Integrating Discrete and Continuous Change in a Logical Framework

Javier Pinto
Depto. de Ciencia de la Computación
Pontificia Universidad Católica de Chile
Casilla 306, Santiago 22, Chile
phone: +(56-2) 686-5868
fax: +(56-2) 686-4444
URL: http://lyrcc.ing.puc.cl/jpinto/
email: jpinto@ing.puc.cl

Integrating Discrete and Continuous Change in a Logical Framework

Javier Pinto
Depto. de Ciencia de la Computación
Pontificia Universidad Católica de Chile

Abstract

The goal of our work is to develop theoretical foundations for the representation of knowledge in domains in which properties may vary continuously. One achievement of our research is that it extends the applicability of current research on theories of action. Furthermore, we are able to apply known approaches to the frame and ramification problems, developed for discretely changing worlds, to domains in which the world changes continuously.

Our approach is based on the *discrete situation calculus* and on a monotonic solution to the frame problem. In order to address the combined frame and ramification problems, we extend Lin and Reiter's work. We use Pinto and Reiter's extension to the situation calculus to represent *occurrences*. We extend this work further to allow for reasoning by default. For example, if we know that a ball is falling and we do not have any reason to believe that an action would interfere with the ball's motion, then we assume that the ball will hit the ground. Finally, we extend the language of the situation calculus to allow for properties that change within situations. We also show that our proposed situation calculus inherits the solutions to the frame and ramification problems.

Keywords: Temporal Reasoning, Natural Actions, Continuous Change, Frame and Ramification Problems.

1 Introduction and Motivation

Research on logical approaches to the representation of knowledge about actions and change has been concerned with the development of solutions to the *frame*, *ramification*, and other related problems. Most of the research on these problems has dealt with domains in which change is discrete.

In order to build intelligent systems capable of inhabiting our world, we require theories that integrate discrete and continuous change. The aim of this paper is to provide a theoretical framework that achieves this integration. An important goal is to produce a framework in which solutions to the *frame* and *ramification* problems in discrete worlds can be naturally extended to domains in which some properties vary continuously with time.

Our work is based on the situation calculus, a state based framework in which the world is assumed to change from one state into another as a result of events happening or actions being performed. In a state based representation, the world's properties do not vary within a state (i.e., the world is *static*). Furthermore, the set of states the world goes through is considered to be *discrete*. Thus, state-based representations seem unsuitable for the representation of worlds in which properties change continuously.

To address this limitation we incorporate *laws of motion* explicitly into the logical language. Thus, we start from the assumption that any continuously varying property follows some *law* that establishes the manner in which the property varies within a state.

Our proposal is based on an important assumption, namely, the *discrete-events assumption*: We assume that during any given finite interval of time only a finite number of actions may occur. This assumption means that for any finite period of time, the world can go through a finite set of *situations*. Within a situation, the world may be changing, but this change is described by some *law of motion*. Thus, a *continuous event*, like *pushing the accelerator* or *skiing down a slope*, can be modeled as a property that varies continuously, and whose variation is started and ended by events.

An essential characteristic of our formalization is that we consider events to be instantaneous. This is not an *assumption* in the sense that we do not restrict the applicability of our language to worlds in which events are instantaneous. Rather, it is an approach to modeling world-events using a logical language. In reality, events may take time; i.e., they may have *durations*. Events with durations are easily modeled in our language. Each event with a duration is modeled as a pair of instantaneous events (its *starting* and *ending* events) along with a property that is true if and only if the event is in progress. From a formal point of view, it can be shown that this way of modeling events with durations is equivalent to one in which events are considered, at the linguistic level, to have a duration. For instance, the event of sipping coffee can be modeled with the instantaneous events *start sipping coffee* and *end sipping coffee*; in addition to a fluent *sipping* that would be true after the *start* event and before the *end* event.

The paper is organized as follows: In section 2, we present Reiter's *discrete situation calculus* (Reiter 1993), which is characterized by the use of an induction axiom for situations. Induction is essential for proving properties of states in the situation calculus (Reiter 1993). Furthermore, we present Lin and Reiter's work on the ramification problem (Lin and Reiter 1994). Later on, we present a solution to the frame and ramification problem for theories with binary state constraints. Our solution is shown to be correct according to Lin and Reiter's correctness criterion.

In section 3 we present Pinto and Reiter's work on an actual line of situations (Pinto and Reiter 1995). Their basic idea is to increase the expressive power of the situation calculus language with a time line. This time line allows one to talk about event occurrences, as it is possible in linear temporal logics (e.g., Allen's Interval Temporal Logic (Allen 1984; Allen and Ferguson 1993) and Kowalski and Sergot's Calculus of Events (Kowalski and Sergot 1986)). We extend Pinto and Reiter's work by showing that it is necessary to introduce a special non-monotonic semantics to characterize what events can be expected to occur given a first order description of a domain. We use circumscription (McCarthy 1986; Lifschitz 1985) to specify the preferential criterion. Finally, we use some examples to illustrate how the extended language inherits the solutions to the frame and ramification problems described in section 2.

In section 4 we present our proposal to represent continuous properties and continuous change. Also, we introduce the notion of *natural event* and present some examples. Furthermore, we show how our approach inherits the solutions to the frame and ramification problems. Finally, in section 5 we conclude and discuss our ongoing research.

2 Formal Framework

In this section we discuss the formal framework in which we formalize our proposal to model continuous change. First, we present a *discrete situation calculus*, which has been extended to accommodate *occurrences* and notions of *linear time*. Later, we discuss how the frame and ramification problems are addressed within this framework. Finally, we present examples to illustrate the approach advocated in the absence of continuous change.

2.1 The Discrete Situation Calculus

The situation calculus is a sorted second order language with equality. We use the standard operators $\wedge$, $\vee$, $\supset$ and $\equiv$, and standard quantifiers $\forall$ and $\exists$. We assume that the conjunction operator $\wedge$ takes precedence over the disjunction operator $\vee$. There are several domain sorts $\mathcal{A}, \mathcal{S}, \mathcal{F}, \mathcal{T}, \mathcal{R}, \mathcal{D}$ for action types, situations, fluents, time, the real numbers, and arbitrary domain objects. The sorts $\mathcal{A}, \mathcal{S}, \mathcal{F}, \mathcal{R}$ and $\mathcal{D}$ are independent, while $\mathcal{T}$ is a sub-sort of $\mathcal{R}$. We consider the interpretation of $\mathcal{R}$ fixed as the set of real numbers and $\mathcal{T}$, also fixed, as the non-negative real numbers. We also assume standard interpretations for real operators ($+$, $-$, $\times$ and $/$). Letters a, s, f, d, t and r are used for variables (and their upper case counterparts for constants) of sorts $\mathcal{A}, \mathcal{S}, \mathcal{F}, \mathcal{D}, \mathcal{T}, \mathcal{R}$ respectively. We have a single 1-place predicate variable φ over situations. Other variables and constants will be assumed to be of sort $\mathcal{D}$. Also, free variables in formulas are assumed to be universally quantified with maximal scope.

We include the special constant S_0 of sort $\mathcal{S}$. The function $do : \mathcal{A} \times \mathcal{S} \rightarrow \mathcal{S}$ is used to denote the situation that results from performing one action on another situation[1]. The relation $< \subseteq \mathcal{S} \times \mathcal{S}$ is used to denote precedence in the tree structure of situations. The relation $Poss \subseteq \mathcal{A} \times \mathcal{S}$ should hold whenever the preconditions for an action hold in a situation. Also, the predicate $holds \subseteq \mathcal{F} \times \mathcal{S}$, is true of a fluent in a situation, if that fluent represents a property that is true in that situation. Other functions and relations will be introduced later. We use the abbreviation $s_1 \leq s_2$ to

[1] This function is called *result* in other formalizations of the situation calculus.

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

The basic structural axioms for situations are:

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

$$do(a_1, s_1) = do(a_2, s_2) \supset a_1 = a_2, \tag{2}$$

$$s_1 < do(a, s_2) \equiv s_1 \leq s_2, \tag{3}$$

$$s_1 < s_2 \supset \neg s_2 < s_1. \tag{4}$$

Axiom (1) is an induction axiom, necessary to prove properties true in all situations (see appendix A for examples of such proofs). It defines the universe of situations as the smallest set that includes S_0 and such that if s is in the set of situations and a is an element of $\mathcal{A}$, then $do(a,s)$ is also an element of $\mathcal{S}$.

