} \cup T_{S_0} \cup T_{una} \cup T_{ss}(T_{ef})$ correspond to the minimal models of $\Sigma_{bd} \cup T_{pos} \cup T_{S_0} \cup T_{una} \cup T_{ef}$ given by Lin and Shoham's minimality criterion, based on minimization of change (Lin and Shoham 1991). The minimality criterion is discussed in more detail in the next section.

Reiter's proposal solves the frame problem by providing an axiomatization that requires a number of axioms proportional to the sum of fluent and action function symbols. As already mentioned, a limitation of this approach is that it does not consider theories that include axioms, like (1) and (2), which may lead to ramifications of the direct effects of actions or to extra qualifications for the execution of the actions. The ramification problem is addressed next.

3 Introducing Ramification Constraints

Lin and Reiter (1994), studied the problem of extending Reiter's approach to the frame problem to theories that include a set of state constraints. In particular, they came up with a set of sufficient conditions under which a set of successor state axioms can be said to correctly solve the combined frame and ramification problem. In this section, we summarize the definitions and results of (Lin and Reiter 1994) that we will use later.

Formally, we extend the theories of action Σ with:

- A finite set $T_{rc}(s)$ of domain dependent state constraints[4], such that each constraint is a simple formula with a unique free variable s. In the future we will write T_{rc} to refer to the universal closure of $T_{rc}(s)$. We also use $T_{rc}(s)$ to refer to the conjunction of all the state constraints applied to the situation term s.

Now, we denote with Σ_{rc} the theories of action extended with state constraints. Thus, a theory of action Σ_{rc} is:

$$\Sigma_{rc} = \Sigma_{bd} \cup T_{pos} \cup T_{S_0} \cup T_{una} \cup T_{ef} \cup T_{rc}. \tag{12}$$

In Lin and Reiter's proposal, a solution to the combined frame and ramification problem for a theory Σ_{rc} is a theory Σ'_{rc}:

$$\Sigma'_{rc} = \Sigma_{bd} \cup T_{pos} \cup T_{S_0} \cup T_{una} \cup T^{rc}_{ss}(T_{ef}, T_{rc}), \tag{13}$$

where $T^{rc}_{ss}(T_{ef}, T_{rc})$ represents a set of successor state axioms derived from the effect axioms and the state constraints. Naturally, not any set $T^{rc}_{ss}(T_{ef}, T_{rc})$ will correspond to a correct solution. In order to assess the correctness of a proposed solution, Lin and Reiter take an alternative non-monotonic solution to the frame problem as a correct solution. According to this criterion, a set of successor state axioms is regarded as

[4]For the purpose of this work, we consider only what Lin and Reiter call *effect-oriented* state constraints, ignoring the *precondition oriented* state constraints.

a solution to the combined frame and ramification problems for a theory Σ'_{rc} if and only if the models of the theory Σ'_{rc} (with a simple extension) are the same as the non-monotonic models of Σ_{rc}. These non-monotonic models are defined using a minimal model semantics. The notion of minimality is based on Lin and Shoham's (1991). Lin and Reiter's key result is a theorem that establishes necessary and sufficient conditions for Σ'_{rc} to be a correct solution to the frame and ramification problems.

Definition 3.1 (Model Preference, Minimality) *Let s, a, a', denote variables of sorts $\mathcal{S}$ and $\mathcal{A}$ (according to our convention). Also, let σ_s and σ_a denote assignment functions from free situation and action variables to objects of the respective sorts. Also, let σ_d denote an assignment function for variables of sort $\mathcal{D}$. A model $\mathcal{M}'$, of a situation calculus theory, is preferred to a model $\mathcal{M}$ (i.e., $\mathcal{M}' < \mathcal{M}$) if and only if the following conditions hold:*

- *$\mathcal{M}'$ and $\mathcal{M}$ share the same domain and they interpret everything the same with the possible exception of the fluent predicates.*
- *For some variable assignment to situations σ_s, $\mathcal{M}'$ and $\mathcal{M}$ are such that:*
  - *For any fluent f, and any assignment σ_d, $\mathcal{M}'$ and $\mathcal{M}$ agree on the interpretation of f with respect to $\sigma_s(s)$. Thus,*

    $$\mathcal{M}', \sigma_s, \sigma_d \models f(\underline{\mathbf{x}}, s)$$

    if and only if:

    $$\mathcal{M}, \sigma_s, \sigma_d \models f(\underline{\mathbf{x}}, s).$$

  - *Let $ab(f(\underline{\mathbf{x}}), a, s)$ be the following abbreviation:*

    $$ab(f(\underline{\mathbf{x}}), a, s) = [f(\underline{\mathbf{x}}, s) \equiv \neg f(\underline{\mathbf{x}}, do(a, s))].$$

    For any assignments σ_a, σ_d, if

    $$\mathcal{M}, \sigma_s, \sigma_a, \sigma_d \models s < do(a, s) \wedge \neg ab(f(\underline{\mathbf{x}}), a, s),$$

    then

    $$\mathcal{M}', \sigma_s, \sigma_a, \sigma_d \models \neg ab(f(\underline{\mathbf{x}}), a, s),$$

  - *There are two assignments σ_a and σ_d such that:*

    $$\mathcal{M}, \sigma_s, \sigma_a, \sigma_d \models s < do(a, s) \wedge ab(f(\underline{\mathbf{x}}), a, s),$$

    but

    $$\mathcal{M}', \sigma_s, \sigma_a, \sigma_d \models \neg ab(f(\underline{\mathbf{x}}), a, s).$$

Finally, $\mathcal{M}$ is minimal if there is no $\mathcal{M}'$ such that $\mathcal{M}' < \mathcal{M}$. For further details, the reader is referred to (Lin and Shoham 1991; Lin and Reiter 1994).

Lin and Reiter's result is formally stated as follows:

Theorem 3.1 *Given a theory Σ_{rc} as in (12), let $T_{ss}^{rc}(T_{ef}, T_{rc})$ be a set of successor state axioms (one for each fluent function symbol) of the form:*

$$Poss(a,s) \supset f(x_1,\ldots x_n, do(a,s)) \equiv [\Psi_f \vee (f(x_1,\ldots x_n,s) \wedge \neg\Psi_{\neg f})],$$

where f is an $(n+1)$-ary fluent predicate symbol, and Ψ_f and $\Psi_{\neg f}$ are simple state formulas whose free variables are among $a,s,x_1,\ldots x_n$, and that satisfy the following:

$$\Sigma_{rc} \models (\forall a,s,x_1,\ldots,x_n).Poss(a,s) \supset [\Psi_f \supset f(x_1,\ldots,x_n,do(a,s))],$$

and

$$\Sigma_{rc} \models (\forall a,s,x_1,\ldots,x_n).Poss(a,s) \supset [\Psi_{\neg f} \supset \neg f(x_1,\ldots,x_n,do(a,s))].$$

Also, let Σ'_{rc} be the set of axioms obtained from Σ_{rc} by replacing T_{ef} and T_{rc} with $T_{ss}^{rc}(T_{ef}, T_{rc})$. Assume that the following conditions hold:

1. $T_{una} \models (\forall a,s,x_1,\ldots,x_n).\neg(\Psi_f \wedge \Psi_{\neg f})$,
2. $\Sigma'_{rc} \cup T_{rc}(S_0) \models (\forall s).S_0 \leq s \supset T_{rc}(s)$.

Given these conditions, an interpretation $\mathcal{M}$ is a model of:

$$\Sigma'_{rc} \cup T_{rc}(S_0) \cup T_{rc}^{-}$$

if and only if $\mathcal{M}$ is a minimal model of Σ_{rc}, where T_{rc}^{-} is the set of state constraints restricted to states reached by performing impossible actions, thus:

$$T_{rc}^{-} = (\forall s).\neg S_0 \leq s \supset T_{rc}(s).$$

The first condition in the theorem:

$$T_{una} \models (\forall a,s,x_1,\ldots,x_n).\neg(\Psi_f \wedge \Psi_{\neg f}),$$

should be understood as a consistency condition. In fact, it is a requirement that no single action have the effect of making a fluent f true in some situation and the effect of making the fluent false in the same situation.

The second condition imposes the requirement that the theory Σ'_{rc} entail the state constraints whenever these are satisfied by the initial situation. Thus, this condition guarantees that the updates performed using the successor state axioms in Σ'_{rc} preserve the consistency of the successor state with respect to the state constraints. Therefore, the state constraints are now implicit in the successor state axioms. The relevance of Lin and Reiter's result is that it provides a criterion to judge whether or not a set T_{ss}^{rc} of successor state axioms constitutes a correct solution to the frame problem. Therefore, we introduce the following:

Definition 3.2 (Solution to the combined frame and ramification problems)
Let Σ_{rc} be a theory of action, as in (12). A theory Σ'_{rc}, as in (13), is a solution to the combined frame and ramification problems for Σ_{rc} if and only if the minimal models of Σ_{rc} are exactly the models of $\Sigma'_{rc} \cup T_{rc}(S_0) \cup T_{rc}^{-}$.

