Concurrency and Action Interaction

Javier Pinto
Depto. de Ciencia de la Computación,
Pontificia Universidad Católica de Chile,
Casilla 306, Santiago 6904411, Chile.
Email: **jpinto@acm.org**

9th November 2000

Abstract

In this article we present an approach to axiomatize Theories of Action for domains in which many actions may be performed simultaneously. This problem arises in any realistic application where a single agent may be able to perform more than one action at once (e.g., a robot with two independently controlled arms). More importantly, this problem arises in multi-agent scenarios, where agents may collaborate or interfere with each other.

There are several forms in which actions may interact with each other. First, actions may or may not be executable depending upon which actions are or are not executed in parallel. Second, actions that are successfully performed in parallel may bring about effects that each action by itself would not have (i.e., synergy). Third, the effects of one action may be cancelled when executed in parallel with other actions. Fourth, actions with durations may interact if their execution times overlap.

We present an approach to model worlds in which the first three forms of interaction mentioned above arise (the fourth form of interaction is discussed elsewhere). An important feature of our approach is that existing solutions to the frame problem, originally proposed for worlds in which actions are instantaneous and non-concurrently executed, are still applicable.

1 Introduction

Over the last few decades there has been a great deal of research devoted to the development of solutions to the *frame*, *qualification* and *ramification* problems. Due to the inherent difficulties associated with these problems, many solutions to them have considered idealized worlds in which several simplifying assumptions are made. For instance, it is common to consider that actions are instantaneous, and that they are executed in isolation (i.e., non-concurrently). Even if these assumptions were not explicitly made, certain representational problems were ignored.

In this article, we concentrate on representational problems that arise when one considers scenarios in which many actions may be performed in parallel. The main challenge is to be able to devise a representational approach building on existing solutions to the frame, ramification and qualification problems.

To illustrate the type of problems that may arise when actions are executed concurrently, consider the following simple scenarios:

1. Several dancers stand in a closed circle. Each dancer takes one step to the right along the circle. For this step to be possible, each dancer must move in synchrony with the others. Each individual action is impossible by itself. If one dancer doesn't move, the dancer immediately to its left, cannot move either. This is an instance of the *precondition interaction* problem (Pinto 1994), another facet of the qualification problem.

2. A bowl filled with soup is lifted from its left side and from its right side. If both lifting actions are performed in synchrony, then the soup is not spilled, otherwise, the soup is spilled. This is a form of *effect cancellation*, where the spilling effect of one action is cancelled by the simultaneous execution of another action.

3. At home, if the air conditioner and the refrigerator are turned on at the same time, there is a power overload and the fuse is activated; this doesn't happen if both appliances are turned on at different time points. This is a case of *synergy* between actions: if the actions are performed simultaneously, then an effect is observed, which is not observed if the actions are performed individually.

4. In order to enter the building one has to pull the door while holding a spring loaded latch open. Thus, there is concurrency between actions that have durations, one action needs to be performed while the other is also performed.

Some of these problems have been identified by other researchers in Theories of Action; e.g., (Gelfond, Lifschitz, and Rabinov 1991; Allen and Ferguson 1994; Baral and Gelfond 1997; Karlsson and Gustafsson 1999). A discussion relating the work we present in this article, and other approaches is left for the discussion section.

Our work is developed within the tradition of the Situation Calculus (McCarthy 1963), Reiter's solution to the frame problem (Reiter 1991), and later work on the qualification and ramification problems (Lin and Reiter 1994; Pinto 1999). The approach to the frame problem proposed by Reiter takes a set of direct *effect axioms*, which specify those world properties that must hold as a result of performing actions (perhaps with contextual qualifications). Given all the effect axioms, Reiter proposed a mechanism to derive *successor state axioms*, which specify necessary and sufficient conditions for properties to hold after some action is performed. Reiter's solution has several advantages. First, the resulting theory is monotonic, requiring no special semantics. Second, with the exception of the use of second order induction, the resulting theories are first order. This allows one to apply first order theorem provers with induction to reason with the logical specifications (Bertossi, Pinto, Saez, Kapur, and Subramaniam 1996). Third, this approach to the frame problem has been proven to be correct (Bedrax-Weiss and Bertossi 1995), and has also been justified in terms

of policies of minimal change (Lin and Reiter 1994). Fourth, it has led to the development of the cognitive robotics programming language *GOLOG*, whose semantics is expressed in terms of translations to Situation Calculus theories.

The original approach to solve the frame problem by deriving successor state axioms, proposed by Reiter, presented several important restrictions. In particular, it is applicable to domains in which actions are instantaneous, no concurrency is allowed, change is discrete, etc. However, there has been a great deal of research that has successfully extended the approach for richer domains. For example, integrating continuous change, non-determinism, explicit time, etc. (Pinto 1994; Reiter 1996; Kelley 1996). In this article we focus on the integration of concurrent actions. This problem has been partially addressed (e.g., (Reiter 1996)). However, it has been assumed that action interactions do not arise. In this article we focus our attention on the interactions arising from the concurrent execution of instantaneous actions, as illustrated by examples 1, 2 and 3 above. In (Baier and Pinto 1998), an approach to deal with problems where actions with durations interact is presented.

This article is organized as follows. In section 2 we present the formal foundations of the concurrent Situation Calculus, the general structure of theories of action with concurrency, and succinctly describe Reiter's approach to the frame problem adapted to an ontology of actions extended with a new sort for concurrent actions. In section 3 we present semantic definitions for the notion of *effect interaction* (synergy and cancellation). Based on this definition, we then show that theories of action in which synergy and/or cancellation of effects arise are too weak to properly characterize change due to the simultaneous execution of actions. Later, in section 5 we discuss approaches to solve this problem. In particular, we advocate a solution in which the interacting effects are decomposed along the temporal dimension. This leads to axiomatizations in which *trigger* actions are used as intermediary agents of change. In section 7 we establish the conditions under which the precondition interaction problem arises and present an approach to deal with it. The approach is based upon a solution to the qualification problem developed in the context of non-concurrent actions (Lin and Reiter 1994). Finally, in section 10 we present our conclusions.

2 The Concurrent Situation Calculus

In this section we present the concurrent Situation Calculus, which evolved from the original Situation Calculus proposed by John McCarthy (McCarthy 1963), and from later proposals by Raymond Reiter (Reiter 1991). We also present Reiter's solution to the frame problem in the context of the concurrent Situation Calculus.

2.1 Foundational Axioms for the concurrent Situation Calculus

The work on the concurrent Situation Calculus presented here is based primarily on (Pinto 1994; Reiter 1996). The concurrent Situation Calculus extends the Situation Calculus with a sort for concurrent actions, as described shortly.

The concurrent Situation Calculus is a second order language with sorts $\mathcal{A}$, $\mathcal{C}$, $\mathcal{S}$ and

$\mathcal{D}$ for atomic actions[1], concurrent actions, situations and domain objects. The concurrent actions are sets of primitive actions.

In general, the sort of the variables and constants used in the examples should be obvious from the context. The distinguished symbols include the constant S_0, denoting the *initial situation*, the function *do*, which takes a concurrent action and a situation and yields another situation. The function *do* is central, and is meant to denote the situation that results after a concurrent action is performed in another situation. Thus, we think of the world evolution as being described by a tree rooted at S_0. Each situation corresponds to a node in the tree, and their descendants are reached by performing some specific concurrent action. Figure 2.1, depicts a subtree where, from the initial situation S_0, one gets to S_1 after performing action c_1. Thus, using the function *do*, the situation S_1 is denoted with the term $do(c_1, S_0)$, the term $do(c_2, do(c_1, S_0))$ corresponds to situation S_6, etc.

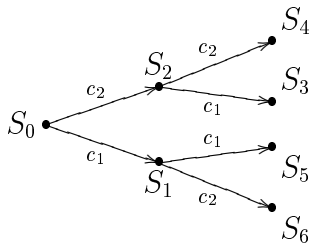

Figure 1: A Situation Calculus subtree.

Dynamic world properties are modeled with predicates that take one argument of type situation. For instance, if the property of being red is a dynamic property of objects in our domain, we could define a predicate $red \subseteq \mathcal{D} \times \mathcal{S}$ to represent it. These predicates are called *fluents*, and, by convention, their last argument is of type situation (and have no other arguments of type situation). Fluents represent properties that are static within a situation, but that may change between situations.

Also, we have the predicate $Poss$, which takes a concurrent action and a situation, and holds when the concurrent action is executable in the situation.

The foundational axioms for the concurrent Situation Calculus are:

$$(\forall\, \varphi).[\varphi(S_0) \wedge (\forall\, s, c)\ (\varphi(s) \supset \varphi(do(c, s)))] \supset (\forall\, s)\ \varphi(s), \tag{1}$$

$$do(c_1, s_1) = do(c_2, s_2) \supset c_1 = c_2 \wedge s_1 = s_2, \tag{2}$$

$$\neg s < S_0, \tag{3}$$

$$s < do(c, s') \equiv Poss(c, s') \wedge s \leq s'. \tag{4}$$

Here and in the rest of the paper, variables that appear free in formulae are assumed to be implicitly universally quantified with maximal scope. Axioms (1)-(4) have the same form as those presented in (Pinto and Reiter 1993; Lin and Reiter 1994) for the non-concurrent

[1]That is, instantaneous primitive actions.

Situation Calculus. The reader is referred to (Reiter 1999) for a detailed discussion on their justification. Concurrent actions are treated as sets of primitive actions. Thus, $\in$ is used as a relation between atomic actions and concurrent actions. We write $a \in c$ to mean that action a is part of the concurrent action c.

In previous works (Pinto 1994), we have considered domain axiomatizations which only include axioms describing preconditions for atomic actions. For non-atomic actions, we used the following:

$$Poss(c, s) \equiv (\forall a)[a \in c \supset Poss(\{a\}, s)]. \tag{5}$$

This axiom embodies the assumption that actions can be performed concurrently, if and only if all the component actions can be performed individually. In general, this is not correct, since there are actions that can be performed concurrently, but not individually and vice-versa. One of the goals of this article is to drop this assumption and describe a method to deal with the *precondition interaction problem* described earlier. This is the subject of section 7. Until then, we keep this assumption. Also, as mentioned in the introduction, we address the *effect interaction problem*. Unfortunately, this problem turned out to be more difficult than originally anticipated. Our approach to deal with this problem was already presented in (Pinto 1998a). A revised and extended version of that work is included in this article

2.2 Domain Axiomatization

A domain axiomatization is divided into several sets of axioms. First, we have a set T_D of axioms that only mention terms of sort $\mathcal{D}$. Also, we have a suitable set of unique names axioms (T_{una}, which we omit) for terms of sorts $\mathcal{A}$ and $\mathcal{D}$. The other sets of axioms are described below.