Axioms (3)-(4) establish the properties of $<$. Axiom (2) along with the axioms for $<$ ensure that the structure of situations form a tree. A careful reader might wonder why the right side of (2) does not include a conjunct $s_1 = s_2$; it turns out to be unnecessary, since this equality can be inferred from the basic axioms (see (121) in proposition A.1 below).

Our notion of $<$ differs from the one used by Reiter (1991) in an important way. For us, $s < s'$ is true if and only if there exists a sequence of actions $a_1, a_2, \ldots, a_k$ such that $s' = do([a_1, a_2, \ldots, a_k], s)$. In contrast, Reiter's notion is stronger, in the sense that not only must a sequence of actions exist that satisfies the previous equality, but also every action along the sequence must be possible in the situations in which they are performed. In a later section, we use this alternative notion. Therefore, we introduce this second notion as a new relation $\prec \subseteq \mathcal{S} \times \mathcal{S}$, and with the abbreviation $\preceq$, such that $s \preceq s' \equiv s \prec s' \vee s = s'$.

Based on Lin and Reiter (1993) $\prec$ is axiomatized as:

$$\neg s \prec S_0, \tag{5}$$

$$s_1 \prec do(a, s_2) \equiv Poss(a, s_2) \wedge s_1 \preceq s_2. \tag{6}$$

In what follows Σ_b will denote the axioms (1)-(6).

In the appendix A, we prove some useful properties of the structure of situations, which result from axioms Σ_b.

Finally, let us introduce McCarthy's *abnormality* predicate *ab* as:

$$ab(f, a, s) \equiv \neg[holds(f, do(a, s)) \equiv holds(f, s)]. \tag{7}$$

Thus, a fluent f is abnormal with respect to an action a in a situation s if the action makes f change from s to $do(a,s)$. Many non-monotonic approaches to the frame problem are based on the minimization of this predicate (e.g., (Baker 1989; Lifschitz and Rabinov 1989)); however, we introduce it only as a shorthand. In particular, we use it to simplify the presentation of the following proposition:

Proposition 2.1 *From axioms (1)-(4) and (7), it follows that*[2]*:*

$$s_1 < s_2 \wedge holds(f, s_1) \wedge \neg holds(f, s_2) \supset$$
$$(\exists s, a)\,(s_1 < do(a, s) \leq s_2) \wedge ab(f, a, s).$$

That is, if a fluent changes its truth value between two situations s_1 and s_2, it must be the case that some action a was responsible for this change. Furthermore, a situation s must exist such that both, s and $do(a, s)$ lie on the path between s_1 and s_2.

2.2 The Frame Problem

One of the hardest problems encountered in reasoning about actions has been the so called *frame problem*. This problem has attracted much interest in the AI community and many solutions for it have been proposed (e.g., (Georgeff 1987; Haas 1987; Lifschitz 1987; Shoham 1988; Baker 1991; Reiter 1991; Schubert 1990; Lin and Shoham 1992)). Basically, the problem is understood as one of finding a way to *succinctly* specify the effects of actions, given the situations in which they are performed. We appeal to the monotonic solution proposed by Reiter (1991), which we describe below.

A theory of action Σ consists of the following sets of axioms:

- A set Σ_{bd}, which includes the set Σ_b of basic situation calculus axioms (refer to section 2.1), as well as a set Σ_d of situation independent axioms formalizing the domain of interest.

- A set T_{pos} of action precondition axioms: a set of equivalences that establish necessary and sufficient conditions for actions to be executable. We assume that all the conditions for the execution of an action are known. Thus, for each action A, we can write an axiom of the form:

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

  The formula $\Pi_A(s)$ is a *simple* state formula with a unique free situation variable s. A simple formula is a first order formula which does not mention the predicate symbols *Poss*, $\prec$ or $\preceq$ (nor $<$ or $\leq$), that mentions a unique situation variable (e.g. s) and no other situation terms, and which does not quantify over variables of sort $\mathcal{S}$ (Reiter 1992b). Furthermore, s is not allowed to appear outside the scope of a *holds* predicate (as its second argument).

- A set T_{S_0} of sentences about S_0. That is, a description of the initial situation.

- A set $T_{ef} = T_{ef}^{+} \cup T_{ef}^{-}$, where T_{ef}^{+} is a set of general positive effect axioms; for fluent function R and action a, axioms of the form:

  $$Poss(a, s) \wedge \gamma_R^{+}(a, s) \supset holds(R, do(a, s)). \tag{8}$$

[2] All proofs are in the appendix.

T_{ef}^{-} is a set of general negative effect axioms; for fluent R and action a, axioms of the form:

$$Poss(a,s) \wedge \gamma_R^-(a,s) \supset \neg holds(R, do(a,s)). \tag{9}$$

Here, $\gamma_R^+(a,s)$ and $\gamma_R^-(a,s)$ are simple formulas which are used to provide conditions under which an action a produces an effect on a fluent R. In (8) and (9) we omit all but the action and state arguments.

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

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

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

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

- A set T_{unf} of unique names axioms for fluents. For distinct fluent names f and f':

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

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

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

Notice that this assumption is necessary only because we have chosen to reify fluents. In the future we will consider these unique names axioms to be part of the set Σ_d of domain specific axioms.

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

$$Poss(a,s) \wedge \neg holds(R,s) \wedge holds(R, do(a,s)) \supset \gamma_R^+(a,s).$$

Where $\gamma_R^+(a,s)$ can be understood as an explanation why R changed from not holding in s to holding in $do(a,s)$, as result of performing action a. Similarly, the assumption that the general effect axioms characterize all the conditions for a state s under which action a makes $\neg holds(R, do(a,s))$ true is formalized with:

$$Poss(a,s) \wedge holds(R,s) \wedge \neg holds(R, do(a,s)) \supset \gamma_R^-(a,s).$$

Reiter provides an approach to mechanically integrate Schubert's explanation closure axioms with the general effect axioms to derive a set of *successor state axioms*. A successor state axiom for fluent R is of the form:

$$Poss(a,s) \supset [holds(R, do(a,s)) \equiv \gamma_R^+(a,s) \vee holds(R,s) \wedge \neg\gamma_R^-(a,s)].$$

That is, given a situation, if an action is possible we can determine whether the fluent R will hold in the subsequent situation. For this approach to work, it is necessary to have a finite set of effect axioms. Reiter's proposal is to replace

the effect axioms with the successor state axioms, yielding a new theory $\Sigma_{bd} \cup T_{ss}(T_{ef}) \cup T_{pos} \cup T_{S_0} \cup T_{una}$. Where we write $T_{ss}(T_{ef})$ to denote the successor state axioms that result from utilizing explanation closure axioms and the effect axioms T_{ef}. So, for each fluent function in the language, we have a successor state axiom. Notice that if for some fluent F there are no effect axioms, the resulting successor state axioms would state that:

$$Poss(a,s) \supset [holds(F, do(a,s)) \equiv holds(F,s)].$$

Thus, F does not change due to the performance of any action.

Reiter's proposal solves the frame problem by providing an axiomatization that requires a number of axioms proportional to the sum of fluent and action function symbols plus the number of actions. As already mentioned, a limitation of this approach is that it does not consider theories that include axioms that constrain the *legal* ways in which fluents can change. For example, consider an axiomatization in which a one handed robot can perform the action *Drop*, which has the effect of leaving its hand empty. If we had a constraint like: *an object is either in the hand or on the floor*, it should be possible to infer that a ramification of a *Drop* action is that whatever is dropped will be on the floor. Such constraints are known as *state constraints*. The ramification problem is addressed next.

2.3 The Ramification Problem

In (Lin and Reiter 1994), Lin and Reiter studied the problem of extending Reiter's approach to the frame problem to theories that include a set of state constraints. In particular, they came up with a set of sufficient conditions under which a set of successor state axioms can be said to correctly solve the frame problem. The notion of *correctness* of the solution is understood in terms of its equivalence to a specific non-monotonic solution to the frame problem. In this section, we summarize the definitions and results of (Lin and Reiter 1994) that we will use later.

Formally, we extend theories of action with:

- A finite set $T_{sc}(s)$ of domain dependent state constraints[3], such that each constraint is a simple formula with a unique free variable s. In the future we will write T_{sc} to refer to the universal closure of $T_{sc}(s)$. Also, we use $T_{sc}(s)$ to refer to the conjunction of all the state constraints applied to the situation term s.