Definition 3.3 (Situation Coincidence) *Let $\mathcal{I}$ and $\mathcal{I}'$ be two interpretations for a situation calculus language that share the same domain. If $||s||$ is a situation in the domain $\mathcal{S}$, then $\mathcal{I}$ and $\mathcal{I}'$ coincide with respect to the situation $||s||$ if all fluent predicates have the same interpretation for $||s||$ in both models.*

Definition 3.4 (Successor State Categoricity) *A theory of action Σ_{rc} is said to be successor state categorical if the following holds: Let $\mathcal{M}$ and $\mathcal{M}'$ be two models of Σ_{rc} that interpret everything the same, with the possible exception of the fluent predicates. Then, if the models coincide with respect to some legal situation $||s||$, then they also coincide with respect to any legal situation that results from performing some action in $||s||$.*

Successor state categoricity means that the actions produce deterministic effects.

Observation 3.1 *The theory $\Sigma_{rc'} \cup T_{rc}(S_0) \cup T_{rc}^-$ is successor state categorical.*

This observation follows from the fact that we have one successor state axiom for each fluent predicate in the domain. These successor state axioms uniquely determine whether or not a fluent holds in a situation given a starting situation.

In the next section, we propose an approach to generate successor state axioms obtained from a set of effect axioms and a set of state constraints. In order to justify the correctness of the approach, we make use of theorem 3.1.

4 State Constraints and Successor State Axioms

In the previous section we outlined Lin and Reiter's results to deal with state constraints and solve the combined frame and ramification problems. In summary, they provide necessary and sufficient conditions that a set of successor state axioms must satisfy in order for it to be considered a solution to the combined frame and ramification problems. The problem of finding a specific set of successor state axioms that would constitute a solution to the combined frame and ramification problems was left open. In this section, we address the problem of automatically generating this set of successor state axioms for some classes of theories.

The approach that we pursue is to replace the ramification constraints with a set of effect axioms. In essence, this means that all the implicit effects (due to ramification constraints) have to be made explicit with new effect axioms. For this approach to work, we must show that no implicit effects are left out of the list of new effect axioms. After all the effect axioms have been obtained, we make the *explanation closure assumption* to derive the final set of *successor state axioms*.

Here, we specify a class of situation calculus theories that include ramification state constraints and propose a methodology to derive successor state axioms for them. The approach we take is based on the use of a partial regression operator (inspired on the regression operator defined by Reiter (1991)):

Definition 4.1 (Partial Regression) . *The* partial regression *operator $\mathcal{R}_{\mathcal{P}}$ from universal formulas (in prenex conjunctive normal form) to formulas is defined recursively* *as follows: Let* φ *and* ψ *be simple formulas whose only situation term is* $do(a,s)$*. Then*

$$\mathcal{R}_P((\forall \underline{\mathbf{x}})\varphi) = (\forall \underline{\mathbf{x}})\mathcal{R}_P(\varphi), \tag{14}$$

$$\mathcal{R}_P(\varphi \wedge \psi) = \mathcal{R}_P(\varphi) \wedge \mathcal{R}_P(\psi), \tag{15}$$

$$\mathcal{R}_P(\varphi \vee \psi) = \mathcal{R}_P(\varphi) \vee \mathcal{R}_P(\psi). \tag{16}$$

If $\mathcal{R}_P$ *is applied to a positive fluent literal of the form* $f(\underline{\mathbf{x}}, do(a,s))$*, then:*

$$\mathcal{R}_P(f(\underline{\mathbf{x}}, do(a,s))) = \gamma_f^+(\underline{\mathbf{x}}, a, s).$$

Here, γ_f^+ *comes from the general positive effect axiom for* f*, as in (9). If* $\mathcal{R}_P$ *is applied to a negative fluent literal of the form* $\neg f(\underline{\mathbf{x}}, do(a,s))$*, then:*

$$\mathcal{R}_P(\neg f(\underline{\mathbf{x}}, do(a,s))) = \gamma_f^-(\underline{\mathbf{x}}, a, s).$$

$\gamma_f^-(\underline{\mathbf{x}}, a, s)$ *comes from the general negative effect axiom for* f*, as in (10). If* $\mathcal{R}_P$ *is applied to a non-fluent literal* λ*, then:*

$$\mathcal{R}_P(\lambda) = \lambda.$$

Let us illustrate the approach we take in order to deal with the combined frame and ramification problem. Let f be a fluent predicate symbol. Also, assume that we are able to determine that:

$$T_{rc} \models (\forall \underline{\mathbf{x}}, s).\ \Phi(\underline{\mathbf{x}}, s) \supset f(\underline{\mathbf{x}}, s),$$

where $\Phi(\underline{\mathbf{x}}, s)$ is a simple formula, which constitutes an explanation for f. If we *partially regress* $\Phi(\underline{\mathbf{x}}, do(a,s))$, to obtain $\mathcal{R}_P(\Phi(\underline{\mathbf{x}}, do(a,s)))$, we can add:

$$Poss(a,s) \wedge \mathcal{R}_P(\Phi(\underline{\mathbf{x}}, do(a,s))) \supset f(\underline{\mathbf{x}}, do(a,s)).$$

as an effect axiom to the theory. This approach requires the computation of the explanation $\Phi(\underline{\mathbf{x}}, s)$, as discussed below.

4.1 Propositional Clauses

In this subsection we present some background material, based on (Kean and Tsiknis 1993), that we need in order to develop our approach to dealing with ramification state constraints. For further details on the material presented here, the reader may consult (Kean and Tsiknis 1990; Kean and Tsiknis 1993).

We deal with propositional clauses. A clause is a disjunction of literals, e.g.:

$$f_1 \vee \neg f_2 \vee \neg f_3, \tag{17}$$

where f_1, f_2, and f_3 are propositional constants. A clause C may be seen as a set of literals. Thus, the clause (17) corresponds to the set $\{f_1, \neg f_2, \neg f_3\}$. We need the following

Definition 4.2 (Fundamental Clause) *A clause is fundamental if it does not contain two complementary literals.*

Definition 4.3 (Subsumption) *A clause C subsumes another clause C' if and only if $C \subseteq C'$.*

Definition 4.4 (Implicate) *If Δ is a set of clauses, the clause C is an implicate of Δ if and only if $\Delta \models C$.*

Definition 4.5 (Prime Implicate) *If Δ is a set of clauses, the clause C is a prime implicate of Δ if and only if C is an implicate of Δ and there is no other implicate C' of Δ such that $\models C' \supset C$.*

As notation, if Δ is a set of clauses, then $PI(\Delta)$ will denote the set of prime implicates of Δ.

Kean and Tsiknis (1990, 1993) present an incremental algorithm (see figure 1) to compute the prime implicates for a set Δ of clauses.

Input: *A set Δ of prime implicates of a set Σ of clauses, and a clause C.*

Output: *The set $\Delta_c \cup \Delta = PI(\Delta \cup \{C\})$*

Step 1.0 *Initialize $\Delta_c = \{C\}$. Delete any $D \in \Delta_c \cup \Delta$ such that there is another $D' \in \Delta_c \cup \Delta$ that subsumes D. If C is deleted, stop.*

Step 2.0 *For each propositional letter x occurring in C:*

- **Step 2.1** *For each $S \in \Delta_c$ and $P \in \Delta$ such that S and P resolve on letter x, add the resolvent between S and P to Δ_c.*
- **Step 2.2** *Delete any $D \in \Delta_c \cup \Delta$ such that there is another $D' \in \Delta_c \cup \Delta$ that subsumes D.*

Figure 1: Algorithm to compute the prime implicates.

Furthermore, we will require the following:

Definition 4.6 (Support, Prime Support) *Let Δ be a set of clauses and l be a literal. A clause S is a support for l if:*

- *$\Delta \models S \vee l$, and*
- *$\Delta \not\models S$.*

Also, a clause S is prime support for a literal l with respect to Δ if S is support for l and S is fundamental, and there is no other support S' that subsumes S.

4.1.1 Propositional Ramification Constraints

Definition 4.7 (RCL) *The* ramification constraints description language *(RCL) of a situation calculus language is the set of fluent predicate symbols, along with all the domain predicate symbols. I.e., the predicate Poss is not part of the RCL.*

Definition 4.8 (Propositional Ramification Constraints)

An RCL is propositional if all domain predicate symbols have arity 0 and the fluent predicates have exactly one argument, and this argument is of type $\mathcal{S}$.

Let Σ_{rcp} denote a theory of action $\Sigma_{rc} = \Sigma_{bd} \cup T_{pos} \cup T_{S_0} \cup T_{una} \cup T_{ef} \cup T_{rc}$, in which:

1. The RCL is propositional.
2. The set of state constraints, $T_{rc}(s)$, contains a finite number of state constraints, each one a formula of the form:

$$[\neg]p_1 \vee \ldots \vee [\neg]p_j \vee [\neg]f_1(s) \vee \ldots \vee [\neg]f_k(s), \tag{18}$$

   The notation $[\neg]$ signifies that the symbol $\neg$ may or may not appear in front of the atom. $p_1, \ldots, p_j$ are 0-ary predicate symbols.