Action Precondition Axioms

The set T_{prec} of action precondition axioms for atomic actions, each of the form[2]:

$$Poss(\{a_l(\underline{\mathbf{x}})\}, s) \supset \Phi_{a_l}(\underline{\mathbf{x}}, s), \tag{6}$$

where $a_l(\underline{\mathbf{x}})$ is an $\mathcal{A}$ term, and $\Phi_{a_l}(\underline{\mathbf{x}}, s)$ is a formula simple on s. A formula is simple on a situation term s, if it does not mention any situation terms, other than s, and does not quantify over s (Reiter 1991). Thus, preconditions for actions are formulated as necessary conditions for $Poss$. If we assume that there are no *qualification state constraints* (Lin and Reiter 1994), we can apply Clark's completion to obtain a complete characterization of $Poss$.

Example: Let us consider the following variant of the soup bowl example from (Gelfond, Lifschitz, and Rabinov 1991): There is a bowl filled with soup, the bowl can be lifted from either of two sides (left and right). If the bowl is lifted from one side only, then the soup spills. If the bowl is lifted from both sides simultaneously, then there is no spillage. To

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

model this example, we have the action constants $Lift_l$, and $Lift_r$. There are three fluents $lifted_l$, $lifted_r$, and $spilling$.

The action precondition axioms for the actions in this domain are:

$$Poss(\{Lift_l\}, s) \supset \neg lifted_l(s), \tag{7}$$
$$Poss(\{Lift_r\}, s) \supset \neg lifted_r(s), \tag{8}$$

The assumption that these preconditions for performing actions are not only necessary, but also sufficient, leads to transforming each implication (7)-(8) into an equivalence (Clark's completion). This assumption is not valid in the presence of *qualification state constraints*. These constraints provide qualifications on the executability of actions, and are translated into further requirementes for $Poss$. As we discuss in section 7, precondition interaction may arise due to the presence of this type of constraint.

Direct Effect Axioms

The set T_{eff} of effect axioms. These axioms are restricted to specify only the effects that are a direct consequence of the action, regardless of what other actions, if any, are performed concurrently.

Since $Poss$ only takes concurrent actions as first parameter, these axioms have a slightly different syntactic form from Reiter's effect axioms (Reiter 1991). A positive direct effect axiom is of the syntactic form (9), while a negative direct effect axiom is of the form (10).

$$Poss(c, s) \wedge A \in c \wedge G_f^+(\underline{\mathbf{x}}, A, s) \supset f(\underline{\mathbf{x}}, do(c, s)), \tag{9}$$
$$Poss(c, s) \wedge A \in c \wedge G_f^-(\underline{\mathbf{x}}, A, s) \supset \neg f(\underline{\mathbf{x}}, do(c, s)). \tag{10}$$

Here, G_f^+ and G_f^- denote qualifications for the effects of an action on the fluent f.

Now, all the effect axioms, for a given fluent, can be compiled together in one positive and one negative general direct effect axiom of the forms (11) and (12) respectively.

$$Poss(c, s) \wedge \gamma_f^+(\underline{\mathbf{x}}, c, s) \supset f(\underline{\mathbf{x}}, do(c, s)), \tag{11}$$
$$Poss(c, s) \wedge \gamma_f^-(\underline{\mathbf{x}}, c, s) \supset \neg f(\underline{\mathbf{x}}, do(c, s)). \tag{12}$$

If there are no positive (or negative) effect axioms for a fluent f, then the formula $\gamma_f^+(\underline{\mathbf{x}}, c, s)$ (or $\gamma_f^-(\underline{\mathbf{x}}, c, s)$ for the negative case) is the atom $False$.

Example (cont.): For the example, we have:

$$Poss(c, s) \wedge Lift_l \in c \supset lifted_l(do(c, s)),$$
$$Poss(c, s) \wedge Lift_r \in c \supset lifted_r(do(c, s)).$$

The Initial Situation

A set of axioms, T_{S_0}, characterizing the initial situation S_0.

Example (cont.): The initial situation for the example is characterized by:

$$\neg lifted_l(S_0) \wedge \neg lifted_r(S_0) \wedge \neg spilling(S_0).$$

Successor State Axioms

Reiter's monotonic solution to the frame problem leads to the derivation of a set of successor state axioms (one per fluent in the language). In the absence of state constraints (i.e., if T_{rc}, below, is empty), the successor state axioms for a fluent f would be derived from (11)-(12) along with the *explanation closure assumption* (Schubert 1990):

$$Poss(c, s) \supset [f(\underline{\mathbf{x}}, do(c, s)) \equiv \gamma_f^+(\underline{\mathbf{x}}, c, s) \vee f(\underline{\mathbf{x}}, s) \wedge \neg\gamma_f^-(\underline{\mathbf{x}}, c, s)]. \tag{13}$$

These axioms state necessary and sufficient conditions for fluents to change after some possible action is performed. This solution was originally proposed for theories of action without concurrency and without state constraints; this work was extended to theories in which state constraints are present (Lin and Reiter 1994; Pinto 1999). However, this research shows that Reiter's solution is not always applicable. In particular, in (Pinto 1994), it is shown that the approach is not directly applicable when there are constraints such as (14), below.

State Constraints

A set of state constraints divided into two subsets:

T_{qc}**:** Qualification constraints, these are state constraints that are implicit axioms about action preconditions. That is, if an action were to lead to a situation in which the constraint is violated, then the action is regarded as not possible.

T_{rc}**:** Ramification Constraints, these are state constraints that are axioms about indirect effects of actions. That is, if an action were to lead to a situation in which the constraint is violated, then the action has further effects that, in effect, repair the constraint. For example, if we have the constraint

Example (cont.):

$$\begin{aligned} spilling(s) \equiv [&\neg lifted_l(s) \wedge lifted_r(s) \vee \\ &lifted_l(s) \wedge \neg lifted_r(s)] \end{aligned} \tag{14}$$

and some action makes $lifted_l$ true and $lifted_r$ false, then it must also make *spilling* true (as a derived consequence).

As discussed in (Lin and Reiter 1994), it is necessary for the axiom writer to distinguish between the two forms of state constraints. Unfortunately, there is no way to distinguish syntactically the two types of state constraints. Furthermore, when concurrent actions are considered, the situation is even more complex. Indeed, it turns out, some constraints might play both roles when considering concurrently executed actions.

In the following sections we discuss the roles that state constraints play in a theory of action with concurrency. We assume that each state constraint is the universal closure of a simple formula in clausal form.

3 Interacting Effects

Until section 7, we assume that the set T_{qc} of qualification constraints is empty and consider theories of action, Σ_{rc}, composed of axiom sets T_D, T_{una}, T_{prec}, T_{eff}, T_{rc} and T_{S_0}, as in the previous section, in addition to the foundational axioms of section 2.

At an informal level, we say that there is interaction between the effects of actions when there is some concurrent action that either:

- has effects that are not present when the sub-actions are performed individually (synergy), or
- does not have effects that some of its sub-actions have when performed individually (cancellation).

The example with the bowl of soup is a case where there is cancellation of effects. Performing $Lift_l$ together with $Lift_r$ cancels the *spilling* effect that each action would have, if performed by itself.

In order to simplify the following discussion, we assume that fluents are propositional (i.e., they only take a situation argument). However, it is straightforward to extend the analysis below to the more general case.

In what follows, we use $\sigma_{\mathcal{A}}$, $\sigma_{\mathcal{C}}$ and $\sigma_{\mathcal{S}}$ to denote assignment functions from variables of sort $\mathcal{A}$, $\mathcal{C}$ and $\mathcal{S}$ to objects of the corresponding sorts. Before we can formalize the notion of effect interaction, we need the following:

Definition 3.1 (Model Similarity) *Let Σ_{rc} be a theory of action, and $\sigma_{\mathcal{S}}$ be an assignment function from free situation variables to situations. We say that two models, $\mathcal{M}$ and $\mathcal{M}'$ of Σ_{rc} are similar with respect to $\sigma_{\mathcal{S}}(s)$ iff:*

- *$\mathcal{M}$, $\mathcal{M}'$ have the same universe, and differ only in their interpretation of fluents,*
- *for any fluent f:*

$$\mathcal{M}, \sigma_{\mathcal{S}} \models f(s)$$

if and only if

$$\mathcal{M}', \sigma_{\mathcal{S}} \models f(s).$$

Definition 3.2 (Synergy) *Synergistic effects arise in a set* $\mathbf{M}_{\Sigma_{rc}}$ *of models of a theory of action* Σ_{rc}*, with respect to a fluent* f*, when there is a model* $\mathcal{M}$ *in* $\mathbf{M}_{\Sigma_{rc}}$*, and assignment functions* $\sigma_{\mathcal{S}}$ *and* $\sigma_{\mathcal{C}}$ *such that:*

1. *In the model* $\mathcal{M}$*, there is a concurrent action that makes the fluent* f *change in some situation:*

$$\mathcal{M}, \sigma_{\mathcal{S}}, \sigma_{\mathcal{C}} \models Poss(c, s) \wedge ab(f, s, c).$$

   where $ab(f, s, c)$ *is a shorthand for:*

$$f(s) \equiv \neg f(do(c, s)).$$

2. *For any model* $\mathcal{M}'$ *in* $\mathbf{M}_{\Sigma_{rc}}$*, similar to* $\mathcal{M}$ *with respect to* $\sigma_{\mathcal{S}}(s)$*,*

$$\mathcal{M}', \sigma_{\mathcal{S}}, \sigma_{\mathcal{C}} \models ab(f, s, c).$$

3. *For some model* $\mathcal{M}''$ *in* $\mathbf{M}_{\Sigma_{rc}}$ *similar to* $\mathcal{M}$ *with respect to* $\sigma_{\mathcal{S}}(s)$*, there is no assignment* $\sigma_{\mathcal{A}}$*, such that:*

$$\mathcal{M}'', \sigma_{\mathcal{S}}, \sigma_{\mathcal{C}}, \sigma_{\mathcal{A}} \models a \in c \wedge ab(f, s, \{a\}).$$

Intuitively, condition 1 states that there is a model and situation $\sigma_{\mathcal{S}}(s)$ in which f changes when the action $\sigma_{\mathcal{C}}(c)$ is performed. Furthermore, condition 2 states that f must change in the models in $\mathbf{M}_{\Sigma_{rc}}$. I.e., there is no model in $\mathbf{M}_{\Sigma_{rc}}$, similar to the first one with respect to $\sigma_{\mathcal{S}}(s)$, in which f does not change when the concurrent action is performed. Condition 3 states that synergy arises. I.e., the change arises with the concurrent action $\sigma_{\mathcal{C}}(c)$, but that there is no single action in $\sigma_{\mathcal{C}}(c)$ that forces f to change.