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

$$\Sigma_{sc} = \Sigma_{bd} \cup T_{ef} \cup T_{pos} \cup T_{S_0} \cup T_{una} \cup T_{sc}. \tag{10}$$

In Lin and Reiter's proposal, a solution to the frame and ramification problem for a theory Σ_{sc} is a theory $\Sigma_{sc'}$ such that:

$$\Sigma_{sc'} = \Sigma_{bd} \cup T_{ss}(T_{ef}, T_{sc}) \cup T_{pos} \cup T_{S_0} \cup T_{una},$$

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

where $T_{ss}(T_{ef}, T_{sc})$ represents a set of successor state axioms derived from the effect axioms and the state constraints. Naturally, not every set $T_{ss}(T_{ef}, T_{sc})$ will correspond to a correct solution. In order to assess the correctness of a proposed solution, Lin and Reiter take an alternative non-monotonic solution to the frame problem as a benchmark. Thus, a set of successor state axioms is regarded as a solution if and only if the models of the theory $\Sigma_{sc'}$ (with a simple extension) are the same as the non-monotonic models of Σ_{sc}. These non-monotonic models are defined using a minimal model semantics.

Lin and Reiter's key result states that the minimal models of Σ_{sc} are exactly the standard models of $\Sigma_{sc'}$ conjoined with the state constraints applied to the initial situation and the impossible situations[4]. The notion of minimality is based on Lin and Shoham's (1991). In order to define the minimality criterion more formally, let s, a, a', f and f' denote variables of sorts $\mathcal{S}$, $\mathcal{A}$, $\mathcal{F}$ (according to our convention). Also, let σ_s, σ_a and σ_f denote assignment functions from free situation, action and fluent variables to domain objects of the respective sorts. A model $\mathcal{M}'$ is preferred to a model $\mathcal{M}$ (i.e., $\mathcal{M}' < \mathcal{M}$) iff the following conditions hold:

- $\mathcal{M}'$ and $\mathcal{M}$ share the same domain and they interpret everything the same with the possible exception of the predicate *holds*.
- For some variable assignment to situations σ_s, $\mathcal{M}'$ and $\mathcal{M}$ are such that:
  - For any assignment σ_f, $\mathcal{M}'$ and $\mathcal{M}$ agree on $holds(f,s)$, that is:

    $$\mathcal{M}', \sigma_s, \sigma_f \models holds(f,s)$$

    if and only if:

    $$\mathcal{M}, \sigma_s, \sigma_f \models holds(f,s).$$

  - For any assignments σ_a, σ_f, if

    $$\mathcal{M}, \sigma_s, \sigma_a, \sigma_f \models s \prec do(a,s) \wedge \neg ab(f,a,s),$$

    then

    $$\mathcal{M}', \sigma_s, \sigma_a, \sigma_f \models \neg ab(f,a,s).$$

  - There are two assignments, σ_a, σ_f, such that:

    $$\mathcal{M}, \sigma_s, \sigma_a, \sigma_f \models s \prec do(a,s) \wedge ab(f,a,s),$$

    but

    $$\mathcal{M}', \sigma_s, \sigma_a, \sigma_f \models \neg ab(f,a,s).$$

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

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

[4] A situation s is impossible iff $\neg S_0 \preceq s$.

Theorem 2.1 *Given a theory Σ_{sc} as in (10), let $T_{ss}(T_{ef}, T_{sc})$ be a set of successor state axioms (one for each fluent function symbol) of the form:*

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

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

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

and

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

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

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

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

$$\Sigma_{sc'} \cup T_{sc}(S_0) \cup T_{sc}^{-}$$

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

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

The first condition in the theorem, $T_{una} \models (\forall a, s, x_1, \ldots, x_n).\neg(\Psi_f \wedge \Psi_{\neg f})$, should be understood as a consistency condition. In fact, it is a requirement that no single action have the effect of making a fluent F true in some situation and the effect of making the fluent false in the same situation.

The second condition imposes the requirement that the new theory with a set of successor state axioms entail the state constraints whenever these are satisfied by the initial situation. Thus, we should guarantee that the updates performed using the successor state axioms preserve the consistency of the successor state with respect to the state constraints. Therefore, the state constraints are now implicit in the successor state axioms.

The relevance of Lin and Reiter's result is that it provides a criterion to judge whether or not a set T_{ss} of successor state axioms constitutes a correct solution to the frame problem. Unfortunately, this correctness criterion is not absolute. In fact, it relies completely upon a very specific solution to the frame problem based on a non-monotonic logic. Lately, Bedrax-Weiss and Bertossi (1995) have assessed the applicability of Reiter's solution to the frame problem using Sandewall's (1995) methodology. In their assessment, Bedrax-Weiss and Bertossi showed that, under certain conditions, Reiter's solution to the frame problem was correct and complete.

In the next sub-section, we propose an approach to generate successor state axioms obtained from a set of effect axioms and a set of state constraints. We make use of theorem 2.1 to justify the correctness of the approach,

2.4 Binary State Constraints

In the previous section, we noted that Reiter's original proposal to deal with the frame problem (Reiter 1991) was not general enough to deal with theories that include a set of state constraints. Then, we briefly outlined Lin and Reiter's results to deal with state constraints and solve the so called ramification problem. In summary, they provide sufficient conditions that a set of successor state axioms must satisfy in order for it to be considered a solution to the combined frame and ramification problems.

We have extended this line of work by studying theories with specific classes of state constraints. For these theories, we have a specific approach to replace state constraints with successor state axioms (Pinto 1997). Here, we only discuss how to address binary state constraints.

Consider a theory Σ_{sc} in which the set T_{sc} of state constraints is written in semi-clausal form. That is, each constraint is a sentence of the form:

$$\rho(\mathbf{x}_r) \vee holds([\neg]f_1(\mathbf{x}_1), s) \vee holds([\neg]f_2(\mathbf{x}_2), s) \vee \ldots \vee holds([\neg]f_m(\mathbf{x}_m), s). \tag{11}$$

In this sentence, $\rho(\mathbf{x}_r)$ represents a formula that does not contain any situation terms. We call this form semi-clausal because $\rho(\mathbf{x}_r)$ is not necessarily a literal. Here, each f_i is a fluent function symbol. The notation $[\neg]$ signifies that the symbol $\neg$ may or may not appear in front of the fluent term. We write $holds(\neg R, s)$ as abbreviation for $\neg holds(R, s)$. In order to simplify variable substitutions, we require that the tuples $\mathbf{x}_1 \ldots \mathbf{x}_m$ of variables be disjoint, and the tuple $\mathbf{x}_r$ include the variables in $\mathbf{x}_1 \ldots \mathbf{x}_m$. This requirement is not as restrictive as it may seem. For example, if we have the following state constraint:

$$holds(onHand(x), s) \vee holds(onTable(x), s),$$

we conform to the requirement by rewriting it as:

$$x_1 = x_2 \supset holds(onHand(x_1), s) \vee holds(onTable(x_2), s).$$

In the future, we will use the following:

Definition 2.1 *A state constraint of the form (11) is called an m-ary state constraint.*

Our goal in this section is to show that for the simplest case in which there are no state constraints with more than two *holds* literals (i.e., binary constraints), it is easy to derive a solution to the frame problem based on a set of successor state axioms. Thus, we provide a mechanical procedure to generate a set of successor state axioms to replace the effect axioms and the binary state constraints. We prove that the successor state axioms generated satisfy the properties necessary to apply Lin and Reiter's result discussed in the previous section.

The solution is derived as follows: We start with a theory Σ_{sc_2}:

$$\Sigma_{sc_2} = \Sigma_{bd} \cup T_{pos} \cup T_{una} \cup T_{S_0} \cup T_{ef} \cup T_{sc_2},$$

where T_{sc_2} is a set of binary state constraints. The first step is to find an alternative formulation of this theory without state constraints. In fact, we derive a set $T'_{ef}(T_{sc_2}, T_{ef})$ of new effect axioms and form the theory:

$$\Sigma'_{sc_2} = \Sigma_{bd} \cup T_{pos} \cup T_{una} \cup T_{S_0} \cup T_{ef} \cup T'_{ef}(T_{sc_2}, T_{ef}).$$

Later, we prove that if Σ_{sc_2} is consistent, then the theory Σ'_{sc_2} is equivalent to Σ_{sc_2}, given a simple condition on the initial situation. In the next step, we use Σ'_{sc_2} to derive a set of successor state axioms $T_{ss}(T'_{ef}(T_{sc_2}, T_{ef}))$. Thus, we finally obtain the theory:

$$\Sigma_{ss_2} = \Sigma_{bd} \cup T_{pos} \cup T_{S_0} \cup T_{una} \cup T_{ss}(T'_{ef}(T_{sc_2}, T_{ef})).$$

The set $T_{ss}(T'_{ef}(T_{sc_2}, T_{ef}))$ is derived directly following Reiter's approach to the frame problem in the absence of state constraints, described in section 2.2. Finally, we use theorem 2.1 to show that Σ_{ss_2} constitutes a solution to the frame and ramification problem for the original theory.