The set $T_{rc}(s)$ can be written as[5]:

$$\begin{aligned}&([\neg]p_1^1 \vee \ldots \vee [\neg]p_{j1}^1 \vee [\neg]f_1^1(s) \vee \ldots \vee [\neg]f_{k1}^1(s)) \wedge \\ &([\neg]p_1^2 \vee \ldots \vee [\neg]p_{j2}^2 \vee [\neg]f_1^2(s) \vee \ldots \vee [\neg]f_{k2}^2(s)) \wedge \\ &\ldots \\ &\ldots\end{aligned} \tag{19}$$

Definition 4.9 ($Prop$) *Let $C(s)$ be a clause of the form:*

$$(\forall s).\ [\neg]p_1 \vee \ldots \vee [\neg]p_n \vee [\neg]f_1(s) \vee \ldots \vee [\neg]f_m(s),$$

then, $Prop(C(s))$ denotes the propositional clause[6]*:*

$$[\neg]p_1 \vee \ldots \vee [\neg]p_n \vee [\neg]f_1 \vee \ldots \vee [\neg]f_m.$$

We also use the inverse operator $Prop^{-1}$, such that $Prop^{-1}(s', Prop(C(s))) = C(s')$.

The set T_{rc} can be *propositionalized*, according to the following:

Definition 4.10 (Propositionalized State Constraints) *Assume that $T_{rc}(s)$ is a set of state constraints as in (19); the set $Prop(T_{rc}(s))$ is the set of propositionalized state constraints obtained from $T_{rc}(s)$ by applying the Prop operator to each one of its clauses. Thus, $Prop(T_{rc}(s))$ is:*

$$\begin{aligned}&([\neg]p_1^1 \vee \ldots \vee [\neg]p_{j1}^1 \vee [\neg]f_1^1 \vee \ldots \vee [\neg]f_{k1}^1) \wedge \\ &([\neg]p_1^2 \vee \ldots \vee [\neg]p_{j2}^2 \vee [\neg]f_1^2 \vee \ldots \vee [\neg]f_{k2}^2) \wedge \\ &\ldots \\ &\ldots\end{aligned}$$

[5] Recall that T_{rc} is the universal closure of $T_{rc}(s)$

[6] We must keep track of which letters correspond to fluent predicates and which don't.

Now, we make the following:

Observation 4.1 *If a clause*

$$C = [\neg]p_1 \vee \ldots \vee [\neg]p_n \vee [\neg]f_1 \vee \ldots \vee [\neg]f_m$$

is an implicate of $Prop(T_{rc}(s))$, then:

$$T_{rc} \models (\forall s)\ Prop^{-1}(s, C).$$

PROOF: It is straightforward to see that this must be true. Indeed, any resolution step that can be taken in the propositionalized set of state constrains can be taken in the original set of constraints. The observation follows trivially.

□

The following theorem, adapted from (Kean and Tsiknis 1993), provides a mechanism to compute a set of prime supports for a fluent literal:

Theorem 4.1 (Set of Prime Supports) *Suppose $Prop(T_{rc}(s))$ is a set of propositionalized state constraints obtained from a set T_{rc} of state constraints, expressed in some propositional RCL. If φ is a propositionalized fluent literal, then:*

$$PS(\varphi, Prop(T_{rc}(s))) = \{P - \varphi | P \in PI(Prop(T_{rc}(s))), \varphi \in P\},$$

is the set of prime supports for φ with respect to $Prop(T_{rc}(s))$.

Kean and Tsiknis original results refer to support for arbitrary clauses, as opposed to single literals. In our case, we don't need the added flexibility provided by the definition of support for clauses. Each prime support in the set $PS(\varphi, Prop(T_{rc}(s)))$ is of the form:

$$[\neg]p_1 \vee \ldots \vee [\neg]p_j \vee [\neg]f_1 \vee \ldots \vee [\neg]f_k.$$

Definition 4.11 (Global Explanation) *The* global explanation *for a fluent literal φ from a set T_{rc} of propositional ramification state constraints, is defined as:*

$$GE(\varphi, T_{rc}(s)) = \bigvee_{S \in PS(\varphi, Prop(T_{rc}(s)))} \neg S.$$

Now, we require the following:

Lemma 4.1 *Let Σ_{rcp} be a situation calculus theory with a propositional RCL, and set T_{rc} of ramification state constraints. Also, let σ_a be an arbitrary assignment function from free action variables to actions. If φ is a fluent predicate constant, then:*

$$\Sigma_{rcp}, \sigma_a \models (\forall s).\ Poss(a, s) \wedge \mathcal{R}_P(Prop^{-1}(do(a, s), GE(\varphi, T_{rc}(s)))) \supset \varphi(do(a, s)).$$

and

$$\Sigma_{rcp}, \sigma_a \models (\forall s).\ Poss(a, s) \wedge \mathcal{R}_P(Prop^{-1}(do(a, s), GE(\neg\varphi, T_{rc}(s)))) \supset \neg\varphi(do(a, s)).$$

PROOF: From the definition of global explanation, it follows that:

$$T_{rc} \models (\forall s).\ Prop^{-1}(s, GE(\varphi, T_{rc}(s))) \supset \varphi(s). \tag{20}$$

On the other hand, if $\Psi(s)$ is a simple formula, then from the definition of partial regression:

$$\Sigma_{rcp}, \sigma_a \models (\forall s).\ Poss(a,s) \wedge \mathcal{R}_P(\Psi(do(a,s))) \supset \Psi(do(a,s)).$$

Therefore, we obtain:

$$\Sigma_{rcp}, \sigma_a \models (\forall s).\ Poss(a,s) \wedge \mathcal{R}_P(Prop^{-1}(do(a,s), GE(\varphi, T_{rc}(s)))) \supset \\ Prop^{-1}(do(a,s), GE(\varphi, T_{rc}(s))),$$

using (20), we have that:

$$T_{rc} \models (\forall s).\ Prop^{-1}(do(a,s), GE(\varphi, T_{rc}(s))) \supset \varphi(do(a,s)).$$

The result follows.

□

Remember that, in our notation, the set T_{rc} denotes a set of ramification state constraints. As further notation, let $T_{ef}^{rc}(T_{ef}, T_{rc})$ denote the set of effect axioms that contain T_{ef} and, for each predicate fluent symbol ϕ, the new effect axioms:

$$(\forall a,s).\ Poss(a,s) \wedge \mathcal{R}_P(Prop^{-1}(do(a,s), GE(\phi, T_{rc}(s)))) \supset \phi(do(a,s)), \tag{21}$$
$$(\forall a,s).\ Poss(a,s) \wedge \mathcal{R}_P(Prop^{-1}(do(a,s), GE(\neg\phi, T_{rc}(s)))) \supset \neg\phi(do(a,s)). \tag{22}$$

Furthermore, let $T_{ss}(T_{ef}, T_{rc})$ denote the set of successor state axioms obtained by the application of the explanation closure of assumption to the set $T_{ef}^{rc}(T_{ef}, T_{rc})$ of effect axioms.

Our result is the following:

Lemma 4.2 *Let $\Sigma_{rcp} = \Sigma_{bd} \cup T_{pos} \cup T_{S_0} \cup T_{una} \cup T_{ef} \cup T_{rc}$ be a theory of action in the situation calculus with a propositional RCL and set T_{rc} of ramification state constraints. Also, let*

$$EC(\Sigma_{rcp}) = \Sigma_{bd} \cup T_{pos} \cup T_{S_0} \cup T_{una} \cup T_{rc}(S_0) \cup T_{rc}^{-} \cup T_{ss}(T_{rc}, T_{ef}),$$

where $T_{ss}(T_{rc}, T_{ef})$ contains one successor state axiom for each fluent. The successor state axioms in this set are obtained by using the explanation closure assumption applied to the set T_{ef} of initial effect axioms and the new effect axioms, of the form (21) and (22). Each successor state axiom is of the form:

$$Poss(a,s) \supset f(do(a,s)) \equiv [\Psi_f \vee (f(s) \wedge \neg\Psi_{\neg f})], \tag{23}$$

where Ψ_f and $\Psi_{\neg f}$ are simple state formulas with no free variables other than s. Then, if we assume that

$$T_{una} \models (\forall a,s).\ \neg(\Psi_f \wedge \Psi_{\neg f})$$

and

$$EC(\Sigma_{rcp}) \models (\forall s).\, S_0 \leq s \supset T_{rc}(s),$$

a model of $EC(\Sigma_{rcp})$ is a minimal model of Σ_{rcp}. Thus $EC(\Sigma_{rcp})$ is a solution to the combined frame and ramification problems for the theory Σ_{rcp}.

PROOF: Using theorem 3.1, and given the conditions of the theorem, all we need to show is that for each successor state axiom of the form (23) we have that:

$$\Sigma_{rc} \models (\forall a,s).Poss(a,s) \supset [\Psi_f \supset f(do(a,s))],$$

and

$$\Sigma_{rc} \models (\forall a,s).Poss(a,s) \supset [\Psi_{\neg f} \supset \neg f(do(a,s))].$$

This is true because the new effect axioms that are derived in order to compute $T_{ss}(T_{ef}, T_{rc})$ are a consequence of Σ_{rcp} (lemma 4.1).