Definition 3.3 (Cancellation of Effects) *Cancellation of effects arise in a set* $\mathbf{M}_{\Sigma_{rc}}$ *of models of a theory of action* Σ_{rc}*, with respect to a fluent* f*, when there is a model* $\mathcal{M}$ *in* $\mathbf{M}_{\Sigma_{rc}}$*, and assignment functions* $\sigma_{\mathcal{S}}$*,* $\sigma_{\mathcal{A}}$ *and* $\sigma_{\mathcal{C}}$ *such that:*

1. *In the model* $\mathcal{M}$*, the fluent* f *does not change when some concurrent action is performed:*

$$\mathcal{M}, \sigma_{\mathcal{S}}, \sigma_{\mathcal{C}} \models Poss(c, s) \wedge \neg ab(f, s, c).$$

2. *For any model* $\mathcal{M}'$ *in* $\mathbf{M}_{\Sigma_{rc}}$*, similar to* $\mathcal{M}$ *with respect to* $\sigma_{\mathcal{S}}(s)$*,*

$$\mathcal{M}', \sigma_{\mathcal{S}}, \sigma_{\mathcal{C}} \models \neg ab(f, s, c).$$

3. *For some model* $\mathcal{M}''$ *in* $\mathbf{M}_{\Sigma_{rc}}$ *similar to* $\mathcal{M}$ *with respect to* $\sigma_{\mathcal{S}}(s)$*:*

$$\mathcal{M}'', \sigma_{\mathcal{S}}, \sigma_{\mathcal{C}}, \sigma_{\mathcal{A}} \models a \in c \wedge ab(f, s, \{a\}).$$

Intuitively, condition 1 states that there is a model in $\mathbf{M}_{\Sigma_{rc}}$ and situation $\sigma_{\mathcal{S}}(s)$ in which f does not change when some specific concurrent action $\sigma_{\mathcal{C}}(c)$ is performed. On the other hand 2, ensures that f must not change. I.e., there is no model in $\mathbf{M}_{\Sigma_{rc}}$, similar to the first one with respect to $\sigma_{\mathcal{S}}(s)$, that changes f when the concurrent action is performed. Condition 3 states that an effect has been cancelled. I.e., there is a primitive action, that is included in the concurrent action, mentioned above, after which the change arises.

Example (cont.): As mentioned before, the bowl of soup example is a case of cancellation. Indeed, if Σ_{bowl} is the theory of action for that example:

$$\Sigma_{bowl} \models Poss(\{Lift_l, Lift_r\}, S_0) \wedge Lift_l \in \{Lift_l, Lift_r\},$$
$$\Sigma_{bowl} \not\models ab(spilling, S_0, \{Lift_l\}),$$
$$\Sigma_{bowl} \not\models \neg ab(spilling, S_0, \{Lift_l\}),$$

and

$$\Sigma_{bowl} \models \neg ab(spilling, S_0, \{Lift_l, Lift_r\}).$$

From here, it is straightforward to verify that the conditions for cancellation are present in the set of first order models of Σ_{bowl} (they are also present in the minimal change models of Σ_{bowl}).

Observations

The definitions for synergy and cancellation given above are based on comparing the results of performing a concurrent action c, and the results of performing some primitive sub-action a, where $a \in c$. However, more general definitions can be given by comparing the results of performing a concurrent action c and the results of performing some other concurrent action c', such that $c' \subsetneq c$. It is not hard to rewrite the definitions. But, we kept the definitions presented here for simplicity of exposition.

Notice that, as motivated by many examples, synergy and cancellation are properties that we would like the models of our theories to exhibit. Therefore, after solving the frame problem, we would like the resulting theories to exhibit these properties as well. The result that we show, is that if the minimal models of a theory of action exhibit synergy or cancellation, then the theory must admit more than one minimal model. Thus, there might be spurious minimal models in which the changes that arise are counterintuitive.

4 Minimal Change Policies

In the previous section we presented a formal characterization of the interaction between the effects of actions. Up until now, we have assumed that the theory of action includes axioms describing necessary effects, along with axioms establishing ramification state constraints. It is well known that these axioms are not enough to characterize change, and that some approach to derive non-change should also be considered, in the form of *frame axioms* or

by some other means. A common approach to complete the specification of the theory of action is to incorporate the assumption of minimal change. This minimal change policy can be formalized with the use of circumscription (McCarthy 1986), as in (Baker 1989; Shanahan 1997). Also, in some cases this assumption can be realized by using *explanation closure* (Schubert 1990), which leads to the formulation of successor state axioms (Reiter 1991). This latter approach results in a theory in which the effects of actions are completely characterized, given complete knowledge about the state in which they are performed.

Here, we argue that an approach based on minimal change cannot properly characterize change in theories of the form discussed in the previous section, in which there is interaction between the effects of actions.

The following definition (based on (Lin and Reiter 1994)) characterizes minimal change models:

Definition 4.1 (Minimal Model) *A model $\mathcal{M}$ of a theory of action Σ_{rc} is minimal relative to $\sigma_{\mathcal{S}}(s)$ iff there is no other model $\mathcal{M}'$ of Σ_{rc}, similar to $\mathcal{M}$ with respect to $\sigma_{\mathcal{S}}(s)$, such that:*

- *For any assignment $\sigma_{\mathcal{C}}$, and any fluent f:*

$$\begin{aligned} \textit{if} \quad & \mathcal{M}, \sigma_{\mathcal{S}}, \sigma_{\mathcal{C}} \models Poss(c, s) \wedge \neg ab(f, s, c), \\ \textit{then} \quad & \mathcal{M}', \sigma_{\mathcal{S}}, \sigma_{\mathcal{C}} \models Poss(c, s) \wedge \neg ab(f, s, c). \end{aligned}$$

- *There is an assignment $\sigma_{\mathcal{C}}$, and a fluent f, such that:*

$$\begin{aligned} & \mathcal{M}, \sigma_{\mathcal{S}}, \sigma_{\mathcal{C}} \models Poss(c, s) \wedge ab(f, s, c), \\ \textit{but} \quad & \mathcal{M}', \sigma_{\mathcal{S}}, \sigma_{\mathcal{C}} \models Poss(c, s) \wedge \neg ab(f, s, c). \end{aligned}$$

Also, we say that a model is minimal, if it is a minimal model relative to situation $\sigma_{\mathcal{S}}(s)$, for any assignment $\sigma_{\mathcal{S}}$.

The following definition is inspired in the notion of well-foundedness defined in (Lifschitz 1994):

Definition 4.2 (Well Foundedness) *Let Σ_{rc} be a theory of action and s a situation variable, we say that Σ_{rc} is well founded if for any model $\mathcal{M}'$ and assignment function $\sigma_{\mathcal{S}}$ there exists a model $\mathcal{M}$ similar to $\mathcal{M}'$ with respect to situation $\sigma_{\mathcal{S}}(s)$, which is also minimal relative to $\sigma_{\mathcal{S}}(s)$.*

Theorem 4.1 *Let Σ_{rc} be a well founded theory of action, composed of the foundational axioms of section 2, along with axioms T_D, T_{una}, T_{prec}, T_{eff}, T_{rc} and T_{S_0}, as in section 2.2. If cancellation of effects or synergy arise in the set of minimal models of Σ_{rc}, then Σ_{rc} does not have a unique minimal model.*

Proof Let us assume that Σ_{rc} is a theory of action in which cancellation of effects arise in its set of minimal models (the proof for synergy is similar). Thus, let $\mathcal{M}''$ be a minimal model of Σ_{rc}, and $\sigma_{\mathcal{S}}$, $\sigma_{\mathcal{C}}$ and $\sigma_{\mathcal{A}}$ be as in definition 3.3. In this model, we have:

$$\mathcal{M}'', \sigma_{\mathcal{S}}, \sigma_{\mathcal{C}}, \sigma_{\mathcal{A}} \models Poss(c, s) \wedge a \in c \wedge ab(f, s, \{a\}),$$

and

$$\mathcal{M}'', \sigma_{\mathcal{S}}, \sigma_{\mathcal{C}} \models \neg ab(f, s, c).$$

Now, we can build a model $\mathcal{M}'$ similar to $\mathcal{M}''$ such that for any fluent g:

$$\begin{aligned} & \mathcal{M}', \sigma_{\mathcal{S}}, \sigma_{\mathcal{C}}, \sigma_{\mathcal{A}} \models g(do(\{a\}, s)), \\ \text{iff} \quad & \mathcal{M}'', \sigma_{\mathcal{S}}, \sigma_{\mathcal{C}}, \sigma_{\mathcal{A}} \models g(do(c, s)). \end{aligned}$$

This assignment satisfies the state constraints, otherwise $\mathcal{M}''$ would not satisfy the state constraints either. Given this construction:

$$\mathcal{M}', \sigma_{\mathcal{S}}, \sigma_{\mathcal{C}}, \sigma_{\mathcal{A}} \models \neg ab(f, s, \{a\}).$$

From the well-foundedness of Σ_{rc}, it follows that there will be a minimal model $\mathcal{M}^{m\prime}$ relative to $\sigma_{\mathcal{S}}(s)$ such that:

$$\mathcal{M}^{m\prime}, \sigma_{\mathcal{S}}, \sigma_{\mathcal{C}}, \sigma_{\mathcal{A}} \models \neg ab(f, s, \{a\}).$$

□

An alternative to this model theoretic argument is to consider the necessary existence of ramification constraints, responsible for the interaction between effects. For illustrative purposes, we provide a sketch for this alternative proof. We can differentiate the direct effects of actions from the indirect effects, derivable as ramifications of the direct effects and the state constraints in T_{rc}. Without the set T_{rc} of ramification constraints, there can be no interacting effects. Indeed, it is easy to see that, in this case, the change that is produced by a concurrent action is simply the conjunction of the changes produced by the individual actions. Therefore, if f is the fluent that changes due to synergistic effect interaction (the argument for cancellation is similar), then T_{rc} along with T_D and T_{una}, must entail a formula of the form:

$$l_1(s) \wedge \ldots \wedge l_n(s) \supset f(s),$$

where $l_1(s) \ldots l_n(s)$ denote fluent literals. Also, we have assumed that f changes positively. Therefore, after the concurrent action is performed, all l_i fluent literals are made true, and each l_i is a direct effect of a primitive action in the synergistic concurrent action. From here it follows that, given the definition of synergy, some action has some, but not all, the l_i's as direct effects. Therefore, this action would have a disjunction of literals as a ramification, without having any of its disjuncts as a direct effect. From here it follows that there will be several minimal models.