Thus, we describe the procedure to obtain $T'_{ef}(T_{sc_2}, T_{ef})$ from a theory Σ_{sc_2}:
Let

$$\rho(\mathbf{x}_r) \vee holds([\neg]f_1(\mathbf{x}_1), s) \vee holds([\neg]f_2(\mathbf{x}_2), s)$$

be an arbitrary binary state constraint in T_{sc_2}. In order to simplify the presentation, we describe how to obtain new effect axioms for state constraints of the form:

$$\rho(\mathbf{x}_r) \vee holds(f_1(\mathbf{x}_1), s) \vee holds(f_2(\mathbf{x}_2), s).$$

Thus, we ignore the case in which we have negative *holds* literals. However, the same discussion applies to constraints for negative literals. The only change is that if $f(\mathbf{x})$ is a fluent term with a $\neg$ in front, i.e. $\neg f(\mathbf{x})$, we reverse the roles of *negative* and *positive*.

Each negative effect axiom for f_1 in T_{ef} is of the form[5]:

$$Poss(a, s) \wedge \gamma^-_{f_1}(a, s) \supset \neg holds(f_1(\mathbf{x}_1), do(a, s)).$$

For each such axiom, we generate the following positive effect axiom for f_2:

$$Poss(a, s) \wedge \gamma^-_{f_1}(a, s) \wedge \neg\rho(\mathbf{x}_r) \supset holds(f_2(\mathbf{x}_2), do(a, s)). \tag{12}$$

Symmetrically, if we have an effect axiom:

$$Poss(a, s) \wedge \gamma^-_{f_2}(a, s) \supset \neg holds(f_2(\mathbf{x}_2), do(a, s)),$$

we add:

$$Poss(a, s) \wedge \gamma^-_{f_2}(a, s) \wedge \neg\rho(\mathbf{x}_r) \supset holds(f_1(\mathbf{x}_1), do(a, s)). \tag{13}$$

Notice that (12) and (13) are entailments of the original theory Σ_{sc_2}. The previous procedure is used with each and every state constraint, and is repeated until no new effect axioms can be generated.

Now, we have the following:

[5]It may be necessary to rename the variables in the effect axiom and state constraints to make the f_1 fluent terms match. Since this renaming operation is straightforward, we ignore it to make the presentation more concise.

Proposition 2.2 *Let*

$$\Sigma_{sc_2} = \Sigma_{bd} \cup T_{pos} \cup T_{una} \cup T_{S_0} \cup T_{ef} \cup T_{sc_2},$$

be a theory of action as described before, and let

$$\Sigma'_{sc_2} = \Sigma_{bd} \cup T_{pos} \cup T_{una} \cup T_{S_0} \cup T_{ef} \cup T'_{ef}(T_{sc_2}, T_{ef}),$$

be the theory obtained from Σ_{sc_2}, in which the set T_{sc_2} is replaced with the new effect axioms $T'_{ef}(T_{sc_2}, T_{ef})$, according to the procedure described above. Then, the following holds[6]*:*

$$\Sigma'_{sc_2} \models T_{sc_2}(S_0) \supset (\forall s).S_0 \preceq s \supset T_{sc_2}(s).$$

Therefore, given a theory Σ_{sc_2}, if we replace the binary state constraints with the set $T'_{ef}(T_{sc_2}, T_{ef})$ we obtain a theory that entails the state constraints whenever the initial state satisfies them. This result is proven by induction[7], using $\varphi(s) = S_0 \preceq s \supset T_{sc_2}(s)$.

The proposition establishes that we can replace the binary state constraints with a new set of effect axioms and the state constraints applied to the initial state only. Thus, Σ'_{sc_2} is a theory suitable for the application of Reiter's solution to the frame problem. Hence, we obtain:

$$\Sigma_{ss_2} = \Sigma_{bd} \cup T_{pos} \cup T_{una} \cup T_{S_0} \cup T_{sc_2}(S_0) \cup T_{ss}(T'_{ef}(T_{sc_2}, T_{ef})), \tag{14}$$

which constitutes the solution to the frame and ramification problems for the original theory Σ_{sc_2}. In order to guarantee that this is a solution to the problem, we have the following:

Theorem 2.2 *The set Σ_{ss_2} (as in (14)) is a solution to the frame and ramification problem for the theory Σ_{sc_2}, as long as the following consistency condition is satisfied:*

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

where Ψ_f and $\Psi_{\neg f}$ are as defined in the statement of theorem 2.1.

This theorem follows directly from theorem 2.1 and proposition 2.2.

Example: To illustrate how to derive the successor state axioms from a basic theory of action, consider the following example[8]:

- Σ_d (situation independent axioms):

$$(\forall d).d = Saucer \vee d = Cup.$$

- T_{pos}:

$$Poss(pickup(d), s) \equiv \neg(\exists d')holds(inHand(d'), s),$$
$$Poss(drop(d), s) \equiv holds(inHand(d), s).$$

[6] Recall that $T_{sc_2}(s)$ denotes the conjunction of all the state constraints applied to the situation term s.

[7] See appendix.

[8] The reader should keep in mind that the predicate *holds* refers to a fluent holding in a situation, as opposed to an agent holding an object.

- T_{ef}:

$$Poss(pickup(d),s) \supset holds(inHand(d),do(pickup(d),s)),$$
$$Poss(drop(d),s) \supset \neg holds(inHand(d),do(drop(d),s)).$$

- T_{sc}:

$$holds(inHand(d),s) \vee holds(onFloor(d),s),$$
$$\neg holds(inHand(d),s) \vee \neg holds(onFloor(d),s).$$

- T_{S_0}:

$$holds(onFloor(Saucer),S_0),$$
$$holds(onFloor(Cup),S_0),$$
$$\neg holds(inHand(Cup),S_0),$$
$$\neg holds(inHand(Saucer),S_0).$$

Therefore, following the procedure outlined before, we can derive the following new effect axioms:

- $T'_{ef}(T_{sc},T_ef)$:

$$Poss(pickup(d),s) \supset \neg holds(onFloor(d),do(pickup(d),s)),$$
$$Poss(drop(d),s) \supset holds(onFloor(d),do(drop(d),s)).$$

Thus, the new theory is built by replacing axiom set T_{sc} with the set:

- $T_{ss}(T'_{ef}(T_{sc},T_ef))$:

$$Poss(a,s) \supset holds(inHand(d),do(a,s)) \equiv$$
$$a = pickup(d) \vee holds(inHand(d),s) \wedge \neg a = drop(d),$$
$$Poss(a,s) \supset holds(onFloor(d),do(a,s)) \equiv$$
$$a = drop(d) \vee holds(onFloor(d),s) \wedge \neg a = pickup(d).$$

It is simple to check that $T_{sc}(S_0)$ is satisfied by $\Sigma_d \cup T_{S_0}$. Furthermore, the condition:

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

gets translated into:

$$T_{una} \models (\forall a,d)\ \neg(a = pickup(d) \wedge a = drop(d)),$$

which is trivially true.

An advantage of theories of action described using successor state axioms for each fluent is that they lend themselves to easy update procedures. For instance, if we know the state of a situation s, then determining whether or not a fluent f holds in a successor situation $do(a,s)$ depends only on the evaluation of the simple formulas Ψ_f and

$\Psi_{\neg f}$. Thus, the computational cost of updating the value of a fluent equals the cost of determining the value of the Ψ formulas.

These theories are also interesting because extending them to allow for more general disjunctive state constraints (i.e., m-ary state constraints with $m > 2$) might be impossible. To exemplify this impossibility, consider a state constraint that states that:

$$holds(f_1, s) \vee holds(f_2, s) \vee holds(f_3, s).$$

Furthermore, assume that A is an action with a single effect axiom and no preconditions (i.e., A is always possible). The effect axiom is:

$$Poss(A, s) \supset \neg holds(f_1, s)$$

Thus, if A is performed in a state in which f_1 holds and f_2 and f_3 do not, the successor state cannot be uniquely determined. Thus, the effects of A are indeterminate. This violates a basic assumption of applicability of Reiter's method to solve the frame problem.

3 Occurrences in the Situation Calculus

The situation calculus is a temporal logic in which time is seen as a branching structure. Intuitively, the structure of situations is a tree rooted at the initial situation S_0, as illustrated by figure 1. Each branch that starts in S_0 can be understood as a *hypothetical future*. The tree structure of the situation calculus shows all possible ways in which the events in the world could unfold. Therefore, any arbitrary sequence of actions identifies a branch in the tree of situations.