□

The previous lemma states that the set $EC(\Sigma_{rcp})$ is a solution to the combined frame and ramification problems if and only if:

1. The consistency conditions are met.
2. The state constraints are satisfied by all the states that are reachable from S_0 by performing sequences of possible actions.

This lemma may seem very weak. As a matter of fact, it provides no guarantees that the set $EC(\Sigma_{rcp})$ is indeed a solution to any problem. However, as our next theorem states, the relevance of the set Σ_{rcp} is that, if there is a solution to the combined frame and ramification problems in the form of successor state axioms, then this solution must be a set of axioms equivalent to $EC(\Sigma_{rcp})$.

Theorem 4.2 *Let Σ_{rcp} be a theory of change with propositional RCL, such that:*

$$\Sigma_{rcp} = \Sigma_{bd} \cup T_{pos} \cup T_{S_0} \cup T_{una} \cup T_{ef} \cup T_{rc}.$$

Also, let Σ_x denote the theory:

$$\Sigma_x = \Sigma_{bd} \cup T_{pos} \cup T_{S_0} \cup T_{una} \cup T_{rc}(S_0) \cup T_{rc}^{-} \cup T_{ss},$$

in which T_{ss} denotes some set of successor state axioms. The set Σ_x represents a solution to the combined frame and ramification problems for Σ_{rcp}, if and only if Σ_x is logically equivalent to $EC(\Sigma_{rcp})$.

PROOF: Let us assume that Σ_x is a solution to the combined frame and ramification problems for Σ_{rcp}. Also, assume that Σ_x is not equivalent to $EC(\Sigma_{rcp})$. Without loss of generality, we assume that Σ_{rcp} is complete with respect to S_0, that is, for any fluent f, either $\Sigma_{rcp} \models f(S_0)$ or $\Sigma_{rcp} \models \neg f(S_0)$. Let $\mathcal{M}_x$ be an arbitrary model of Σ_x, and let

$\mathcal{M}'_{rcp}$ be a model of $EC(\Sigma_{rcp})$, with the same domain, and that interprets everything the same, except for the fluent predicates[7]. Clearly Σ_x and $EC(\Sigma_{rcp})$ are successor state categorical. Since $\mathcal{M}'_{rcp}$ and $\mathcal{M}_x$ are models of Σ_{rcp}, both models coincide with regards to S_0. By using induction, it is easy to show that either, both models are the same, or there exists a situation $||s||$ and an action $||a||$, such that both models coincide on $||s||$ but don't coincide on the situation that results from performing $||a||$ on $||s||$. Furthermore, since $\Sigma_x \models \Sigma_{rcp}$, all abnormalities in $EC(\Sigma_{rcp})$ must show up in Σ_x. Therefore, if $\mathcal{M}_x$ coincides with $\mathcal{M}'_{rcp}$ on a situation, then, they coincide in all legal successor situations. Hence, both models must be the same.

$\square$

4.1.2 An Example

In this simple example, an agent has a hat that he can put on or take off. If he puts the hat on, then the hat will be on his head. This agent never takes his hat off unless he is in his bedroom. When he takes his hat off, he puts it inside the wardrobe, unless the wardrobe is locked. If the wardrobe is locked, he puts the hat on the bed. To model this example, we write a theory Σ_{rcp} in a situation calculus language with action constants *putHatOn* and *takeHatOff*, and with fluent predicates *onHead*, *inWardrobe*, *onBed*, and *locked*. All the fluents take a single argument, and the argument is of type situation. The set Σ_d is empty. The precondition axioms in T_{pos} are:

$$Poss(putHatOn, s) \equiv ((inWardrobe(s) \wedge \neg locked(s)) \vee onBed(s)),$$
$$Poss(takeHatOff, s) \equiv onHead(s).$$

Thus, to put the hat on, either the hat must be in the wardrobe, and the wardrobe must not be locked, or the hat must be on the bed.

The effect axioms in T_{ef} are:

$$Poss(a,s) \wedge a = putHatOn \supset onHead(do(a,s)), \tag{24}$$
$$Poss(a,s) \wedge a = takeHatOff \wedge \neg locked(s) \supset inWardrobe(do(a,s)), \tag{25}$$
$$Poss(a,s) \wedge a = takeHatOff \wedge locked(s) \supset onBed(do(a,s)). \tag{26}$$

The first effect axiom says that after putting the hat on, the hat will be on the head. Axiom (25) states that if the hat is taken off when the wardrobe is not locked, then the hat goes in the wardrobe. Axiom (26) says that if the hat is taken off when the wardrobe is locked, then the hat is left on the bed.

The unique names axioms for actions T_{una} are:

$$putHatOn \neq takeHatOff.$$

The initial situation is characterized with T_{S_0}:

$$\neg locked(S_0).$$

[7] We consider *Poss* to be a defined predicate.

The ramification state constraints in the set T_{rc} are:

$$\neg onHead(s) \vee \neg inWardrobe(s), \tag{27}$$
$$\neg onBed(s) \vee \neg inWardrobe(s), \tag{28}$$
$$\neg onHead(s) \vee \neg onBed(s), \tag{29}$$
$$onBed(s) \vee onHead(s) \vee inWardrobe(s). \tag{30}$$

The constraints (27)-(29) state that the hat cannot be in two places at once. Constraint (30) says that the hat must be in one of the three places. We have chosen not to introduce locations and things as objects of discourse. This is done in order to have a propositional RCL. Notice that an axiom like (2) cannot be expressed in a propositional RCL, unless we had a fixed and finite domain.

Now, we will derive the set of successor state axioms $T_{ss}(T_{rc}, T_{ef})$ as in lemma (4.2). First, we need to obtain the prime implicates of $Prop(T_{rc}(s))$. The resulting set $PI(Prop(T_{rc}(s)))$ contains the following propositional clauses:

$$\neg onHead \vee \neg inWardrobe,$$
$$\neg onBed \vee \neg inWardrobe,$$
$$\neg onHead \vee \neg onBed,$$
$$onBed \vee onHead \vee inWardrobe.$$

From where we obtain the prime supports for the different fluent literals in the language:

$$PS(onHead, Prop(T_{rc}(s))) = \{onBed \vee inWardrobe\},$$
$$PS(onBed, Prop(T_{rc}(s))) = \{onHead \vee inWardrobe\},$$
$$PS(inWardrobe, Prop(T_{rc}(s))) = \{onBed \vee onHead\},$$
$$PS(\neg onHead, Prop(T_{rc}(s))) = \{\neg inWardrobe, \neg onBed\},$$
$$PS(\neg onBed, Prop(T_{rc}(s))) = \{\neg onHead, \neg inWardrobe\},$$
$$PS(\neg inWardrobe, Prop(T_{rc}(s))) = \{\neg onBed, \neg onHead\}.$$

From the prime supports for the fluent literals, we obtain the global explanations for each of the propositional fluent literals:

$$GE(onHead, T_{rc}(s)) = \neg onBed \wedge \neg inWardrobe,$$
$$GE(onBed, T_{rc}(s)) = \neg onHead \wedge \neg inWardrobe,$$
$$GE(inWardrobe, T_{rc}(s)) = \neg onBed \wedge \neg onHead,$$
$$GE(\neg onHead, T_{rc}(s)) = inWardrobe \vee onBed,$$
$$GE(\neg onBed, T_{rc}(s)) = onHead \vee inWardrobe,$$
$$GE(\neg inWardrobe, T_{rc}(s)) = onBed \vee onHead.$$

Now, using lemma (4.2), we take each global explanation and derive new effect axioms for each of the fluent literals. To do this, we need to compute the partial regression for the unpropositionalized global explanations. For instance:

$$\begin{aligned}
&\mathcal{R}_P(Prop^{-1}(do(a,s),GE(\neg inWardrobe,T_{rc}(s)))) = \\
&\mathcal{R}_P(onBed(do(a,s)) \vee onHead(do(a,s))) = \\
&(a = takeHatOff \wedge locked(s)) \vee a = putHatOn.
\end{aligned}$$

Therefore, we can derive the effect axiom:

$$\begin{aligned}
Poss(a,s) \wedge ((a = takeHatOff \wedge locked(s)) \vee a = putHatOn) \supset \\
\neg inWardrobe(do(a,s)).
\end{aligned} \tag{31}$$

In a similar way we obtain:

$$\begin{aligned}
&\mathcal{R}_P(Prop^{-1}(do(a,s),GE(\neg onBed,T_{rc}(s)))) = \\
&\mathcal{R}_P(onHead(do(a,s) \vee inWardrobe(do(a,s)))) = \\
&a = putHatOn \vee (a = takeHatOff \wedge \neg locked(s)).
\end{aligned}$$

the new effect axiom for $\neg onBed$ is:

$$\begin{aligned}
Poss(a,s) \wedge (a = putHatOn \vee (a = takeHatOff \wedge \neg locked(s))) \supset \\
\neg onBed(do(a,s)).
\end{aligned} \tag{32}$$

Finally, we also get:

$$\begin{aligned}
&\mathcal{R}_P(Prop^{-1}(do(a,s),GE(\neg inWardrobe,T_{rc}(s)))) = \\
&\mathcal{R}_P(onBed(do(a,s)) \vee onHead(do(a,s))) = \\
&(a = takeHatOff \wedge locked(s)) \vee a = putHatOn.
\end{aligned}$$

This leads to:

$$\begin{aligned}
Poss(a,s) \wedge ((a = takeHatOff \wedge \neg locked(s)) \vee \\
(a = takeHatOff \wedge locked(s))) \supset \neg onHead(do(a,s)),
\end{aligned}$$

which is equivalent to:

$$Poss(a,s) \wedge a = takeHatOff \supset \neg onHead(do(a,s)). \tag{33}$$

Notice that there are no new effect axioms for positive literals. This is because the global explanations for the positive literals only mention negative literals, and there are no effect axioms in T_{ef} for negative literals. The partial regression for these literals is the atom $False$. Therefore, the derived effect axioms are tautologies.