Example (cont.): According to the previous result, the soup bowl example cannot have a single minimal model. Indeed, one minimal model is the intended one. That is, the model in which $lifted_l$ is true in a situation $do(c, S_0)$ if and only if $Lift_l$ belongs to c. For instance, we would have a model in which:

$$spilling(do(\{Lift_l\}, S_0) \wedge lifted_l(do(\{Lift_l\}, S_0)) \wedge \\ \neg lifted_r(do(\{Lift_l\}, S_0)).$$

However, there is also a minimal model in which:

$$\neg spilling(do(\{Lift_l\}, S_0) \wedge lifted_l(do(\{Lift_l\}, S_0)) \wedge \\ lifted_r(do(\{Lift_l\}, S_0)).$$

Thus, in this second model, lifting the left side of the bowl has the effect of lifting both sides. It is not difficult to verify that there is a minimal model in which this holds. On the other hand,

$$\neg spilling(do(\{Lift_l, Lift_r\}, S_0) \wedge \\ lifted_l(do(\{Lift_l, Lift_r\}, S_0)) \wedge lifted_r(do(\{Lift_l, Lift_r\}, S_0)).$$

is true in all minimal models.

5 Dealing with Interacting Effects

Why is the presence of interacting effects a problem? If one wants to build a theory of actions in the presence of concurrency, it would be inconvenient to specify the direct effects of all possible concurrent actions; in particular, modularity would be lost if every new action that is incorporated in the model required a reformulation of the effect axioms for concurrent actions. Therefore, it would be desirable that these effects be derivable from the direct effects of the atomic actions that are performed as part of the concurrent action. However, as the previous result states, if the effects of actions interact, an axiomatization of the style proposed in section 2.2, along with a minimal model semantics, fails to characterize the intended models.

How can we proceed? In situations like the one illustrated by the soup bowl example, it is very clear which model is the intended one. At this point, there are a few alternatives. Clearly, one can argue that there is not enough information built into the theory in order to single out the *right* model. In fact, one can add *causal* information, and state that lifting one side of the bowl cannot make the other side move, but can cause a spill. This causal information can be used to filter unwanted models. Therefore, in this approach, the theories of action would be extended with explicit *causal axioms*. This extension would need to be accompanied by a more sophisticated (1995) defines an approach based on a minimal model semantics, while Thielscher (1997) provides an approach based on a transition semantics based on causal chains.

A second alternative is to use a *categorization based* approach. For example, we can define a stratification of the fluents. In the example, we could define $lifted_l$ and $lifted_r$ to be in one

stratum, and *spilling* can be placed in another stratum. Then, we can prefer models where persistence in one stratum is preferred to persistence in another. In this case, each stratum would correspond to a separate category. This categorization of fluents is also used to deal with the ramification problem. For example, in (Lifschitz 1990; Brewka and Hertzberg 1993; del Val and Shoham 1993). This also related to Sandewall's approach based on *occlusion* where non occluded properties may be regarded as belonging to one stratum and the rest belonging to a different stratus. It may be the case, as Shanahan suggests (Shanahan 1999), that "state constraints are more useful when there is a clear division of fluents into *primitive* and *derived*.

A third approach, which we pursue here, is based on the observation that the proposed theory fails to characterize the nature of the reality being modeled, and propose an alternative to modeling situations in which the effects of the individual actions interact. In this approach we assume that indirect effects of actions might arise because of the existence of intervening events. For instance, in our running example, if one side of the bowl of soup is lifted, then there is an event, *Spills*, which is triggered by the lifting action. The *Spills* event is triggered in an intermediate state in which only one side of the bowl is lifted, and yet *spilling* is not true. Therefore, modeling the problem in this form implies changes in the way the problem is axiomatized. In particular, because some of the state constraints originally proposed might be violated. This approach has the following advantages:

- In many cases, the nature of the problem that is modeled is such that the intermediate states exist, probably with a very short time span. For instance, in the soup bowl example, it is not unreasonable to consider the existence of an intervening state where *spilling* is not true and only one side of the bowl is lifted. This intervening state will have a very short duration and will trigger the event *Spills*. Therefore, introducing an intermediate state seems to be a more accurate portrayal of the phenomenon.

- Minimal change approaches are still applicable. The problem that was described in the previous section does not arise. In fact, constraints responsible for interaction are replaced by a triggering axiom, (see below) and new effect axioms. In this approach, minimization of change is only meant to deal with the frame problem.

- The conditions on triggered events are expressed in terms of the truth values of fluents, as opposed to the actions that are performed. For example, independently of what actions are involved, if only one side of the bowl is lifted, and *spilling* is not already true, then *Spills* will occur. This is independent of what other actions might be present in the model. Therefore, the axiomatization remains modular.

Natural Actions and Triggers

In order to handle triggers, we extend the language of the Situation Calculus with the notion of *natural actions* (Pinto 1994; Reiter 1996). Natural actions are not regular *agent actions*. Instead, they are actions that arise due to the nature of the world. One characteristic of these actions is that they must occur (i.e., they are triggered) whenever they are possible.

Thus, we need:

$$(\forall\, a, s).\ Poss(\{a\}, s) \wedge natural(a) \supset occurs(a, s). \tag{15}$$

I.e., if a natural action is possible, it must occur. There are several ways in which occurrences can be modeled in the Situation Calculus. One approach is to consider that there is a single branch or narrative line in the tree of situations, such that the actions that occur fall in that branch (Miller and Shanahan 1994; Pinto and Reiter 1993). An alternative is to consider occurrences as constraints that limit the situations that can be considered legal, as in (Pinto 1998b). Here, we use a simplified version of the latter approach. We define the predicate *legal* for situations, such that:

$$\begin{aligned}(\forall\, s).\ legal(s)\ \equiv (s \geq S_0 \wedge \\ (\forall\, s', c, a)\ [s' < s \wedge occurs(a, s') \wedge do(c, s') \leq s \supset a \in c]).\end{aligned} \tag{16}$$

Thus, a situation is considered legal if all the actions that lead from S_0 to it are possible (expressed with the $s \geq S_0$ literal), and if all occurrences within the branch $S_0 - s$ are present in the branch. Therefore, to reason with occurrences, we focus on the situations that are *legal*.

Notice that the predicates *occurs*, *legal*, as well as the predicate *Poss*, are not fluents, since they are not used to describe the *state* of a situation. Rather, they are used to constraint the dynamics of the world, as described by the *legal* subset of situations. Naturally, the state of a situation determines the truth of *occurs* and *legal*, indirectly affecting the dynamics of the world.

Later on, we make use of the notion of situation stability: Situations in which natural actions are possible are not considered stable; this motivates the introduction of the predicate $stable \subseteq \mathcal{S}$ with the definition:

$$stable(s) \equiv \neg(\exists\, a).natural(a) \wedge Poss(\{a\}, s). \tag{17}$$

Furthermore, we introduce the predicate $leads \subseteq \mathcal{C} \times \mathcal{S} \times \mathcal{S}$ with the definition:

$$\begin{aligned}leads(c, s, s') \equiv s < do(c, s) \leq s' \wedge stable(s') \wedge \\ ((\forall\, s_i)\ s < s_i < s' \supset \neg stable(s_i)) \wedge \\ ((\forall\, c_i, s_i, a)\ do(c, s) < do(c_i, s_i) \leq s' \wedge a \in c_i \supset \\ natural(a)).\end{aligned} \tag{18}$$

This latter predicate relates a situation s and a concurrent action c to the *stable* situation that results after all natural actions triggered by c in s are performed. The third conjunct on the right side of the equivalence requires that all intermediate situations be unstable. The last conjunct establishes that the resulting stable situation be reached by performing only natural actions.

Triggers and interacting effects.

As mentioned in the sketch of the proof of theorem 4.1, the interactions between effects arise because of the presence of ramification constraints. Our proposal is based on the hypothesis

that the ramification constraints can be decomposed along the temporal dimension. What this means is that some constraints hide a process that can be modeled at a higher level of detail. For instance, a constraint of the form:

$$l_1(s) \wedge l_2(s) \supset l(s). \tag{19}$$

where l_1, l_2 and l represent some fluent literals, might be used to model a causal relationship between these literals. Since causality cannot be captured with implication alone, problems arise. One possibility for decomposing this type of constraint is to introduce new events as intermediary *causes*. Thus, instead of writing a constraint like (19), we propose to introduce some intermediate natural action A_l in the language, and add the following axioms[3]:

$$natural(A_l), \tag{20}$$
$$Poss(\{A_l\}, s) \equiv l_1(s) \wedge l_2(s) \wedge \neg l(s), \tag{21}$$
$$Poss(c, s) \wedge A_l \in c \supset l(do(\{A_l\}, s)). \tag{22}$$

Under these conditions, the interaction of effects, due to the original state constraints, is converted into a triggered effect. Synergy or cancellation due to state constraints can be handled in this manner in many important cases, as described in the next section.

Example (cont.): In the soup bowl example, we can replace the ramification constraints with the precondition axioms:

$$\begin{aligned} Poss(\{Spills\}, s) \equiv & \neg lifted_l(s) \wedge lifted_r(s) \wedge \neg spilling(s) \vee \\ & lifted_l(s) \wedge \neg lifted_r(s) \wedge \neg spilling(s) \\ Poss(\{Nspills\}, s) \equiv & lifted_l(s) \wedge lifted_r(s) \wedge spilling(s) \vee \\ & \neg lifted_l(s) \wedge \neg lifted_r(s) \wedge spilling(s), \end{aligned}$$

Along with the requirement:

$$natural(Spills) \wedge natural(Nspills),$$

and the effect axioms:

$$\begin{aligned} & Poss(c, s) \wedge Spills \in c \supset spilling(do(c, s)), \\ & Poss(c, s) \wedge Nspills \in c \supset \neg spilling(do(c, s)). \end{aligned}$$

Here, we have added two different natural actions, one that results in spillage, and the other that stops it. Perhaps a more rigorous treatment of the constraints would have entailed the introduction of four natural actions (one per each constraint that results from writing constraint (14) in clausal form). However, for simplicity we used only two of them.