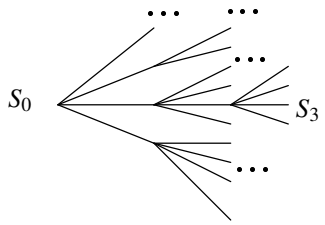


Figure 1: A tree of situations.

Thus, the situation calculus serves as a tool for the representation of *hypothetical knowledge.* As such it has several important drawbacks. In particular, it does not provide for ways to constrain the manner in which events occur. Also, it does not provide a notion of linear time. Therefore, propositions like "I will be home at three o'clock" cannot be represented in the language. This is due to the branching nature of the underlying structure of situations. In fact, the language of the situation calculus does not provide the elements to *prefer* one development of events over others. For example, in figure 1 we may know that situation S_3 is real, in the sense that the actions that lead to it from S_0 have occurred. However, the original language lacks the power to make such a statement.

In contrast, formalisms based on linear time provide ways in which to express definite temporal knowledge about a given domain. For example, not only can we state that "I am at the office but I will be home at three o'clock," but we can also conclude that I will have to perform some action to take me home, and that such an action will have to be performed between now and three o'clock. In the situation calculus, on the other hand, the most we can do is conclude that, to be home at three o'clock, I must perform some action or actions to take me there.

In this section we present Pinto and Reiter's extension to the language of the situation calculus by incorporating the basic elements of a linear temporal logic (Pinto and Reiter 1995). This is an extension, in the sense that new distinctions are introduced. In particular, we talk about a subset of situations that are *actual* and provide an axiomatization that describes its properties. The set of actual situations represents the situations that arise in the actual world. The path of actual situations is a rooted branch of the situation calculus tree. This extension does not impose any additional constraints on the situations. Thus, we retain the expressive power of the original situation calculus. The new distinctions introduced have a wide range of applicability. The main expressive enhancement is the ability to state that a sequence of events has occurred or will occur. One of the consequences of such an ability is that we can write sentences describing constraints among action occurrences. For example, we can write that one of the effects of some action is the triggering of other actions. Also, we may write that some actions have occurred without completely specifying which actions have occurred. The discussion that follows concentrates on non-concurrent actions.

3.1 An Actual Path of Situations

Here, we address the problem of describing the evolution of the world as it actually unfolds. To this end, we incorporate a predicate *actual* for situations. The intended meaning is that a situation is *actual* if it lies on the path that describes the world's true evolution. The axioms for *actual* are:

$$actual(S_0), \tag{15}$$

$$(\forall a, s).actual(do(a, s)) \supset actual(s) \wedge Poss(a, s), \tag{16}$$

$$(\forall a_1, a_2, s).actual(do(a_1, s)) \wedge actual(do(a_2, s)) \supset a_1 = a_2. \tag{17}$$

These axioms express that the initial situation is always actual, and if a situation is actual, then its immediate predecessor must also be. Axiom (17) says that an actual situation has at most one actual successor situation. An important characteristic of actual situations is that they all lie on the same path (i.e., they constitute a *time line*), as the following shows:

Proposition 3.1 *From the basic axioms (section 2.1) and axioms (15)-(17) it follows:*

$$(\forall s_1, s_2).s_2 \leq s_1 \supset [actual(s_1) \supset actual(s_2)],$$

$$(\forall s_1, s_2).actual(s_1) \wedge actual(s_2) \supset s_1 < s_2 \vee s_2 < s_1 \vee s_1 = s_2.$$

We also introduce a notion of time, which allows us to establish a direct relationship between the situation calculus and linear temporal logics (e.g., the calculus of events (Kowalski and Sergot 1986; Pinto and Reiter 1993)). Intuitively,

each situation has a starting time and an ending time. During the time span of a situation no fluents change truth values[9]. We incorporate the sort $\mathcal{T}$, interpreted as a continuous time line, into our language. The sort $\mathcal{T}$ is considered isomorphic to the non-negative reals. Also, we introduce the functions *start* from $\mathcal{S}$ to $\mathcal{T}$, and *end*, from $\mathcal{S} \times \mathcal{A}$ to $\mathcal{T}$. Events/actions occur at the ending time of situations. This is captured by the following axioms[10]:

$$(\forall s, a)\ end(s,a) = start(do(a,s)), \tag{18}$$

$$(\forall a, s)\ start(s) < start(do(a,s)), \tag{19}$$

$$start(S_0) = 0. \tag{20}$$

Thus, each situation is labeled with a start time and these times begin at 0 in S_0 and increase monotonically away from the initial situation. Hence, we have the immediate consequence:

Proposition 3.2 *From (18)-(19) it follows that:*

$$(\forall s, s').s < s' \supset start(s) < start(s').$$

There exists an obvious asymmetry between the *end* and *start* functions. This is due to the temporal asymmetry of the situation calculus. In fact, any given situation has a *unique* past. Therefore, for any single situation, there exists a unique *start*. On the other hand, any situation has multiple *possible* futures. Each possible future is reached by performing different sequences of actions. Thus, the end of a situation will depend on which future we consider. Hence, the term $end(s,a)$ denotes the time that would end the situation s if action a were to be performed. Since each actual situation has a unique actual successor situation, it turns out that the end time of actual situations is determinate.

Occurrences are introduced as a relation between *event types* and situations. For example:

$$occurs(pickup(D), S),$$

says that the event of picking up D occurred in situation S. Occurrences are defined in terms of the actual path as follows:

$$occurs(a,s) \equiv actual(do(a,s)). \tag{21}$$

Thus, each actual situation s, after S_0, is related to a unique action that must have occurred and which leads to s.

In some cases, it might be convenient to establish a relationship between events/actions that occur and the time at which they occur (rather than the situation). For this purpose, we introduce a predicate $occurs_{\mathcal{T}} \subseteq \mathcal{A} \times \mathcal{T}$, defined as:

$$occurs_{\mathcal{T}}(a,t) \equiv (\exists s).occurs(a,s) \wedge start(do(a,s)) = t. \tag{22}$$

[9] This restricts the properties of the world that can be represented as fluents, for instance, the position of a moving ball cannot be represented by a fluent. This problem is addressed in section 4.

[10] In what follows, we overload the symbols $<$ and $\leq$ used as relations between situations (as defined earlier), as well as for the standard ordering relation between the elements of $\mathcal{T}$.

Also, we define a relation $holds_{\mathcal{T}}$ between fluents and time points and a relation *during* between time points and situations:

$$holds_{\mathcal{T}}(f,t) \equiv (\exists s).actual(s) \wedge during(t,s) \wedge holds(f,s), \tag{23}$$

$$during(t,s) \equiv actual(s) \wedge start(s) < t \wedge (\forall a)\ [occurs(a,s) \supset end(s,a) \geq t]. \tag{24}$$

We introduce the notion of an event occurring between situations as follows:

$$occursBet(a,s_1,s_2) \equiv (\exists s).s_1 < s < s_2 \wedge occurs(a,s). \tag{25}$$

We can also show that if s_1 and s_2 are actual situations, and nothing occurs between them, then one must be the result of performing some action on the other. This is stated in the following:

Proposition 3.3 *From the basic axioms (section 2.1) and axioms (15)-(25) it follows:*

$$actual(s_2) \wedge s_1 < s_2 \wedge \neg(\exists a')\ occursBet(a',s_1,s_2) \supset (\exists a)\ s_2 = do(a,s_1).$$

In the next section we consider theories that include sentences in which an ordering between situations is provided. For this purpose, we introduce the following:

Definition 3.1 *A formula $\mathcal{O}_<(x_1,x_2,\ldots,x_n)$ is an* **ordering formula** *if it only mentions the terms $x_1,x_2,\ldots,x_n$, does not include any quantifiers and all its literals are $<$ literals.*

For example, if $\mathcal{O}_<(x_1,x_2) = x_1 < x_2$ then $\mathcal{O}_<(s_1,s_2) = s_1 < s_2$ where $<$ refers to the ordering between situations, and $\mathcal{O}_<(t_1,t_2) = t_1 < t_2$ where $<$ refers to the less-than operator for the real numbers.

Now, we have the following:

Proposition 3.4 *From proposition A.1 and axioms (15)-(25):*

$$(\exists s_1,\ldots,s_n).occurs(A_1,s_1) \wedge \ldots \wedge occurs(A_n,s_n) \wedge \mathcal{O}_<(s_1,\ldots,s_n) \equiv$$
$$(\exists t_1,\ldots,t_n).occurs_{\mathcal{T}}(A_1,t_1) \wedge \ldots \wedge occurs_{\mathcal{T}}(A_n,t_n) \wedge \mathcal{O}_<(t_1,\ldots,t_n).$$

where $\mathcal{O}_<$ is an ordering formula.