Finally, we apply explanation closure to the effect axioms (24)-(26) and (31)-(33) and obtain the successor state axioms $EC(\Sigma_{rcp})$:

$$\begin{aligned}
&Poss(a,s) \supset \\
&\quad [onHead(do(a,s)) \equiv a = putHatOn \vee onHead(s) \wedge \neg a = takeHatOff],
\end{aligned} \tag{34}$$

$$Poss(a,s) \supset [onBed(do(a,s)) \equiv [a = takeHatOff \wedge locked(s) \vee \\ onBed(s) \wedge \neg(a = putHatOn \vee \\ (a = takeHatOff \wedge \neg locked(s)))]], \tag{35}$$

$$Poss(a,s) \supset [inWardrobe(do(a,s)) \equiv [a = takeHatOff \wedge \neg locked(s) \vee \\ inWardrobe(s) \wedge \neg(a = putHatOn \vee \\ (a = takeHatOff \wedge locked(s)))]]. \tag{36}$$

Now, are axioms (34)-(36) a solution to the combined frame and ramification problems for Σ_{rcp}? The answer is yes. In order to prove that, we need to show that the conditions for the applicability of Lin and Reiter's theorem hold. First, we need to prove that the successor state axioms satisfy the consistency condition. Thus, we have to prove that:

$$T_{una} \models \neg(a = putHatOn \wedge a = takeHatOff),$$
$$T_{una} \models \neg(a = takeHatOff \wedge locked(s) \wedge \\ (a = putHatOn \vee (a = takeHatOff \wedge \neg locked(s)))),$$
$$T_{una} \models \neg(a = takeHatOff \wedge \neg locked(s) \wedge \\ (a = putHatOn \vee (a = takeHatOff \wedge locked(s)))).$$

All these conditions are trivially true.

We also need to show that:

$$EC(\Sigma_{rcp}) \cup T_{rc}(S_0) \models (\forall s).S_0 \leq s \supset T_{rc}(s).$$

The proof involves the use of induction. First, we assume that the constraints hold for an arbitrary situation and prove that executing the actions *putHatOn* or *takeHatOff* lead to a situation in which the constraints are still satisfied. Now, by simple inspection of the successor state axioms (34)-(36), it should be clear that performing either action (*putHatOn* or *takeHatOff*) results in a state that satisfies the state constraints. For instance, if the action *putHatOn* can be performed, the fluent *onHead* will be made to hold in the situation that results from that action. Also, in any situation that results from performing *putHatOn* the fluents *onBed* and *inWardrobe* will both be false. Thus, all constraints in T_{rc} would be satisfied. Similarly, the constraints are also satisfied in any situation that results from performing *takeHatOff*.

4.1.3 Multiple Minimal Models.

Let us modify the example slightly. Replace axioms (24)-(26) with the following set T_{ef} of effect axioms:

$$Poss(a,s) \wedge a = putHatOn \supset onHead(do(a,s)), \tag{37}$$
$$Poss(a,s) \wedge a = takeHatOff \supset \neg onHead(do(a,s)). \tag{38}$$

Since we do not modify the set T_{rc}, the global explanations remain the same. However, the partial regressions change according to the new set T_{ef}. Thus, we obtain:

$$\mathcal{R}_P(onHead(do(a,s))) = a = putHatOn,$$
$$\mathcal{R}_P(\neg onHead(do(a,s))) = a = takeHatOff.$$

The partial regression for the other fluent literals is $False$. We obtain the following new effect axioms:

$$Poss(a,s) \wedge a = putHatOn \supset \neg onBed(do(a,s)).$$
$$Poss(a,s) \wedge a = putHatOn \supset \neg onWardrobe(do(a,s)).$$

From where we obtain the following successor state axioms:

$$Poss(a,s) \supset [onHead(do(a,s)) \equiv a = putHatOn \vee onHead(s) \wedge \neg a = takeHatOff],$$

$$Poss(a,s) \supset [onBed(do(a,s)) \equiv onBed(s) \wedge \neg a = putHatOn],$$

$$Poss(a,s) \supset [inWardrobe(do(a,s)) \equiv inWardrobe(s) \wedge \neg a = putHatOn].$$

By simple inspection, it is fairly clear that these successor state axioms sanction situations that violate the ramification state constraints. Therefore, these axioms do not constitute a solution to the combined frame and ramification problems. This happens because the theory is not successor state categorical. Indeed, the original effect axioms do not uniquely determine what happens after the hat is taken off. Indeed, the hat could be put on the bed or in the wardrobe. Since the successor state axioms are a realization of a minimal change policy, and since it is consistent to believe that, individually, neither *inWardrobe* nor *onBed* must change, they are assumed not to change. However, in order to obey the ramification state constraints, exactly one of them must change. Thus, there are two possible successor states after performing the action *takeHatOff*. Hence, the *successor state axiom* approach to the frame problem, does not work in this case.

4.2 Non-Propositional Ramification Constraints

Previously, we dealt with theories written in situation calculus languages with propositional RCLs. These languages do not allow for a great deal of expressive power. In fact, our initial motivating example contains the ramification state constraint (2) that cannot be expressed as a propositional ramification constraint. Here, we discuss the problem of extending this approach to more general classes of theories of action with ramification state constraints. Thus, let T_{rc} denote a set of finite clauses of the form:

$$[\neg]\rho_1(\underline{\mathbf{x_1}}) \vee \ldots \vee [\neg]\rho_n(\underline{\mathbf{x_n}}) \vee [\neg]\phi_1(\underline{\mathbf{y_1}},s) \vee \ldots \vee [\neg]\phi_m(\underline{\mathbf{y_m}},s), \quad (39)$$

where $\underline{\mathbf{x_1}}, \ldots, \underline{\mathbf{x_n}}, \underline{\mathbf{y_1}}, \ldots \underline{\mathbf{y_m}}$ are tuples of domain variables and constants. The tuples are not necessarily disjoint; s is a situation variable. Also, we assume an implicit universal quantifier (with maximal scope) for all free variables in the clauses. The ρ_i symbols denote non-fluent predicates, while the ϕ_i symbols denote fluent predicates.

In this subsection, we generalize the method to generate successor state axioms, described previously, to theories in which the state constraints are of the more general form (39). In order to do so, we redefine the concepts of *subsumption, implicate, prime implicate, prime support* and *global explanation*:

Definition 4.12 (Subsumption) *A clause C is said to subsume another clause C' if:*

$$\Sigma_d \models (\forall C) \supset (\forall C'),$$

where $(\forall C)$ denotes the universal closure of clause C.

Thus, a clause is subsumed by another clause if the domain axioms allow to infer one from the other.

Definition 4.13 (Implicate, Prime Implicate) *Let T be a set of clauses, and C be a clause. C is said to be an implicate of T if and only if $(\forall T) \models (\forall C)$. C is said to be a prime implicate of T if C is an implicate of T and no other implicate C' of T subsumes C.*

Definition 4.14 (Set of Prime Implicates) *Let T be a set of clauses. A set $PI(T)$ is a set of prime implicates of T if it is a minimal set of clauses such that:*

1. *Every clause in $PI(T)$ is a prime implicate of T.*
2. *If $C \in T$, then there is a clause C' in $PI(T)$ such that C' subsumes C.*
3. *If C_1 and C_2 are clauses in $PI(T)$ that can be resolved, then there is a clause C' in $PI(T)$ that subsumes the resolvent between C_1 and C_2.*

For an arbitrary set T of clauses, the set $PI(T)$ might not be finite. As an example, consider the following clauses:

$$c(f(x)) \vee \neg a(x),$$
$$a(f(y)) \vee \neg a(y).$$

It should be obvious, in this case, that the set of prime implicates is infinite. In fact, it will contain:

$$c(f(f(x))) \vee \neg a(x),$$
$$c(f(f(f(x)))) \vee \neg a(x),$$
$$\ldots.$$

The notion of *prime support*, previously defined for propositional ramification constraint languages, can be borrowed for this setting. Before we introduce the definition, we need the following notation. If θ is a substitution, then $eq(\theta)$ denotes a conjunction of equality atoms that result from the substitution. E.g., if θ is $\{x/A, y/B\}$, then $eq(\theta)$ denotes $(x = A \wedge y = B)$.