Our proposal to decompose ramification state constraints in the temporal dimension, by the introduction of new natural actions with their derived triggered effects, is based on the following observations:

[3]Also, this natural action terms should also be considered in the unique names axioms for actions

- In our everyday lives we function under the *common-sense* assumption that actions are deterministic. Non-determinism, in this context, arises because of lack of knowledge of the state of the world, or for lack of control regarding the actions that are performed.
- In physical domains, such as the bowl with soup example, one can accurately predict the direct effects of actions.
- Indirect effects of actions arise because the action creates the conditions for these indirect effects to arise. In general, these effects can be considered to arise subsequent to the performance of the action.

Our view is based on the last observation. Synergy or cancellation arise because of the indirect effects of one or more actions. Thus, this approach is applicable in settings where these observations hold. The applicability can also be judged by comparing it to other approaches to formalize the same phenomena. In the following section, we propose the alternative of treating the set T_{rc} of ramification constraints as *causal rules*, inspired by the work of Lin (1995, 1998). Then, we show how these causal rules can be translated into natural actions and triggered effects in a mechanical manner.

6 Causal Rules

6.1 Propositional Fluents

In this section we discuss some preliminary results regarding the applicability of this approach to handle interactions between effects in theories with concurrent actions. For simplicity, we assume that all fluents are propositional. Thus, fluents are predicates that only take a single argument of sort $\mathcal{S}$.

We start with a theory of action Σ_{rc}, as before. The theory includes the set T_b of axioms (1)-(5), along with a set T_D of domain specific axioms (that only mention terms of type $\mathcal{D}$), a set T_{prec} of action precondition axioms of the form (6), a set T_{eff} of direct effect axioms like (11)-(12), a set T_{rc} of ramification state constraints of the form (40), and a set of axioms T_{S_0} that describe the initial situation (a simple formula applied to S_0).

As mentioned earlier, a set of state constraints are not enough to characterize change in the presence of concurrent actions and interaction between effects. What we need is more refined *causal* information. We assume that each state constraint in the set T_{rc} is written as a *causal* implication or *causal rule*.

Now, we take a theory of action Σ_{rc} and mechanically derive a new theory Σ_{nat}. The steps that are followed to derive this new theory are the following:

1. Each axiom in T_{rc} is written as a *causal rule*, such as:

$$l_1(s) \wedge \ldots \wedge l_n(s) \supset l(s). \tag{23}$$

The implication symbol will be interpreted as a *causality operator*, and not as logical implication. Therefore, the syntactic form of these axioms is very important (the contrapositive is not regarded as equivalent). Perhaps, one should use another operator

for causality, and rewrite the rules accordingly. Instead, we appeal to the reader's understanding.

How this step is achieved depends crucially on the domain of application. All causal rules of the form (23), with the same literal on the right-hand side can be compiled together in a single causal rule of the form:

$$\Phi_l(s) \supset l(s). \tag{24}$$

For notational convenience, we write $T_{rc}(s)$ as the conjunction of all the formulas in T_{rc}, leaving s free. We assume that there are no axioms in T_{eff} in which the literal l is falsified.

Our interpretation of these causal rules corresponds roughly to what Lin has called *dynamic causal rules* (Lin 1998).

These causal rules will not be part of the resulting theory Σ_{nat}, but are used in the steps below.

2. The original language is extended with one new action constant per compiled causal rule of the form (24). The T_{prec} axioms are extended with action precondition axioms of the form (one axiom per causal rule (24)):

$$Poss(\{A_l\}, s) \equiv \Phi_l(s), \tag{25}$$

where A_l is a new action constant. Let T^n_{prec} denote the extended set of effect axioms. Also, T_{una} is extended for these new action constants, leading to T^n_{una}

3. The axioms T_{eff} are extended with a new effect axiom for each new action A_l:

$$Poss(c, s) \wedge A_l \in c \supset l(do(\{A_l\}, s)).$$

Let T^n_{eff} denote the extended set of axioms.

4. Σ_{rc} is extended with a new set of natural axioms. For each new action symbol added to the language in the previous step, we add:

$$natural(A_l).$$

We assume that these are the only natural actions. This assumption can be formalized by a minimization of the extension of this predicate. Given the simple form of these axioms, we can simply write: $natural(a) \equiv a = A_{l1} \vee a = A_{l2} \ldots$. Let T_{nat} be a singleton set with this complete specification for $natural$.

5. The axioms T_{S_0} are extended with a $T_{rc}(S_0)$, that is, for each state constraint $(\forall\, s)\ \Psi(s)$ in T_{rc}, we add $\Psi(S_0)$ as an axiom. Let $T^n_{S_0}$ denote the extended set of axioms about S_0.

6. The theory Σ_{nat} is given by the foundational axioms of the situation calculus, axioms about natural actions (15)-(18), and T_D, T^n_{una}, T^n_{eff}, T^n_{prec}, T_{nat} and $T^n_{S_0}$.

The theory Σ_{nat} is of a form such that we can apply Reiter's approach to derive a monotonic solution to the frame problem. Indeed, except for the axioms about natural actions, the structure of the axiomatization is a standard one. Furthermore, the axioms about natural actions do not interfere with the changes that are direct consequences of actions. Therefore, we can obtain successor state axioms for all fluents of the domain. Let $\mathbf{R}(\Sigma_{nat})$ be the theory with successor state axioms obtained from Σ_{nat} following Reiter's approach.

6.2 Some Properties

Previously, we described how Σ_{rc} can be transformed into a theory $\mathbf{R}(\Sigma_{nat})$. What properties does the resulting theory have with respect to the original one? First, does the resulting theory entail the state constraints for legal situations? The answer is no, thus:

$$\mathbf{R}(\Sigma_{nat}) \not\models (\forall\, s).\ legal(s) \supset T_{rc}(s).$$

$T_{rc}(s)$ is the conjunction of the constraints (26)-(27) applied to s.

Example (cont.): Consider the theory Σ_{rc}:

T_{prec}:
$$Poss(\{A\}, s),$$

T_{eff}:
$$Poss(\{A\}, s) \supset f_1(do(\{A\}, s)),$$

T_{rc}:
$$f_1(s) \supset f_2(s), \tag{26}$$
$$f_2(s) \supset f_3(s), \tag{27}$$

T_{S_0}:
$$\neg f_1(S_0) \wedge \neg f_2(S_0) \wedge \neg f_3(S_0).$$

The theory that results from applying the previous scheme leads to a new theory in which the new action constants A_{f2} and A_{f3} are introduced. After deriving the successor state axioms, we obtain a theory $\mathbf{R}(\Sigma_{nat})$ that includes the following axioms:

$$\begin{aligned}
&natural(a) \equiv a = A_{f2} \vee a = A_{f3},\\
&Poss(\{A\}, s),\\
&Poss(\{A_{f2}\}, s) \equiv f_1(s) \wedge \neg f_2(s),\\
&Poss(\{A_{f3}\}, s) \equiv f_2(s) \wedge \neg f_3(s),\\
&Poss(c, s) \supset f_1(do(c, s)) \equiv A \in c \vee f_1(s),\\
&Poss(c, s) \supset f_2(do(c, s)) \equiv A_{f2} \in c \vee f_2(s),\\
&Poss(c, s) \supset f_3(do(c, s)) \equiv A_{f3} \in c \vee f_3(s),\\
&\neg f_1(S_0) \wedge \neg f_2(S_0) \wedge \neg f_3(S_0).
\end{aligned}$$

Clearly:

$$\mathbf{R}(\Sigma_{nat}) \models f_1(do(\{A\}, S_0)) \wedge \neg f_2(do(\{A\}, S_0)).$$

Thus, violating the original state constraint (26).

However, we do have the following:

$$\mathbf{R}(\Sigma_{nat}) \models (\forall\, s).\ leads(\{A\}, S_0, s) \supset T_{rc}(s).$$

In general:

Proposition 6.1 *Let Σ_{rc} be a theory of action, as in the previous sections, and let $\mathbf{R}(\Sigma_{nat})$ be the theory that results from the application of the steps described earlier. Then:*

$$\mathbf{R}(\Sigma_{nat}) \models (\forall\, s).\ stable(s) \wedge legal(s) \supset T_{rc}(s).$$

The proof of this proposition is fairly direct. If the state constraints were violated in a given situation, then the situation would not be a stable one, prompting for some natural action to occur in it.

If Σ_{rc} is a theory of action without natural actions, and if one wants to know whether some fluent f holds after performing an action A in some situation S, one simply tries to find out whether:

$$\Sigma_{rc} \models S_0 < S \wedge f(do(\{A\}, S)).$$

However, in theories with natural actions, such as $\mathbf{R}(\Sigma_{nat})$, this question becomes:

$$\mathbf{R}(\Sigma_{nat}) \models (\forall\, s).\ leads(\{A\}, S, s) \supset f(do(\{A\}, s)).$$

Thus, we are interested in *stable* changes. Now, we can define abnormality with respect to stable situations as follows:

$$\begin{aligned} &ab^{+}(f, s, c) =_{\text{def}} \\ &\quad stable(s) \wedge leads(c, s, s') \wedge [f(s) \equiv \neg f(s')] \end{aligned}$$

Another property of $\mathbf{R}(\Sigma_{nat})$, in relation to Σ_{rc}, is that there is a natural correspondence between the models of $\mathbf{R}(\Sigma_{nat})$ and the minimal models of Σ_{rc}. If $\mathcal{M}^{+}$ is a model of $\mathbf{R}(\Sigma_{nat})$, then there must be a minimal model $\mathcal{M}$ of Σ_{rc}, such that for each abnormality ab^{+} in $\mathcal{M}^{+}$ there is an abnormality ab in $\mathcal{M}$.

In order to know whether some fluent becomes true after performing some action in a stable situation, we must find a stable situation after the action is performed. If fluents are propositional and finite innumber, and the theory is consistent, it is not hard to show that this stable situation must exist. However, this is not true for theories with non-propositional fluents.

6.3 Non-propositional Fluents

If the language has non-propositional fluents, the existence of a stable situation, after an action is performed, is not guaranteed. Indeed, assume a theory Σ_{rc} in which T_{rc} has a state constraint such as:

$$f(x, s) \supset f(r(x), s),$$

along with an action function a and effect axiom:

$$Poss(c, s) \wedge a(x) \in c \supset f(x, do(c, s)).$$

and an initial situation specification with:

$$(\forall\, x)\ \neg f(x, S_0).$$

It is not difficult to build a theory Σ_{rc} with these axioms such that $\mathbf{R}(\Sigma_{nat})$ will have models in which $leads(a(0), S_0, s)$ is identically false (where 0 is some domain constant).