The expanded ontology for the situation calculus introduced in this section provides the essential features of other temporal logics.

3.2 What Occurs?

3.2.1 Motivation

Adding the *occurs* predicate to the language does not introduce any complications in the case in which all the events that occur are completely determined. However, problems do arise when this is not the case. For instance, consider

the Yale Shooting Problem (Hanks and McDermott 1987); it is common to pose it as the problem of determining the truth value of the literal:

$$holds(Alive, do(Shoot, do(Wait, do(Load, S_0)))).$$

That is, we want to know whether a fluent holds in a completely determined situation.

This problem can be formulated in a different fashion. In particular, using the situation calculus extended with a time line, we can formalize it as follows:

$$Poss(Load, s) \wedge Poss(Shoot, s), \tag{26}$$

$$Poss(Unload, s) \equiv holds(Loaded, s), \tag{27}$$

$$Poss(a, s) \wedge a = Load \supset holds(Loaded, do(a, s)), \tag{28}$$

$$Poss(a, s) \wedge a = Shoot \wedge holds(Loaded, s) \supset \neg holds(Alive, do(a, s)), \tag{29}$$

$$Poss(a, s) \wedge a = Unload \supset \neg holds(Loaded, do(Unload, s)), \tag{30}$$

$$holds(Alive, S_0), \tag{31}$$

$$(\exists s_1, s_2, s_3, s_4).occurs(Load, s_1) \wedge occurs(Wait, s_2) \wedge occurs(Shoot, s_3) \wedge s_1 < s_2 < s_3 < s_4 \wedge actual(s_4). \tag{32}$$

Axioms (26) and (27) specify T_{pos}. (28)-(30) are the effect axioms. (31) establishes the initial conditions. (32) lists all the actions that we know have occurred and provides an ordering among the situations in which the actions occur.

Following Reiter's mechanism to derive the successor state axioms, we obtain:

$$\begin{aligned} Poss(a, s) \supset [&holds(Loaded, do(a, s)) \equiv \\ &a = Load \vee \neg a = Unload \wedge holds(Loaded, s)], \\ Poss(a, s) \supset [&holds(Alive, do(a, s)) \equiv \\ &\neg(a = Shoot \wedge holds(Loaded, s)) \wedge holds(Alive, s)]. \end{aligned}$$

Then, we may ask whether *Alive* holds after the shooting. By formulating the *YSP* in this manner, there are two distinct problems that have to be solved:

- First, we have the *frame problem* whose solution allows us to determine the effects of each action in the environment, given the state of the situation in which they are performed. Here, we assume that the frame problem is properly dealt with using Reiter's approach described in an earlier section. Thus, we assume that for each fluent we have a successor state axiom. It is important to point out that we assume that the frame problem is solved before we address the second problem described below. This requirement should be understood as giving higher priority to solving the frame problem.

- The second problem that we have to address is the one of determining exactly which actions occur. For instance, in the formulation of the *YSP* above, it could be the case that an *Unload* action occurred after the *Load* and

before the *Shoot*. Also, between two situations there may be an arbitrary number of unrelated actions that may occur. However, we would like to infer that these extra actions do not occur. To deal with this problem, we take the view that some sort of minimization has to be used. In fact, our approach to address this problem follows a similar intuition to the one that serves as the basis for Reiter's solution to the frame problem. That is, in the solution to the frame problem, it is assumed that each action has no more effects than those that necessarily follow from the effect axioms. In a similar spirit, we take the point of view that, after the frame problem is solved, no actions occur unless their occurrence is dictated by the axioms of the theory. Our conceptual approach to minimizing occurrences is similar to the one advanced by Stein and Morgenstern (1994) in their *motivated action theory*. Indeed, they use a preferential semantics to select the models of an action theory in which the fewest possible extraneous actions occur.

A very important issue arises when the frame problem is separated from the problem of determining the set of occurrences. To illustrate it, we introduce a simple abstract example. Consider a theory of action whose language includes a fluent constant F, situation constants S' and S'' and action constants A_1 and A_c. Also, assume that the only effect axiom is:

$$Poss(A_c, s) \supset \neg holds(F, do(A_c, s)).$$

Furthermore, assume that the theory satisfies the following sentence:

$$holds(F, S') \wedge \neg holds(F, S'') \wedge S' < S''.$$

That is, F has changed from S' to S''. We can easily conclude that some action must have occurred which negatively affected F (see proposition 2.1). So, assume that action A_1 is known to have occurred at S' or between S' and S'', that is:

$$occurs(A_1, S') \vee occursBet(A_1, S', S'').$$

Also, assume that there are no effect axioms relating A_1 to F. Now, given that this represents all we know about our domain, we have two possible ways of understanding the nature of the change in F:

1. Action A_1 has made F change negatively from S' to S''. Thus, A_1 affects F in a way that is not sanctioned by the effect axioms.

2. Some other action (e.g. A_c), whose occurrence we did not know about, occurred between S' and S'' and made F change from S' to S''.

Interestingly, the literature on the situation calculus and the frame problem is concerned only with the first sort of explanation. In fact, if all we know is that A_1 occurred, this is usually expressed as:

$$S'' = do(A_1, S').$$

Most non-monotonic solutions to the frame problem will conjecture that there is an unstated qualification which makes A_1 affect F negatively. In fact, non-monotonic solutions are concerned with explaining change due to A_1, and the only

possible explanation (given that other actions are not postulated) is that A_1 is responsible for the change. In contrast, Reiter's monotonic solution to the frame problem would lead to a contradiction when using 1 as hypothesis. This is so since in the absence of effect axioms relating F and A_1, we will obtain a successor state axiom stating that F never changes from a situation s to a situation $do(A_1, s)$.

An alternative to explaining change by assuming unstated qualifications to known actions (e.g., assuming that a *Wait* event may have unstated effects such as $Unload$ing guns) is to postulate the existence of intervening actions that explain the change. For instance, given the effect axiom that states that A_c always makes F false, Reiter's solution to the frame problem will allow for models in which such an action explains the changes in F. On the other hand, if there were no effect axioms relating actions to negative change in F, then using Reiter's solution to the frame problem would lead to a contradiction. This contradiction seems to be the only appropriate conclusion. In fact, if the theory does not mention any actions that would change F, then F should not change.

Shanahan (1993) advances a similar view to the one expressed here. Indeed, in reference to the *stolen car scenario* (Kautz 1986), he argues quite convincingly that one should characterize the sequence of actions that start at an initial situation (S_0) and end in some posterior situation for which one needs an explanation. In other words, if we know that our car is parked at some point, and we know that the car is not parked at a later time, we should conclude that some intervening event, that might make the car disappear, must have occurred. Thus, we should not explain the disappearance of the car as a strange effect of some action that happens to have occurred in between.

As has been described in this section, we follow the approach of *assuming* that we do not know all the actions that occur. However, we also assume that any action whose occurrence is not necessary to explain the information at hand does not occur. Thus, we divide the problem of reasoning about actions and change into two parts:

- First, given the state of a situation s, what is the state of a situation $do(a, s)$? This question is answered using Reiter's solution to the frame problem. Thus, we use *successor state axioms*.
- Second, given a possibly incomplete description of the actions that occur, what are the actions that occur? This is answered using the *non-intervening events* assumption. Thus, we choose models that contain minimal sets of occurring actions. This is formalized using circumscription.

In related work, Miller and Shanahan (1994) study the formalization of *narratives* within the situation calculus. Narratives are conceptually similar to ordered sets of actions that are known to have occurred. In their work, the first problem mentioned above is solved by appealing to Baker's circumscriptive solution to the frame problem (Baker 1991). The second problem is also solved using a circumscriptive policy. They choose to minimize a predicate $Happens$, which plays a conceptually similar role to the predicate $occurs_{\mathcal{T}}$.

3.2.2 The Preferential Policy

As mentioned before, we use a minimization policy in order to select models of a theory in which nothing occurs unless its occurrence is *necessary*. Thus, assume that Σ_{occ} is a theory of action with occurrences. Also, assume that

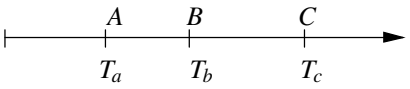


Figure 2: Time line in model $\mathcal{M}$.