Definition 4.15 (Prime Support) *Let T be a set of clauses, C be a clause, l and l' be literals, and θ be a substitution. If $(C \vee l')\theta$ belongs to $PI(T)$, and $l\theta = l'\theta$, then $C\theta \vee \neg eq(\theta)$ is a prime support for the literal l.*

Observation 4.2 *The set of prime supports for a literal l from a set T of clauses can be obtained as:*

$$\begin{aligned} &PS(l,T) = \\ &\quad \{C\theta \vee \neg eq(\theta) | \textit{ for some substitution } \theta, (C \vee l')\theta \in PI(T) \textit{ and } l'\theta = l\}. \end{aligned} \tag{40}$$

Definition 4.16 (Global Explanation) *Let l be a literal, and T_{rc} be a set of ramification state constraints. The global explanation for l with respect to T_{rc} is defined as:*

$$GE(l,T_{rc}) = \bigvee_{S \in PS(l,T_{rc})} \neg S. \tag{41}$$

Lemma 4.1 generalizes to this setting as well. Thus:

Observation 4.3 *Let $\Sigma_{rcf} = \Sigma_{bd} \cup T_{pos} \cup T_{S_0} \cup T_{una} \cup T_{ef} \cup T_{rcf}$ be a theory of action, with a set T_{rcf} of state constraints that has a finite set, $PI(T_{rcf})$, of prime implicates. Also, let σ_a be an arbitrary assignment function from free action variables to actions. Finally, let ϕ denote a fluent predicate constant. The following holds:*

$$\Sigma_{rcf}, \sigma_a \models (\forall \underline{\mathbf{x}}, s).\ Poss(a,s) \wedge \mathcal{R}_P(GE(\phi(\underline{\mathbf{x}},s), T_{rcf}(s))) \supset \phi(\underline{\mathbf{x}}, do(a,s)), \tag{42}$$

and

$$\Sigma_{rcf}, \sigma_a \models (\forall \underline{\mathbf{x}}, s).\ Poss(a,s) \wedge \mathcal{R}_P(GE(\neg\phi(\underline{\mathbf{x}},s), T_{rcf}(s))) \supset \neg\phi(\underline{\mathbf{x}}, do(a,s)). \tag{43}$$

Therefore, as before, this lemma sanctions the derivation of new effect axioms from the original set of effect axioms and the state constraints. Also:

Lemma 4.3 *Let Σ_{rcf} be a theory of action as in observation 4.3. Also, let*

$$EC(\Sigma_{rcf}) = \Sigma_{bd} \cup T_{pos} \cup T_{S_0} \cup T_{una} \cup T_{rcf}(S_0) \cup T^{-}_{rcf} \cup T_{ss}(T_{rcf}, T_{ef}),$$

where $T_{ss}(T_{rcf}, T_{ef})$ contains one successor state axiom for each fluent. The successor state axioms in this set are obtained by using the explanation closure assumption applied to the set T_{ef} of initial effect axioms and the new effect axioms, of the form (42) and (43). Each successor state axiom is of the form:

$$Poss(a,s) \supset \phi(\underline{\mathbf{x}}, do(a,s)) \equiv [\Psi_\phi \vee (\phi(\underline{\mathbf{x}},s) \wedge \neg\Psi_{\neg\phi})], \tag{44}$$

where Ψ_ϕ and $\Psi_{\neg\phi}$ are simple formulas. Then, if we assume that

$$T_{una} \models (\forall).\neg(\Psi_\phi \wedge \Psi_{\neg\phi}),$$

and

$$EC(\Sigma_{rcf}) \models (\forall s).\ S_0 \leq s \supset T_{rcf}(s),$$

a model of $EC(\Sigma_{rcf})$ is a minimal model of Σ_{rcf}. Thus $EC(\Sigma_{rcf})$ is a solution to the combined frame and ramification problems for the theory Σ_{rcf}.

Also, theorem 4.2 can be generalized for a theory Σ_{rcf} with an analogous formulation. Therefore, in order to apply the technique described above to a theory of action with a set T_{rc} of ramification state constraints, we need to show that the set of prime implicates of T_{rc} is finite. Afterwards, we need to prove that the conditions for the applicability of Lin and Reiter's results are met.

4.2.1 An Example

Let us consider again the example from the introduction. The theory includes the axioms:

- T_{pos}, action preconditions:

$$Poss(putOn(Glasses), s) \equiv \neg on(Chris, Glasses, s).$$

- T_{ef}, effect axioms:

$$(\forall s).\ Poss(a,s) \wedge a = putOn(q) \wedge p = Chris \wedge q = Glasses \supset on(p,q,do(a,s)).$$

- T_{una}, unique names axioms:

$$Glasses \neq NightTable.$$

- T_{rcf}, state constraints:

$$on(NightTable, Glasses, s) \vee on(Chris, Glasses, s), \tag{45}$$

$$x \neq z \supset \neg on(x,y,s) \vee \neg on(z,y,s). \tag{46}$$

From the set T_{rcf} we can also obtain $PI(T_{rcf})$, which includes:

$$on(NightTable, Glasses, s) \vee on(Chris, Glasses, s), \tag{47}$$

$$x = z \vee \neg on(x,y,s) \vee \neg on(z,y,s). \tag{48}$$

$$x = NightTable \vee \neg on(x, Glasses, s) \vee on(Chris, Glasses, s), \tag{49}$$

$$x = Chris \vee \neg on(x, Glasses, s) \vee on(NightTable, Glasses, s). \tag{50}$$

These axioms are obtained from all possible resolution steps that can be performed between the clauses in T_{rcf}, and eliminating redundant clauses. Any further resolution steps produce clauses subsumed by the existing ones.

Now, we need to obtain the global $GE(on(p,q,s), T_{rc})$ explanation for the literal $on(p,q,s)$. The resulting set is:

$$\begin{aligned}
&p = NightTable \wedge q = Glasses \wedge \neg on(Chris, Glasses, s) \vee \\
&p = Chris \wedge q = Glasses \wedge \neg on(NightTable, Glasses, s) \vee \\
&p = Chris \wedge q = Glasses \wedge x \neq NightTable \wedge on(x, Glasses, s) \vee \\
&p = NightTable \wedge q = Glasses \wedge x \neq Chris \wedge on(x, Glasses, s).
\end{aligned}$$

On the other hand, the explanation for $\neg on(p,q,s)$ is:

$$\begin{array}{l} x \neq p \wedge on(x,q,s) \vee \\ p \neq NightTable \wedge q = Glasses \wedge \neg on(Chris, Glasses, s) \vee \\ p \neq Chris \wedge q = Glasses \wedge \neg on(NightTable, Glasses, s). \end{array}$$

In the same way that we used partial regression for a propositional RCL, we obtain:

$$\mathcal{R}_P(GE(on(p,q,s), T_{rcf})) \equiv p = Chris \wedge q = Glasses \wedge a = putOn(q).$$

Notice that there are no actions that produce a negative *on* literal as effect. From the explanations for $\neg on(p,q,s)$ we obtain:

$$\mathcal{R}_P(GE(\neg on(p,q,s), T_{rcf})) \equiv a = putOn(q) \wedge Chris \neq p \wedge q = Glasses.$$

Therefore, using a similar methodology to the one described for propositional ramification constraints, we can derive the following successor state axiom for *on*:

$$\begin{array}{l} Poss(a,s) \supset [on(p,q,do(a,s)) \equiv \\ \qquad p = Chris \wedge q = Glasses \wedge a = putOn(Glasses) \vee \\ \qquad on(p,q,s) \wedge \neg(Chris \neq p \wedge q = Glasses \wedge a = putOn(Glasses))]. \end{array}$$

4.2.2 Applicability

So far, we have presented a method to derive a set of successor state axioms for a theory of action with a set of ramification constraints expressed as a set of clauses. In order to obtain the set of successor state axioms, we need to generate the set of *prime implicates* for the set T_{rc}. However, as we pointed out, this set need not be finite. If this set is not finite, then we cannot build a finite successor state axiom for each fluent predicate in the language, and the approach does not work. Therefore, it is important to be able to characterize situations in which this approach does work. Once we obtain a set of successor state axioms, we still need to show that they constitute a correct solution to the combined frame and ramification problems. To this end, we can use automatic theorem proving techniques to determine whether or not a theory of action, specified with a set of successor state axioms (without effect axioms, nor ramification constraints), maintains state constraints as invariants. This exact problem is pursued by Bertossi *et al.* (1996), where they show how this proofs can be automatically carried out in an inductive theorem proving system based on rewrite rules (Kapur and Zhang 1988).

We have presented an approach to solve the combined frame and ramification problems for theories that include state constraints expressed as clauses of the form (39). The approach is applicable only when the set of prime implicates, derived from the set of ramification constraints, is finite. If so, then we generate the successor state axioms as was illustrated with the previous example. It would be interesting to characterize classes of axiomatizations for which the set of prime implicates of the ramification constraints is finite. A simple case is specified below.