However, there are many cases in which we can guarantee the existence of stable states after performing any action. For instance, we can take stratified sets of causal rules. The definition of stratification is similar to the one used in logic programming (Apt, Blair, and Walker 1987). Thus, we can define a numbering scheme in which each fluent receives a number. A set T_{rc} is considered stratified if there is a numbering scheme in which, for each causal rule, the numbers assigned to the fluents in the antecedent, are strictly smaller than the numbers assigned to the fluent in the consequent. It is not difficult to show that if the set T_{rc} is stratified, a translation would yield a theory in which stability is guaranteed.

7 Precondition Interaction

In this section we address the problem of deriving preconditions for the executability of concurrent actions. As mentioned before, there are two ways in which interactions between the preconditions of actions may arise:

- One action may not be performed, unless it is accompanied by other, cooperating, action(s). For instance, if we start with two objects C and D in locations X and Y respectively, then moving C from X to Y cannot be performed unless accompanied by another action that moves D from Y to some other clear position Z.

- One action may be performed, unless it is accompanied by another, conflicting, action. For instance, in the situation above, if Z was clear at the start, then moving C from X to Z is possible, unless it is accompanied by moving D from Y to Z at the same time.

The approach to derive preconditions for concurrent actions, is in some ways similar to the approach presented by Lin and Reiter (Lin and Reiter 1994) to derive preconditions for atomic (non-concurrent) actions. We start with a theory Σ_{qc} that contains T_D, T_{una}, T_{S_0} and T_{prec} as in previous sections, along with a set T_{ss} of successor state axioms, and a set T_{qc} of qualification constraints. Thus, we presume that the set T_{rc} is empty.

In the approach discussed in the following subsections new preconditions for the executability of concurrent actions are introduced. First, from direct consistency conditions we incorporate preconditions to state that if two actions, would have direct contradictory effects in a given situation, then the actions cannot be performed together. Later, other precondition axioms are added in order to ensure that, by executing a concurrent action, we do not end up in a situation in which a qualification constraint is violated. This condition, as in (Lin and Reiter 1994), is derived by appealing to regression.

7.1 Consistency Between Direct Effects

The simplest case in which precondition interactions arise in a domain appears in an axiomatization in which actions may have effects that contradict each other. For instance, consider the following axiomatization:

T_{eff} :

$$Poss(c, s) \wedge turnOn \in c \supset on(do(c, s)), \tag{28}$$

$$Poss(c, s) \wedge turnOff \in c \supset \neg on(do(c, s)), \tag{29}$$

T_{S_0} :

$$on(S_0). \tag{30}$$

Here, there are no explicit precondition axioms; therefore, the primitive actions *turnOn* and *turnOff* are always possible. However, by simple inspection of these axioms, the concurrent action $\{turnOn, turnOff\}$ cannot be possible. In fact, from (28) and (29) it follows that:

$$turnOn \in c \wedge turnOff \in c \supset \neg Poss(c, s)$$

More generally, as was stated before, the direct effect axioms can be compiled together in axioms of the form:

$$Poss(c, s) \wedge \gamma_f^+(\underline{\mathbf{x}}, c, s) \supset f(\underline{\mathbf{x}}, do(c, s)), \tag{11}$$

$$Poss(c, s) \wedge \gamma_f^-(\underline{\mathbf{x}}, c, s) \supset \neg f(\underline{\mathbf{x}}, do(c, s)). \tag{12}$$

Therefore, in reminiscence of Reiter's consistency condition (Reiter 1991), we need to add, for each fluent, the following precondition axiom for concurrent actions, which can be derived from (11) and (12):

$$Poss(c, s) \supset \neg(\exists \underline{\mathbf{x}}).(\gamma_f^-(\underline{\mathbf{x}}, c, s) \wedge \gamma_f^+(\underline{\mathbf{x}}, c, s)) \tag{31}$$

Thus, in the example above, we would add:

$$Poss(c, s) \supset \neg(turnOn \in c \wedge turnOff \in c)$$

7.2 Precondition Interaction and State Constraints

How should knowledge about action preconditions be represented? It would certainly be inconvenient to explicitly express the extension of *Poss* for all possible combinations of actions that can be performed concurrently. Thus, we would like to have a general way in which to obtain the extension of *Poss* from the *action precondition axioms* of section 2.2, along with those derived in the previous section. We postulate here, that a convenient manner of expressing further preconditions is through the use of *qualification state constraints*, these constraints may express things like *no two objects may be in the same place at the same time.* Then, if a combination of primitive actions leads to a state that violates this constraint, we simply reject this as a possible combination.

As an example, consider a simple domain in which we have two tokens D_1 and D_2 and four place holders L_1, L_2, L_3 and L_4. In order to simplify things, we subdivide the sort $\mathcal{D}$ into $\mathcal{D}_d$ and $\mathcal{D}_l$, again, the sort for the variables should be clear from the context. Also, we have a fluent function at which takes a token and a place holder as arguments. Thus, $at(D_1, L_1, S_0)$ would mean that token D_1 is in location L_1 in situation S_0. We have a single action function $move$ taking a token and two place holders as arguments. Thus, the action $move(D_1, L_1, L_2)$ denotes the action of moving token D_1 from place holder L_1 to place holder L_2. The following axioms characterize this domain:

T_{dom} :

$$(\forall\, l)\ l = L_1 \vee l = L_2 \vee l = L_3 \vee l = L_4, \tag{32}$$

$$L_1 \neq L_2 \wedge L_1 \neq L_3 \wedge L_1 \neq L_4 \wedge L_2 \neq L_3 \wedge L_2 \neq L_4 \wedge L_3 \neq L_4, \tag{33}$$

$$(\forall\, d)\ d = D_1 \vee d = D_2, \tag{34}$$

$$D_1 \neq D_2, \tag{35}$$

T_{prec} :

$$Poss(\{move(d, l_1, l_2)\}, s) \supset at(d, l_1, s), \tag{36}$$

T_{eff} :

$$Poss(c, s) \wedge move(d, l_1, l_2) \in c \supset at(d, l_2, do(c, s)), \tag{37}$$

T_{qc} :

$$at(d_1, l, s) \wedge at(d_2, l, s) \supset d_1 = d_2, \tag{38}$$

$$at(d, l_1, s) \wedge at(d, l_2, s) \supset l_1 = l_2, \tag{39}$$

T_{rc} :

$$at(d, l_1, s) \wedge at(d, l_2, s) \supset l_1 = l_2, \tag{40}$$

T_{S_0} :

$$at(D_1, L_1, S_0) \wedge at(D_2, L_2, S_0). \tag{41}$$

In order to derive the successor state axioms from these axioms, we first need to get rid of the ramification state constraints in favor of new effect axioms. We obtain the following new effect axiom:

$$Poss(c, s) \wedge move(d, l_2, l_1) \in c \wedge l_1 \neq l_2 \supset \neg at(d, l_2, do(c, s)). \tag{42}$$

Using the resulting set of effect axioms, the following successor state axiom ensues:

$$\begin{aligned} Poss(c, s) \supset at(d, l, do(c, s)) \equiv [&(\exists\, l')\ move(d, l', l) \in c \vee \\ &at(d, l, s) \wedge \neg(\exists\, l')\ (l \neq l' \wedge move(d, l, l') \in c)] \end{aligned} \tag{43}$$

Following Lin and Reiter's work, one can now use the qualification state constraints in order to derive new precondition axioms. This approach makes use of Reiter's regression operator $\mathcal{R}$. For example:

$$\begin{aligned} &\mathcal{R}(at(d, l, do(c, s))) = \\ &\quad [(\exists\, l')\ move(d, l', l) \in c \vee \\ &\quad\ at(d, l, s) \wedge \neg(\exists\, l')\ (l \neq l' \wedge move(d, l, l') \in c)] \end{aligned} \tag{44}$$

A crucial difference between our work and Lin and Reiter's is that we consider actions that may be performed in parallel. For each state constraint we add new precondition axioms. Thus, if $\mathcal{Q}(s)$ denotes an arbitrary qualification state constraint, we add the following:

$$Poss(c, s) \supset \mathcal{R}(\mathcal{Q}(do(c, s))), \tag{45}$$

Intuitively, (45) states that a necessary condition for c to be possible in situation s is that the qualification constraints hold after c is actually executed in s.

In particular for constraint (38) above, we obtain:

$$Poss(c, s) \supset \mathcal{R}(at(d_1, l, do(c, s)) \wedge at(d_2, l, do(c, s)) \supset d_1 = d_2). \tag{46}$$

Now, $\mathcal{R}(at(d, l, do(c, s)))$ is given by (44). Therefore, the resulting precondition can be written as:

$$\begin{aligned} Poss(c, s) \supset \{&[(\exists l')\ move(d_1, l', l) \in c \vee at(d_1, l, s) \wedge \neg(\exists l')\ (l \neq l' \wedge move(d_1, l, l') \in c)] \wedge \\ &[(\exists l')\ move(d_2, l', l) \in c \vee at(d_2, l, s) \wedge \neg(\exists l')\ (l \neq l' \wedge move(d_2, l, l') \in c)] \\ &\supset d_1 = d_2\}. \end{aligned} \tag{47}$$

As a shorthand let us introduce the abbreviation:

$$\begin{aligned} &\mathcal{P}_1(d_1, d_2, l, c, s) = \\ &\quad \{[(\exists l')\ move(d_1, l', l) \in c \vee at(d_1, l, s) \wedge \neg(\exists l')\ (l \neq l' \wedge move(d_1, l, l') \in c)] \wedge \\ &\quad [(\exists l')\ move(d_2, l', l) \in c \vee at(d_2, l, s) \wedge \neg(\exists l')\ (l \neq l' \wedge move(d_2, l, l') \in c)] \\ &\quad \supset d_1 = d_2\}. \end{aligned} \tag{48}$$

Similarly, we can derive an expression for the precondition constraint that can be obtained from (39). The resulting constraint is:

$$Poss(c, s) \supset \mathcal{P}_2(d, l_1, l_2, c, s).$$

where:

$$\begin{aligned} &\mathcal{P}_2(d, l_1, l_2, c, s) = \\ &\quad \{[(\exists l')\ move(d, l', l_1) \in c \vee at(d, l_1, s) \wedge \neg(\exists l')\ (l_1 \neq l' \wedge move(d, l_1, l') \in c)] \wedge \\ &\quad [(\exists l')\ move(d, l', l_2) \in c \vee at(d, l_2, s) \wedge \neg(\exists l')\ (l_2 \neq l' \wedge move(d, l_2, l') \in c)] \\ &\quad \supset l_1 = l_2\}. \end{aligned} \tag{49}$$

Also, in order to derive the action precondition axiom for concurrent actions, we need to incorporate an axiom of the form (31) for the fluent at. From (37) and (42), we derive:

$$Poss(c, s) \supset \neg(move(d, l_1, l_2) \in c \wedge move(d, l_2, l_1) \in c \wedge l_1 \neq l_2). \tag{50}$$

Finally, we rewrite axioms in T_{prec}, thus (36) is replaced with:

$$Poss(c, s) \supset [\{move(d, l_1, l_2)\} \in c \supset at(d, l_1, s)], \tag{51}$$

Now, by appealing to Clark's completion for $Poss$, from (47),(50) and (51):

$$\begin{aligned} Poss(c,s) \equiv & (\forall\, d, d_1, d_2, l, l_1, l_2). \\ & (\forall\, d, l_1, l_2)\ [\neg(move(d, l_1, l_2) \in c \wedge move(d, l_2, l_1) \in c \wedge l_1 \neq l_2)] \wedge \\ & (\forall\, d_1, d_2, l)\ [\mathcal{P}_1(d_1, d_2, l, c, s)] \wedge \\ & (\forall\, d, l_1, l_2)\ [\mathcal{P}_2(d, l_1, l_2, c, s)] \wedge \\ & (\forall\, d, l_1, l_2)\ [\{move(d, l_1, l_2)\} \in c \supset at(d, l_1, s)]. \end{aligned} \tag{52}$$

The axiomatization that embodies the solutions to the problems of interest contains axioms (32)-(35), (41), successor state axiom (43), the general precondition axiom (52), along with unique names for actions, and the state constraints relativized to the initial situation.