Σ_{occ} has a model $\mathcal{M}$ in which all the actions that occur are as depicted in the time line of figure 2. Hence, in $\mathcal{M}$, the only action occurrences are A, B and C at times T_a, T_b and T_c respectively. Furthermore, assume that $\mathcal{M}'$ is another model of Σ_{occ} in which the occurrences are depicted in the time line of figure 3. In the model $\mathcal{M}'$, we have that the

Figure 3: Time line in model $\mathcal{M}'$.

only action occurrences are A and C at times T_a and T_c respectively. Our preferential semantics is such that the model $\mathcal{M}'$ should be preferred to $\mathcal{M}$. As partial justification for this semantics we will show that it yields the expected results when complete information can be assumed. Furthermore, in some cases this policy subsumes the policies stated by other researchers in the area. For example, Shanahan's assumption "that the only events which occur are those which are known to occur" (Shanahan 1995) is subsumed by our policy when the chronological ordering between the actions is known. Thus, if it is consistent to make Shanahan's assumption, then our policy yields the same results when the chronological ordering of the actions is known.

In order to formalize the minimization, we use *Circumscription* (McCarthy 1980; McCarthy 1986; Lifschitz 1985). To simplify the presentation of the circumscription policy, we consider the predicates *occurs*, $holds_{\mathcal{T}}$, *during*, and *occursBet* and the function *end* to be abbreviations (as defined by (21), (23), (24), (25) and (18)), which are not part of the extended language of the situation calculus. By doing this, we need not include these elements in the list of symbols whose interpretation is made variable in the circumscription policy.

In the following discussion, Σ_{occ} denotes a theory of action that includes:

- Axiom sets as in section 2.3, in which the sets T_{ef} and T_{sc} can be replaced with a set $T_{ss}(T_{ef}, T_{sc})$ of successor state axioms according to the criterion presented in section 2.3.
- Axioms (15)-(25).
- A set T_{occ} of occurrence axioms. These axioms are meant to constrain the interpretations of the actual line of situations. First, we consider the case in which T_{occ} includes axioms that directly constrain the the structure of the actual line of situations. These are of the form:

$$(\exists s_1, \ldots, s_n).occurs(A_1, s_1) \wedge \ldots \wedge occurs(A_n, s_n) \wedge \mathcal{O}_{<}(s_1, s_2, \ldots, s_n). \tag{33}$$

  where $\mathcal{O}_{<}$ is an ordering formula. Later we extend the classes of formulas contained in T_{occ}.

The circumscription policy is as follows:

$$Circ(\Sigma_{occ}; occurs_{\mathcal{T}}; actual, start), \tag{34}$$

Thus, we select the models of Σ_{occ} that have a minimal extension for the predicate $occurs_{\mathcal{T}}$. Below, we present a precise semantic characterization of (34). The predicate *actual* and the function *start* are variable elements in the circumscription. Clearly, these elements need to vary since $occurs_{\mathcal{T}}$ is defined in terms of them. Notice that the situation tree is fixed except for the elements that determine what the actual line looks like.

In the examples that follow, we make use of model-theoretic arguments to justify our conclusions. Therefore, based on Lifschitz's results (Lifschitz 1985), we present the model-theoretic meaning of the above circumscription.

Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two arbitrary models of a theory of action Σ_{occ}. Model $\mathcal{M}_1$ is preferred to model $\mathcal{M}_2$, written $\mathcal{M}_1 < \mathcal{M}_2$ iff:

1. $||\mathcal{M}_1|| = ||\mathcal{M}_2||$. That is, $\mathcal{M}_1$ and $\mathcal{M}_2$ share the same domain.

2. The interpretation for every function symbol other than *start*, and the interpretation of every predicate other than *actual* and $occurs_{\mathcal{T}}$ is the same.

3. $occurs_{\mathcal{T}}^{\mathcal{M}_1} \subset occurs_{\mathcal{T}}^{\mathcal{M}_2}$. That is, the extension of the predicate $occurs_{\mathcal{T}}$ in $\mathcal{M}_1$ is a proper subset of the extension of $occurs_{\mathcal{T}}$ in $\mathcal{M}_2$.

A model $\mathcal{M}$ of Σ_{occ} is minimal with respect to $<$, if there is no other model $\mathcal{M}'$ of Σ_{occ} such that $\mathcal{M}' < \mathcal{M}$. Finally, from Lifschitz result, it follows that $\mathcal{M}$ is a model of

$$Circ(\Sigma_{occ}; occurs_{\mathcal{T}}; actual, start)$$

if and only if $\mathcal{M}$ is a minimal model of Σ_{occ} according to $<$ as defined previously.

In what follows we introduce theories that contain Skolemized versions of axioms of the form (33). In these cases, we replace (34) with:

$$Circ(\Sigma_{occ}; occurs_{\mathcal{T}}; actual, start, \mathbf{S_k}). \tag{35}$$

Where $\mathbf{S_k}$ denotes the set of Skolem constants introduced.

As mentioned before, we claim that if we have complete information about the actions that occurred then the circumscription policy gives the expected results. This is formally stated as follows:

Observation 3.1 *If Σ_{occ} is a theory in which the set T_{occ} contains a single axiom of the form:*

$$occurs(A_1, S_1) \wedge \ldots \wedge occurs(A_n, S_n) \wedge S_1 < \ldots < S_n. \tag{36}$$

and the theory Σ_{occ} has a model $\mathcal{M}$ that satisfies:

$$occurs(a, s) \equiv a = A_1 \wedge s = S_1 \vee a = A_2 \wedge s = S_2 \vee \ldots \vee a = A_n \wedge s = S_n. \tag{37}$$

Then, every model of $Circ(\Sigma_{occ}; occurs_{\mathcal{T}}; actual, start, \mathbf{S_k})$ *will satisfy (37) and will also satisfy*

$$S_n = do(A_{n-1}, S_{n-1}) \wedge \ldots \wedge S_2 = do(A_1, S_1) \wedge S_1 = S_0. \tag{38}$$

The importance of this observation is that it establishes that if it is consistent to assume that all the actions that are known to occur are the only ones that occur, and these actions are totally ordered, then the models of the circumscription will not include any other occurrences. Furthermore:

Observation 3.2 *Let* Σ_{occ} *be a theory in which* T_{occ} *contains a single sentence of the form:*

$$occurs(A_1, S_1) \wedge \ldots \wedge occurs(A_n, S_n) \wedge \mathcal{O}_{<}(S_1, S_2, \ldots, S_n).$$

Where $\mathcal{O}_{<}$ *is an ordering formula. Assume that there is a model* $\mathcal{M}$ *of* Σ_{occ} *that satisfies:*

$$occurs(a, s) \equiv a = A_1 \wedge s = S_1 \vee a = A_2 \wedge s = S_2 \vee \ldots \vee a = A_n \wedge s = S_n \tag{39}$$

Then, $\mathcal{M}$ *is a model of:*

$$Circ(\Sigma_{occ}; occurs_{\mathcal{T}}; actual, start, \mathbf{S_k}). \tag{40}$$

However, it is not necessarily the case that for any model $\mathcal{M}'$ *of (40)* $\mathcal{M}'$ *satisfies (39).*

This property of our minimization policy is extremely important. In fact, it serves to illustrate the fact that it does not necessarily correspond to the intuition that *if it is consistent to assume that no occurrences other than the known ones exist, then there are no other occurrences.* As an example, consider a situation calculus language with action constants A, B and C, fluent F_c and a theory Σ_{occ} with the following axioms:

- Axioms about *Poss*:

  $$Poss(B, s) \equiv holds(F_c, s)$$
  $$Poss(A, s) \wedge Poss(C, s).$$

  Thus, B is possible only if F_c is true. A and C are always possible.

- Successor state axioms:

  $$Poss(a, s) \supset [holds(F_c, do(a, s)) \equiv (holds(F_c, s) \wedge \neg a = A)]. \tag{41}$$

  Thus, F_c is falsified by action A and is otherwise unaffected.

- Initial state:

  $$holds(F_c, S_0).$$

- T_{occ}:

  $$occurs(A, S_1) \wedge occurs(B, S_2).$$

  Thus, we know that A and B have occurred, but we don't know in which order. Here, S_1 and S_2 are Skolem constants.

Clearly, $\Sigma_{occ} \models S_2 < S_1$. Thus, any preferred model will satisfy this ordering.