Definition 4.17 (Monadic RCL, Monadic Theory) *A ramification constraint language is called* monadic *if all the domain predicates take exactly one argument and all the fluent predicates take two arguments. The first argument is of type $\mathcal{D}$ and the second argument is of type $\mathcal{S}$. A theory of action Σ_{rc} is called monadic if the set T_{rc} of state constraints can be expressed as a set of clauses of the form:*

$$[\neg]p_1(x_1) \vee \ldots \vee [\neg]p_n(x_n) \vee [\neg]f_1(y_1,s) \vee \ldots \vee [\neg]f_m(y_m,s),$$

where $p_1,\ldots,p_n$ are monadic predicates (not necessarily distinct), $f_1,\ldots,f_n$ are binary fluent predicates. s is a unique situation variable, $x_1,\ldots,x_n,y_1,\ldots y_m$ are domain variables (not necessarily distinct).

We can show that:

Theorem 4.3 *If Σ_{rc} is a monadic theory of action, with set T_{rc} of state constraints, then the set $PI(T_{rc})$ is finite.*

PROOF: The proof is based on the fact that any clause of the form:

$$r_1(x_1) \vee \ldots \vee r_n(x_n) \vee r_{n+1}(x_{n+1}).$$

in which $r_i = r_{n+1}$ for some $i < n$ is equivalent to the clause:

$$r_1(x_1) \vee \ldots \vee r_n(x_n).$$

Therefore, all the constraints that can be derived will contain no more than k literals, where k is the number of different predicate constants in the RCL. Finally, there will be a finite number of possible constraints than can be built with k predicate symbols and no function symbols. Therefore, one cannot have infinitely many prime implicates.

□

This result can be extended for other types of syntactically restricted sets of ramification constraints. What we need is to ensure that the ramification constraints don't allow for the derivation of infinitely many independent clauses (i.e., clauses that don't subsume one another). As part of our future work, we would like to study theories in which the ramification constraints are expressed as Horn clauses. For example, if the horn clauses are *stratified* (Chandra and Harel 1985; Przymusinski 1987), then the set of prime implicates is finite.

5 Related Work

In the field of databases, there has been a great deal of work involving integrity constraints. The issues that arise in databases regarding integrity constraints are analogous to those arising in knowledge representation research. First, integrity constraints can be *static* or *dynamic*. Static integrity constraints express invariants that must be respected by database states. These static constraints do not deal with the dynamics of change. That is, the constraints do not limit the form in which the database evolve into some state. For instance, a state constraint might say that an employee cannot work in two different projects at the same time. Whether a state satisfies this constraints depends only on the fluent values in that state, and does not depend on other states. On the other hand, a *dynamic* integrity constraint limits the way in which the state of a database may evolve. For example, a dynamic constraint might state that no employee's salary can be reduced. These latter constraint is dynamic, because it specifies restrictions on how the database may evolve over time. To check this type of constraints, one needs to look at the values of fluents in more than one state, and, also, look at the way in which the database evolves from one state to the next.

The ramification problem arises when the dynamics of the world (that we capture with effect axioms) interacts with the state (or *static*) constraints. In our work, we propose to compile state constraints into the dynamics of change. Therefore, we enforce the state constraints by embedding them into the dynamics of change. This is done by adding ramifications as explicit effects of primitive actions.

In database systems, an update might produce a state that violates an integrity constraint. The database system can react in two ways. First, it can reject the update; this means that the integrity constraints are understood as providing further qualifications for the executability of transactions. Second, it can propose further updates in order to restore integrity after the update is processed (Decker 1996; Kakas and Mancarella 1990). This can be accomplished by invoking some external procedure in order to resolve the inconsistency (Wallace 1991), or, as in active databases (Widom and Ceri 1996), by specifying active rules that are triggered when integrity is violated. Our work is related to this second case. We also introduce further change (or updates) in order to ensure that the state constraints are maintained.

Similarly, in Knowledge Representation, it has been recognized that theories of action that incorporate state constraints need to deal with two different, but related problems (Ginsberg and Smith 1988a; Ginsberg and Smith 1988b; Lin and Reiter 1994): Firstly, we have the *qualification problem*, that arises in theories where constraints play the same role as in databases. I.e., actions (or database transactions) are considered impossible when they would lead to an inconsistent state (with respect to the constraints). In this case, state constraints provide information about the *Poss* predicate (in the situation calculus). In our work, we do not deal with the qualification problem and assume that qualification state constraints have been dealt with appropriately. For instance, by compiling them into the axioms about *Poss* (as in (Lin and Reiter 1994)).

Secondly, one needs to deal with the ramification problem. In this article, we have concentrated our attention on the problem of compiling away the ramification constraints into the successor state axioms. Thereby, solving the ramification and frame problems for theories of action. Our contribution extends our previous work (Pinto 1994) on this subject, and is based on Reiter's monotonic solution to the frame problem (Reiter 1991) and on the characterization of a solution to the frame and ramification problems presented in (Lin and Reiter 1994). The solution to the frame and ramification problem is based on a minimal model semantics. The approach is applicable when, for a given initial state, all minimal models coincide in the values of fluents in states that result from applying actions. Our approach, inherits some important limitations of this work. In fact, the approach does not apply to theories that are inherently ambiguous. For instance, theories in which the results of actions are not completely specified, or theories in which actions might have indeterminate effects.

However, recent work on theories of actions and ramifications suggest the existence of other limitations that we have not accounted for. In particular, it has been pointed out that there are theories in which ramification constraints might not be enough to characterize a successor state (given an initial state and some action), but that such a successor state should be obtainable given some information not captured by the effect axioms or the state constraints (McIlraith 1997; Thielscher 1997; McCain and Turner 1995; Lin 1995). For instance, consider a theory with the following ramification state constraint:

$$up(L1,s) \wedge up(L2,s) \supset open(s). \tag{51}$$

which is meant to state that certain spring loaded lid becomes open if its two locks are up. The lid remains closed if only one of the locks is up. Furthermore, it is assumed that the only direct effects that have been explicitly given only specify changes in the state of the locks. This problem, proposed in (Lin 1995), is complicated because the effects derived from this constraint are inherently ambiguous. Indeed, the above constraint can be rewritten, for example, as:

$$up(L1,s) \wedge \neg open(s) \supset \neg up(L2,s).$$

Assume that one starts in a state S where $\neg up(L1,S) \wedge up(L2,S) \wedge \neg open(S)$, then if an action that moves $L1$ to the up position is executed, the resulting state could be such that *open* remains false and $L2$ becomes not up, or the state where $L2$ remains up, and *open* becomes true. Unless more information is added to the theory, there is no way to resolve the ambiguity.

A common perspective in recent work (McIlraith 1997; Thielscher 1997; McCain and Turner 1995; Lin 1995), is that extra *causal* information should be provided. This causal information can be taken to be implicit in the syntax of state constraints. For instance, from this perspective, the directionality of (51) carries with it extra *cause-effect* information. For example, McIlraith (1997) studied the problem of dealing with ramification state constraints in theories utilized for model-based diagnosis. The state constraints considered by McIlraith are similar to (51) in their syntactic form. McIlraith showed examples in which the constraints, written as implications, have a natural reading that is not captured by the logical meaning of the implication. McIlraith interprets the implication connective as "causal or *definitional*," and provides a mechanism to produce successor state axioms for theories in which the state constraints (sets of implications) are stratified.

Alternatively, causal information can be given in a more explicit fashion, for instance, Lin, in (1995), proposes to add explicit *causality* information that can then be used to resolve the ambiguity. In the example, one would state that flipping the lock status only causes a change in the lock position. Lin, in (1996), extends this work to theories in which there are indeterminate effects. One important aspect of Lin's proposal is that he appeals to minimal model semantics to characterize the causal information.

On another proposal, McCain and Turner (1995) present an approach in which *causal laws* play the role of inference rules (captured also as sentences in a conditional modal logic).

On a more ellaborate proposal, Thielscher (1997) advances an approach in which *influence* information is given as an asymmetric binary relation between fluents. Thus, in the lid example, the position of lock 1 may influence *open* but not vice-versa. Thielscher shows that domain constraints and the influence information are enough to automatically generate causal relationships.

The main departure of our work from the work based on *causation* (McIlraith 1997; Thielscher 1997; McCain and Turner 1995; Lin 1995) is that we consider theories of action in which:

1. The dynamics of the domain is specified through effect axioms, which specify conditions under which actions change the values of fluents.

2. State constraints specify conditions that a single state must conform. The presence of these state constraints may force secondary action effects that are not explicit in the effect axioms.

3. There is no explicit causal information to decide how the world changes when the above information (along with minimal change policies) is not enough to characterize successor states.

It is clear that the use of causality allows for the treatment of more general classes of problems, beyond those presented in this article. But, can we deal, within this framework, with problems such as the one posed above? The answer is positive, however, as we reported in (Pinto 1998a; Pinto 1998b), extra information is codified in axioms about *triggers*[8]; namely, specification of conditions under which actions are triggered by other actions or by the state of the world. This approach does not have explicit causal information (unless one reads effect axioms and triggering axioms, in the form of implications, as causality). For instance, in the locks example, one can add a *release* action that is triggered when both locks become open. This *release* action would have as a direct effect the opening of the lid, unless the lid is already open.