With the resulting axiomatization, in particular, appealing to axiom (52), we can answer the following questions:

- Can the concurrent action $\{move(D_1, L_1, L_2)\}$ be performed in S_0? No. This is due to a violation of state constraint (38), which has been compiled into the general precondition axiom.

  In fact, from T_{una}, T_{S_0} it follows that

  $$\mathcal{P}_1(D_1, D_2, L_2, \{move(D_1, L_1, L_2)\}, S_0)$$

  is false

- Can the concurrent action $\{move(D_1, L_1, L_2), move(D_2, L_2, L_1)\}$ be performed in S_0? Yes, because the resulting state would satisfy the qualification constraints, and through regression, these conditions have been compiled into axiom (52).

- Can the concurrent actions $\{move(D_1, L_1, L_3)\}$ and $\{move(D_1, L_1, L_4)\}$ be independently performed in S_0. The answer is yes in both cases. Notice that in both cases, the resulting situations are such that L_1 ceases to be at location L_1. This is a ramification of constraint (40), compiled into successor state axiom (43).

- Can the concurrent action $\{move(D_1, L_1, L_2), move(D_1, L_1, L_3)\}$ be performed in S_0. This ought to be false. Here we notice the double role that the constraint (39), repeated as (40), plays in the derivation of the final theory. Notice that in the previous point both component actions are possible, and the successor state axiom has the built in ramification mentioned above. However, in this case, the same logical axiom, used as a qualification constraint, has led to the general precondition axiom that forbids the performance of this concurrent action.

8 Ramifications versus Qualifications

8.1 Distiguishing Constraints

Lin and Reiter, in (Lin and Reiter 1994), when discussing the limitations of their approach to handling state constraints in the non-concurrent situation calculus, express:

> ... we did not provide a syntactic way of distinguishing ramification from qualification constraints. This is a perplexing and unsatisfying situation made even worse by the fact that conjoining a ramification constraint and a qualification constraint leads to a syntactically acceptable constraint of neither type.

In this article, we have assumed all along that constraints are given in clausal form. Furthermore, we postulate that a clause can be taken as a minimal syntactic form that a constraint can take. Thus, conjoining constraints leads to a non-minimal composite constraint. These non-minimal constraints should not be analyzed in the same way as a single constraint. Therefore, we don't find the fact that a conjunction of a qualification constraint and a ramification constraint does not lead to a constraint of either type to be disturbing. However, it is certainly unsatisfactory not to be able to syntactically characterize ramifications from qualifications.

Unfortunately, not only do we not have a solution to the latter problem, but we have uncovered further complications that arise when considering concurrency. Namely, the fact that constraints can play both roles in the presence of concurrency. Interestingly, in the case of theories without concurrency, this cannot be the case. Indeed, let us assume that in a theory without concurrency we have a constraint that plays both roles: *ramification* and *qualification*. After solving the ramification problem, there are new effects for actions to account for ramifications. These new effects force the successor situation to comply with the constraint. Therefore, the qualification would be redundant. In contrast, in a theory with concurrency, the ramifications of a concurrent set of actions might still invalidate a constraint. In which case, the qualification constraint becomes relevant, making the preconditions for the concurrent action false.

8.2 Integration of State Constraints

When dealing with the ramification problem for theories with concurrency, we assumed that the set of qualification constraints was empty. Also, when dealing with the qualification problem for theories with concurrency, we assumed that the set of ramification constraints was empty. Naturally, we should consider the situation in which we are presented with a theory of action for some domain that includes state constraints for ramifications and for qualifications. The question is whether or not the approaches dealing with effect interactions and precondition interactions are mutually consistent. Unfortunately, it turns out that this need not be the case. Indeed, when introducing natural actions to deal with the effect interaction problem, we specified necessary and sufficient conditions for the executability of the natural actions (see (25)). Furthermore, when dealing with the qualification problem, we added new qualifications for all the actions. These qualifications are expressed with necessary conditions for the executability of these actions, regardless of whether these actions are natural or not.

Thus, if the ramification constraints are embedded in successor state axioms, i.e., theories in which there is no effect interaction, then one can derive new preconditions for action executability. In this case, one can guarantee that any situation reached by performing possible actions from the initial situation will satisfy both the ramification constraints and the qualification constraints.

On the other hand, assume that we start with a theory of action that includes a set of ramification constraints, and a set of qualification constraints. Furthermore, assume that effect interactions arise in this theory, and that we use the approach presented in this paper in which the ramification constraints are interpreted as causal rules. The natural actions utilized in order to model the causal chains have action preconditions of the form (25). One possible way to integrate the approach to handle qualification constraints and causal rules is to relax axioms for natural actions of the form (25). For instance, instead of taking these axioms as providing necessary and sufficient conditions for the executability of the natural actions, one can take these conditions as only necessary. Then, one uses Clark's completion to integrate these preconditions and those derived from the qualification constraint. The problem with this approach is that there is no guarantee that the formalism would entail that in all legal stable situations the constraints are satisfied. When and whether this might prove to be an undesirable feature of our approach is unclear and more research is necessary.

9 Related Work

As discussed earlier, the problem that arises in the context of the interaction among effects of concurrently executed actions is directly related to the ramification problem. In fact, effect interactions can be handled in a more or less straigthforward way except when ramification constraints are present. Therefore, much of the work on the ramification problem could be of interest here. In particular (Thielscher 1997; Pinto 1999; McIlraith 2000; Shanahan 1999; Gustafsson and Doherty 1996). Since the main focus of this article is not the ramification problem, below we only mention the work that postpone a comparison between our approach to dealing with ramifications with the approaches referred to above for a later contribution.

In this article we have not spoken about time. All that can be said is that time progresses as actions are executed. However, the language does not restrict the temporal extension of actions or situations in any manner. Perhaps the only implicit constraint that we have is that all actions performed together must start and end at the same time. Indeed, as has been observed elsewhere, actions with temporal extent or duration can be modeled by using one instantaneous action to model the start of the action with duration, one instantaneous action to model the end of the action with duration, and one fluent to model the ongoing execution of the action. When dealing with this type of actions, one can interpret concurrent execution of actions as an overlap between the execution times of both actions. Thus, the problems that might arise with these actions are of a different nature. This problem has been discussed in (Baier and Pinto 1998).

In (Reiter 1999; Reiter 1996), Reiter presents an approach to model concurrency *via interleaving*. The basic idea is as follows. If a and a' are two actions that are executed concurrently in situation s, then the situations $do(a, s)$ and $do(a', do(a, s))$ can be regarded as having the same starting time. The direct effects of these actions would be the conjoined effects of both primitive actions. For this approach to work, one needs to ensure that all interleavings lead to the same results. One important limitation of this view of concurrency is that the phenomenon of interaction between actions cannot be captured. The example provided by Reiter is a duel scenario in which two duelists shoot at each other in unison. The situation that results from this concurrent shooting is that both duelists are shot. Obviously,

the result of the interleaved shooting is that one duelist gets shot first.

Clearly, the view of concurrency via interleaving has the advantage of simplicity. In particular, the issue of precondition interaction does not arise. Furthermore, the ontology is simpler, since there is no need for a separate sort for concurrent actions. Thus, the main drawback of interleaved concurrency is its representational limitations (illustrated with the duel example). Furthermore, in order to determine whether interleaving can be used, one needs to prove that all possible orderings of the concurrently executed actions lead to the same results.

In (Karlsson and Gustafsson 1999), Karlsson and Gustafsson introduce the logic of action TAL-C, which is based on the temporal action logic TAL (Doherty 1994) and on Features and Fluents (Sandewall 1995). This logic was explicitly designed in order to address the problem of modeling interactions between actions that are concurrently executed. However, most of the results presented in this work pertain to the execution of concurrent actions when actions have durations. The actions that are executed can overlap in time, and interactions might arise as a result of the different effects that the actions have. On the other hand, our work is centered upon the modeling of interaction between instantaneous actions executed concurrently. As mentioned before, in (Baier and Pinto 1998), we address the modeling of this type of action. In their paper, Karlsson and Gustafsson, argue for the introduction of the concept of *influence*. This concept, is used in order to state that fluents are influenced to take certain truth values. Combinations of influences are then used to explain when fluents actually change. The most interesting observation that Karlsson and Gustafsson make is that interaction arises at the fluent level and not at the action level. This we understand as meaning that, in our formalism, interaction has to be modeled through *state constraints*, as indeed we do. However, one feature that TAL-C has, and that our approch lacks, is the ability to deal with non-determinism. However, the issue of non-determinism seems to us to be independent from concurrency, and is treated elsewhere (Pinto, Sernadas, Sernadas, and Mateus 1998).

In (Baral and Gelfond 1997), Baral and Gelfond propose the language $\mathcal{A}_C$, which extends the syntax and semantics of the language $\mathcal{A}$ (see (Gelfond and Lifschitz 1992)) in order to represent concurrent actions. Their approach follows an earlier approach advanced in (Gelfond, Lifschitz, and Rabinov 1991), which was formulated in a situation calculus language. Baral and Gelfond do not address the qualification problem for concurrent actions (or precondition interaction problem). With regards to the effect interaction problem, they propose an approach in which the interaction is modeled explicitly. For instance, if there are two actions a_1 and a_2 and there is effect cancellation, then there would be effect axioms for a_1, effect axioms for a_2, and effect axioms for a_1 and a_2 together. If the effects do not interact, then the effects of a_1 and the effects of a_2 are *inherited* by the concurrent action $\{a_1, a_2\}$. The main drawback of this approach is the lack of *modularity*, if one were to introduce a new action in the domain, one would have to study the possible interaction of effects with all other actions in the domain. Our initial goal was to avoid exactly this problem.

In (Giordano, Martelli, and Schwind 1998), a modal action logic in which actions are represented as modalities, *à la* dynamic logic (Harel 1979), is used to model concurrent actions. This modal action logic introduces two types of modalities to represent the effects of actions on the state of the world. One type of modality is used to describe the effects of

actions when executed in isolation, while the other is used to describe the effects of actions regardless of whether or not other actions are performed concurrently. Furthermore, causal rules are introduced in order to address the ramification. Although this approach differs from ours in many ways, there is an important common characteristic, which is the use of *causal rules* in order to deal with the interaction between concurrently executed actions. Giordano *et al.* do not address the *precondition interaction* problem.

In (Thielscher 2000b), Thielscher presents a thorough approach to modeling action and change based on the *Fluent Calculus*, "a specification language for robots with sensors in nondeterministic, concurrent and ramifying environments". The fluent calculus presented there is the result of further research based on previous work by Thielscher on previous versions of the fluent calculus (e.g., (Thielscher 1997; Thielscher 2000a)). In the report mentioned, Thiescher puts together a coherent framework for the specification of agents in dynamics worlds. Although there are many differences in the logical language and in the general approach taken to modeling dynamic worlds, there are interesting similarities between Thielscher's work and ours. First, our treatment of ramifications in the context of concurrent actions relies heavily on structuring ramification constraints as *causal rules*. When ramification constraints are violated, new actions are invoked that repair the state of the world as dictated by the causal rules. In our approach the scheme results in a *natural process*, which ends in a situation that is consistent with the state constraints. Thielscher's approach is conceptually similar, in the sense that ramification are modeled as causal chains. Unfortunately, a technical comparison between the two approaches would take us to far afield. Suffice it to say here that we treat *causality* as a meta level notion which is translated into effect axioms, whereas Thielscher introduces causality as a predicate which is then used to derive ramifications (causal chains). With regards to the treatment of preconditions for concurrent actions and the resulting qualificatio problem, Thielscher makes use of explicit *concurrent action precondition* axioms. This is an important departure from our approach, in which we explicitly wanted to avoid having to specify a possibly combinatorial number of such axioms.

10 Summary and Discussion

This article deals with two separate but related problems that arise in theories of action with *true concurrency*; namely, the problems of describing effect interactions and precondition interactions for concurrently executed instantaneous actions. The starting premise of this research is that most interactions arise because of the existence of domain constraints. For example, the constraint that an object cannot be in two places at the same time, makes the actions of moving the object to two different locations at the same time impossible.

To study the interaction between actions when executed concurrently, we first dealt with the problem of writing theories of action in which the effects of actions interact. The main result drawn from this work is that to model domains in which effect interactions arise, effect axioms and state constraints are not enough to provide all the necessary information to perform the right inferences. Our position regarding this issue, is that this should be taken as a modeling problem. We postulate that effect interactions arise due to the existence of intermediary natural actions, which are triggered by the direct effects of the concurrently

executed actions. An alternative approach was to address this same problem by integrating *causal* information. Furthermore, we showed how to take this causal information and translate it into the approach based on intermediary natural actions.

The second problem addressed in this article, namely the *precondition interaction* problem, turned out to be easier to solve. The starting point is a theory in which successor state axioms are already available for all fluents in the language. Then, the preconditions for the execution of concurrent actions (c) are obtained by regressing the qualification constraints from the situation that results from the execution of the concurrent action ($do(c, s)$) to the starting situation (s). This approach is similar to the approach used by Lin and Reiter to derive preconditions for non-concurrent actions in theories without concurrency.

Perhaps the most disturbing shortcoming of the work we present in this article is the fact that we might end up with a theory in which the qualification constraints are satisfied but in which the ramification state constraints are not. This might happen if one uses triggers to introduce actions that act as intermediary causes. The derivation of further preconditions for all actions from the qualification constraints might make these actions not possible in situations where the original ramification constraints are not satisfied. More research is required in order to determine how this problem might arise in practice.

Acknowledgements

References

Allen, J. and G. Ferguson (1994). Actions and Events in Interval Temporal Logic. *Journal of Logic and Computation 4*(5). Special Issue on Actions and Processes.

Apt, K., H. Blair, and A. Walker (1987). Towards a Theory of Declarative Knowledge. In J. Minker (Ed.), *Foundations of Deductive Databases and Logic Programming*, pp. 89–148. Morgan Kaufmann.

Baier, J. and J. Pinto (1998). Non-instantaneous Actions and Concurrency in the Situation Calculus (Extended Abstract). In G. de Giacomo and D. Nardi (Eds.), *10th European Summer School in Logic, Language and Information.*

Baker, A. (1989, May). A Simple Solution to the Yale Shooting Problem. In H. L. R. Brachman and R. Reiter (Eds.), *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, pp. 11–20. Morgan Kaufmann Publishers, Inc.

Baral, C. and M. Gelfond (1997). Reasoning About Effects of Concurrent Actions. *Journal of Logic Programming 31*, 85–118.

Bedrax-Weiss, T. and L. Bertossi (1995). Underlying Semantics for the Assessment of Reiter's Solution to the Frame Problem. In *Advances in Artificial Intelligence, Proc. 1st International Conference of the Brazilian AI Society (SBIA 95)*, Number 991 in LNAI, pp. 59–68. Springer.

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.

Brewka, G. and J. Hertzberg (1993). How to do things with worlds: on formalizing actions and plans. *Journal of Logic and Computation 3*, 517–532.

del Val, A. and Y. Shoham (1993). Deriving Properties of Belief Update from Theories of Action (II). In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-93)*, Chambery, pp. 732–737.

Doherty, P. (1994). Reasoning about action and change using occlusion. In A. Cohn (Ed.), *ECAI 94. 11th European Conference on Artificial Intelligence*.

Gelfond, M. and V. Lifschitz (1992). Representing Actions in Extended Logic Programs. In *Joint International Conference and Symposium on Logic Programming*, pp. 559–573.

Gelfond, M., V. Lifschitz, and A. Rabinov (1991). What are the Limitations of the Situation Calculus? In S. Boyer (Ed.), *Automated Reasoning, Essays in Honor of Woody Bledsoe*, pp. 167–181. Kluwer Academic Publishers.

Giordano, L., A. Martelli, and C. Schwind (1998). Dealing with Concurrent Actions in Modal Action Logics. In H. Prade (Ed.), *Proc. of the 13th European Conference on Artificial Intelligence, ECAI'98*, pp. 537–541. John Wiley & Sons.

Gustafsson, J. and P. Doherty (1996). Embracing Occlusion in Specifying the Indirect Effects of Actions. In *Proceedings of the 5th Int'l Conference on the Principles of Knowledge Representation in Reasoning (KR'96)*.

Harel, D. (1979). *First-Order Dynamic Logic*, Volume 68 of *Lecture Notes in Computer Science*. New York: Springer-Verlag.

Karlsson, L. and J. Gustafsson (1999, Oct). Reasoning about concurrent interaction. *Journal of Logic and Computation 9*(5).

Kelley, T. G. (1996). Modeling Complex Systems in the Situation Calculus: A Case Study Using the Dagstuhl Steam Boiler Problem. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifth International Conference(KR'96)*, Cambridge, Massachussetts, U.S.A. Morgan Kaufmann.

Lifschitz, V. (1990). Frames in the space of situations. *Artificial Intelligence 46*, 365–375.

Lifschitz, V. (1994). Circumscription. In *Handbook of Logic in AI and Logic Programming*, Volume 3, pp. 298–352. Oxford University Press.

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. (1998). On the Relationships between Static and Dynamic Causal Rules in the Situation Calculus. In *Proceedings of the AAAI Workshop on Causality*. AAAI. To appear.

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

McCarthy, J. (1963). Situation, actions, and causal laws. Technical Report Memo 2, Stanford Artificial Intelligence Project.

McCarthy, J. (1986). Applications of Circumscription to Formalizing Commonsense Knowledge. *Artificial Intelligence 28*, 89–116.

McIlraith, S. (2000, January). An Axiomatic Solution to the Ramification Problem (Sometimes). *Artificial Intelligence 116*(1-2), 87–121.

Miller, R. and M. Shanahan (1994). Narratives in the Situation Calculus. *The Journal of Logic and Computation 4*(5), 513–530.

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). Concurrent actions and interacting effects. In A. Cohn, L. Schubert, and S. Shapiro (Eds.), *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixth International Conference (KR'98)*, San Francisco, CA, pp. 292–303. Morgan Kaufmann Publishers.

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

Pinto, J. and R. Reiter (1993, June). Temporal Reasoning in Logic Programming: A Case for the Situation Calculus. In D. S. Warren (Ed.), *Proceedings of the Tenth International Conference on Logic Programming*, Budapest, pp. 203–221. The MIT Press.

Pinto, J., A. Sernadas, C. Sernadas, and P. Mateus (1998). Non-determinism and uncertainty in the situation calculus. Submitted for publication.

Pinto, J. A. (1999). Compiling Ramification Constraints into Effect Axioms. *Computational Intelligence 15*(3), 280–307.

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. (1996, November). Natural Actions, Concurrency and Continuous Time in the Situation Calculus. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifth International Conference(KR'96)*, Cambridge, Massachussetts, U.S.A. Morgan Kaufmann.

Reiter, R. (1999). KNOWLEDGE IN ACTION: Logical Foundations for Describing and Implementing Dynamical Systems. Book Draft, available from `http://www.cs.utoronto.ca/~cogrobo`.

Sandewall, E. (1995). *Features and Fluents, A Systematic Approach to the Representation of Knowledge about Dynamical Systems.* Oxford University Press.

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.

Shanahan, M. (1999). The Ramification Problem in the Event Calculus. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-99)*.

Shanahan, M. P. (1997). *Solving the Frame Problem: A Mathematical Investigation of the Common Sense Law of Inertia*. MIT Press.

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

Thielscher, M. (2000a). The concurrent, continuous fluent calculus. *Studia Logica*.

Thielscher, M. (2000b). The fluent calculus. Technical Report CL-2000-01, Computational Logic Group, Artificial Intelligence Institute, Department of Computer Science, Dresden University of Technology.