6 Conclusion

We have presented a method for the mechanical generation of successor state axioms for theories written in the situation calculus and that contain a set of effect axioms and a set of ramification state constraints. We show how to derive a set of successor state axioms for theories in which the domains are described with a propositional language, and in which the dynamic properties of the domain are represented by fluents that only receive a situation argument. It is well known that the solution to the frame problem based on the use of successor state axioms is not universally applicable. In fact, there are many theories that do not lend themselves to this type of solution. The main limitation is that this approach to the frame problem only works in deterministic worlds. Furthermore, even for deterministic worlds, there are theories for which no successor state axioms can be derived. Our result with respect to propositionally described domains is that if a solution to the frame problem, based on successor state axioms, exists. Then this solution is equivalent to the one generated mechanically.

[8]The formalization of triggers is treated in (Pinto 1998c).

Also, we discussed when a similar approach can be utilized for more general classes of theories of action with ramification state constraints. We argued that the approach also produces correct results for theories in which the domain description languages are monadic (i.e., a single non situation argument for fluents) or when the domain description languages are of certain special syntactic form.

In summary, we propose a mechanical approach to derive an axiomatic solution to the combined frame and ramification problems for several classes of theories of action that include effect axioms and ramification state constraints. This solution requires that we prove that a theory of action keeps as an invariant, the truth of the state constraints. This process can also be automated. In fact, Bertossi *et al.* (1996) present a mechanism for the automatic proof of invariants in the situation calculus. An interesting aspect of this result, is that if there is a solution to the combined frame and ramification problems using successor state axioms, the approach presented here will derive it.

Acknowledgements

I would like to thank the anonymous reviewers for carefully reading the original manuscript and for suggesting modifications that have improved the article. Also, thanks to my colleagues, Leopoldo Bertossi and Marcelo Arenas, for their feedback on this research. This work has been partly funded by the FONDECYT grants # 1950885, # 1971305, and # 1980945.

References

Bertossi, L., J. Pinto, P. Saez, D. Kapur, and M. Subramaniam (1996). Automating Proofs of Integrity Constraints in Situation Calculus. In Z. Pawlak and Z. W. Ras (Eds.), *Proc. of the 9th International Symposium on Methodologies for Intelligent Systems*, Number 1079 in Lectures Notes in Artificial Intelligence, pp. 212–222. Springer-Verlag. ISBN 3-540-61286-6.

Chandra, A. and D. Harel (1985). Horn clause queries and generalizations. *Journal of Logic Programming 2*(1), 1–5.

Decker, H. (1996). An extension of sld by abduction and integrity maintenance for view updating in deductive databases. In *Joint International Conference and Symposium on Logic Programming*.

Finger, J. (1986). *Exploiting Constraints in Design Synthesis*. Ph. D. thesis, Department of Computer Science, Stanford University.

Ginsberg, M. L. and D. E. Smith (1988a). Reasoning about Actions I: A Possible Worlds Approach. *Artificial Intelligence 35*(2), 165–195.

Ginsberg, M. L. and D. E. Smith (1988b). Reasoning about Actions II: The Qualification Problem. *Artificial Intelligence 35*(3), 311–342.

Haas, A. R. (1987). The Case for Domain-Specific Frame Axioms. In F. M. Brown (Ed.), *The Frame Problem in Artificial Intelligence, Proceedings of the 1987 Workshop*, Los Altos, CA, pp. 343–348. Morgan Kaufmann Publishers, Inc.

Kakas, A. and P. Mancarella (1990). Database Updates through Abduction. In *16^{th} VLDB Conference, Brisbane, Australia*, pp. 650–661.

Kapur, D. and H. Zhang (1988). RRL: A Rewrite Rule Laboratory. In *Proc. Ninth International Conference on Automated Deduction (CADE-9), LNCS 310*, Number 310 in LNCS. Springer-Verlag.

Kean, A. and G. Tsiknis (1990). An Incremental Method for Generating Prime Implicants/Implicates. *Journal of Symbolic Computation 9*, 185–206.

Kean, A. and G. Tsiknis (1993). Clause Management Systems. *Computational Intelligence 9*(1), 11–40.

Lifschitz, V. (1987, April). Formal Theories of Action. In F. M. Brown (Ed.), *The Frame Problem in Artificial Intelligence*, Los Altos, California, pp. 35–57. Morgan Kaufmann Publishers, Inc.

Lin, F. (1995). Embracing Causality in Specifying the Indirect Effects of Actions. In *Proc. International Joint Conference on Artificial Intelligence, Montreal*, pp. 1985–1991. Morgan Kaufmann.

Lin, F. (1996). Embracing Causality in Specifying the Indeterminate Effects of Actions. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-96)*.

Lin, F. and R. Reiter (1994). State Constraints Revisited. *Journal of Logic and Computation 4*(5), 655–678. Special Issue on Actions and Processes.

Lin, F. and Y. Shoham (1991). Provably Correct Theories of Action (preliminary report). In *Proceedings of the ninth National Conference on Artificial Intelligence (AAAI-91)*, pp. 349–354.

Lin, F. and Y. Shoham (1992). Provably Correct Theories of Action. Technical report, University of Toronto.

McCain, N. and H. Turner (1995). A Causal Theory of Ramifications and Qualifications. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-95)*.

McCarthy, J. (1968). Programs with Common Sense. In M. Minsky (Ed.), *Semantic Information Processing*, Chapter 7, pp. 403–418. The MIT Press.

McCarthy, J. and P. J. Hayes (1969). Some Philosophical Problems from the Standpoint of Artificial Intelligence. In B. Meltzer and D. Michie (Eds.), *Machine Intelligence* **4**, pp. 463–502. Edinburgh, Scotland: Edinburgh University Press.

McIlraith, S. A. (1997). Representing actions and state constraints in model-based diagnosis. In *Proceedings of the National Conference on Artificial Intelligence (AAAI–97)*, pp. 43–49.

Pednault, E. (1988). Synthesizing Plans that Contain Actions with Context-Dependent Effects. *Computational Intelligence 4*, 356–372.

Pinto, J. (1994, February). *Temporal Reasoning in the Situation Calculus.* Ph. D. thesis, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada. URL = ftp://ftp.cs.toronto.edu/˜cogrobo/jpThesis.ps.Z.

Pinto, J. (1998a, January). Causality in theories of action. In R. Miller and M. Shanahan (Eds.), *Fourth Symposium on Logical Formalizations of Commonsense Reasoning*, Queen Mary and Westfield College, London, U.K. URL = http://www.-ida.liu.se/ext/etai/nj/fcs-98/listing.html.

Pinto, J. (1998b). Causality, indirect effects and triggers (preliminary report). In *Seventh International Workshop on Non-monotonic Reasoning*.

Pinto, J. (1998c). Occurrences and Narratives as Constraints in the Branching Structure of the Situation Calculus. *Journal of Logic and Computation 8*, 777–808.

Pinto, J. (1998d). Using Histories to Model Observations in Theories of Action. In G. Antoniou and M. Truszczynski (Eds.), *Learning and Reasoning with Complex Representations*, Number 1359 in Lectures Notes in Artificial Intelligence, pp. 223–235. Springer-Verlag. ISBN 3-540-64413-X.

Przymusinski, T. C. (1987). On the Declarative Semantics of Deductive Databases and Logic Programs. In J. Minker (Ed.), *Foundations of Deductive Databases and Logic Programming*, Chapter 5, pp. 193–216. Morgan Kaufmann.

Reiter, R. (1991). *The Frame Problem in the Situation Calculus: A Simple Solution (sometimes) and a completeness result for goal regression*, pp. 359–380. Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy. San Diego, CA: Academic Press.

Reiter, R. (1992, July). On Specifying Database Updates. Technical Report KRR-TR-92-3, University of Toronto.

Reiter, R. (1993, December). Proving Properties of States in the Situation Calculus. *Artificial Intelligence 64*(2), 337–351.

Schubert, L. (1990). Monotonic solution of the frame problem in the situation calculus: an efficient method for worlds with fully specified actions. In H. Kyberg, R. Loui, and G. Carlson (Eds.), *Knowledge Representation and Defeasible Reasoning*, pp. 23–67. Kluwer Academic Press.

Shapiro, S. (1993, October). First-order Solutions to the Frame Problem. Master's thesis, University of Toronto.

Shoham, Y. (1988). *Reasoning about Change*. Cambridge, Mass.: The MIT Press.

Thielscher, M. (1997). Ramification and causality. *Artificial Intelligence 89*, 317–364.

Wallace, M. (1991). Compiling Integrity Checking into Update Procedures. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-91)*.

Widom, J. and S. Ceri (1996). *Active Database Systems, Triggers and Rules for Advanced Database Processing*. Morgan Kaufmann Publishers, Inc.

Winslett, M. (1988). Reasoning About Action Using a Possible Models Approach. In *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-88)*, pp. 89–93.