Now, let us change the example by replacing the successor state axiom (41) with:

$$Poss(a,s) \supset [holds(F_c, do(a,s)) \equiv a = C \vee (holds(F_c,s) \wedge \neg a = A)]. \tag{42}$$

Thus, C changes F_c positively. In this situation, there are models of Σ_{occ} that satisfy the ordering $S_1 < S_2$. Therefore, the previous proposition establishes that there will be models of the circumscription in which A occurred before B. It is not difficult to see that in these models C occurs in between. Thus, this is an example in which it is not the case that the circumscriptive theory entails that the known actions are all the actions that occur, even if it is consistent to assume so. Notice that there is a model in which $S_2 < S_1$ and C does not occur in it.

In our view, it would be incorrect to force an ordering among actions to ensure that fewer actions occur. For example, assume that we are told that: *John and Mary will meet at their place at some time. If Mary gets there before John, then she will call Sue*. If this is all that is known we do not want to infer that John will get home first just to enforce a minimization of actions (disallowing models in which Mary makes her call).

In the previous discussion we have shown that the preferential policy that we presented gives the correct answers when the set T_{occ} of occurrence axioms are of the form (36). Given proposition 3.4, this is also the case when T_{occ} is a sentence of the form:

$$(\exists t_1, \ldots, t_n).occurs_{\mathcal{T}}(A_1, t_1) \wedge \ldots \wedge occurs_{\mathcal{T}}(A_n, t_n) \wedge \mathcal{O}_{<}(t_1, \ldots, t_n).$$

when $\mathcal{O}_{<}$ provides a complete order. Unfortunately, sentences of this form are too restrictive and do not provide the expressiveness required by many examples (as shown below). Thus, we would like to be able to introduce a more general kind of occurrence statements in the set T_{occ}. Now, given such a set of sentences, we need some argument to the effect that the preferential policy presented gives the right answers. The only argument that we provide is the fact that the minimization policy selects those models which seem *intuitively* better in a set of examples. Unfortunately, this is unsatisfactory. A completely satisfactory solution to this problem would be to present some *independent justification* for the policy utilized. However, *independent justifications* are hard to come by. Thus, as other researchers in theories of action, we must use examples that would provide *evidence* for the correctness of the solution proposed. This we do in the next section.

Aside from axioms of the form (33), we will use axioms stating conditional occurrences. This axioms are of the form:

$$\Phi(s,a) \supset (\exists s').s \leq s' \wedge occurs(a,s'), \tag{43}$$

where $\Phi(s,a)$ is a simple state formula. Conditional occurrence axioms of this form provide further constraints on the actual line of occurrences. These axioms can also be written by utilizing situation free sentences that contain time terms. For example:

$$holds(Hungry,t) \supset (\exists t').t < t' \wedge occurs_{\mathcal{T}}(Eat,t')$$

states that if in the line of actual situations there is a time in which the fluent constant $Hungry$ holds, then there must be an action Eat in the actual line after the time at which $Hungry$ holds.

Finally, we also use axioms of the form:

$$(\exists t_1, \ldots, t_n).occurs_{\mathcal{T}}(A_1, t_1) \wedge \ldots \wedge occurs_{\mathcal{T}}(A_n, t_n) \wedge \mathcal{O}'(t_1, \ldots, t_n).$$

In which $\mathcal{O}'(t_1, \ldots, t_n)$ is an extended ordering formula that mentions temporal constants aside from the terms $t_1, \ldots, t_n$.

In the next section we apply our approach to several examples from the literature. In particular, we show that the intended conclusions for the YSP example above are indeed what follows from the circumscription.

3.2.3 Examples

In this section we discuss two examples. These examples are taken from Sandewall's test suite (Sandewall 1995). Each problem is formalized as a circumscriptive theory (35) and differs in the domain axioms that are specified with each example.

The YSP The Yale Shooting problem (Hanks and McDermott 1987) was presented in the previous section. The relevant axioms we repeat here in a Skolemized form:

$$Poss(Load, s) \wedge Poss(Shoot, s), \tag{44}$$

$$Poss(Unload, s) \equiv holds(Loaded, s), \tag{45}$$

$$holds(Alive, S_0), \tag{46}$$

$$occurs(Load, S_1) \wedge occurs(Wait, S_2) \wedge occurs(Shoot, S_3), \tag{47}$$

$$S_1 < S_2 < S_3 < S_4, \tag{48}$$

$$actual(S_4), \tag{49}$$

$$Poss(a, s) \supset [holds(Loaded, do(a, s)) \equiv a = Load \vee \neg a = Unload \wedge holds(Loaded, s)], \tag{50}$$

$$Poss(a, s) \supset [holds(Alive, do(a, s)) \equiv \neg(a = Shoot \wedge holds(Loaded, s)) \wedge holds(Alive, s)]. \tag{51}$$

The objective of the YSP is to determine whether or not $Alive$ holds after the shooting. First, it is not difficult to see that there is a model $\mathcal{M}_{int}$ of the circumscription such that:

$$\mathcal{M}_{int} \models S_1 = S_0 \wedge S_2 = do(Load, S_1) \wedge S_3 = do(Wait, S_2) \wedge S_4 = do(Shoot, S_3).$$

Furthermore,

$$\mathcal{M}_{int} \models \neg holds(Alive, S_4).$$

However, we need to show that there is no model of the circumscriptive theory that satisfies $holds(Alive, S_4)$. We do so by contradiction. That is, assume that there is a model $\mathcal{M}$ such that

$$\mathcal{M} \models holds(Alive, S_4)$$

is true.

From the *occurs* and ordering statements (47)-(48), we infer that $\mathcal{M}$ satisfies:

$$(\exists t_1, t_2, t_3).occurs_{\mathcal{T}}(Load, t_1) \wedge occurs_{\mathcal{T}}(Wait, t_2) \wedge occurs_{\mathcal{T}}(Shoot, t_3) \wedge t_1 < t_2 < t_3.$$

By induction and using the successor state axiom for *Alive* (51), it follows that $\mathcal{M}$ satisfies:

$$(\forall s, s').s' > s \wedge \neg holds(Alive, s) \supset \neg holds(Alive, s').$$

That is, no resuscitations are possible. Therefore, if *Alive* holds in S_4, it follows that *Alive* holds in all previous situations. In particular, it follows that $holds(Alive, do(Shoot, S_3))$. Therefore, the successor state axiom for Alive tells us that $holds(Loaded, S_3)$ must be false. Using the successor state axiom for *Loaded* it follows that an $Unload$ action must have occurred between the *Load* and the *Shoot*. So, we obtain:

$$\begin{aligned}\mathcal{M} \models (\exists t_1, t_2, t_3, t_4).&occurs_{\mathcal{T}}(Load, t_1) \wedge occurs_{\mathcal{T}}(Wait, t_2) \wedge \\ &occurs_{\mathcal{T}}(Shoot, t_3) \wedge occurs_{\mathcal{T}}(Unload, t_4) \wedge \\ &t_1 < t_2 < t_3 \wedge t_1 < t_4 < t_3.\end{aligned}$$

Thus, in $\mathcal{M}$, $Unload$ occurred some time between the *Load* and the *Shoot*. However, from this model, we can easily obtain a model $\mathcal{M}'_{int}$ (e.g., $\mathcal{M}_{int}$, described previously) without the $Unload$ occurrence and that preserves the rest of the interpretation. Essentially, we only change the actual line of situations from one branch to another. Clearly, $\mathcal{M}'_{int} < \mathcal{M}$, therefore, $\mathcal{M}$ cannot be a model of (35). It follows that in all the models of the circumscription, *Alive* does not hold in S_4.

Ferry-boat Connection Scenario This example was first proposed in (Sandewall 1991). It is as follows:

> "A motorcycle is initially driving along a road on island Fyen, in the direction of a ferry-boat landing. The ferry-boat departs at time 100. If the motorcycle reaches the ferry-boat landing before time 100, it will be in Jutland from time 110, otherwise it stays. The motorcycle is known to reach the landing sometime between time 99 and 101."

This problem serves as a test to determine how well our language allows the expression of temporal relations between events. We extend the language with a fluent function *At* with one argument. We fix the domain objects as those denoted by *Fyen, Landing, Boat* and *Jutland*, which we assume to be distinct. Aside from the four primitive fluents $At(Fyen)$, $At(Landing)$, $At(Boat)$ and $At(Jutland)$, we introduce the fluents *Gone* and *Arrived*. The fluent *Gone* holds as a consequence of the departure of the ferry. The fluent *Arrived* holds after the ferry-boat arrives in Jutland. Also, we introduce a two-place action function *move*, such that $move(d_1, d_2)$ would denote the action of moving from location d_1 to location d_2. Other actions are denoted by the constants *Depart* and *Arrive*. The axioms are as follows:

- T_{sc}:

$$d_1 \neq d_2 \supset \neg holds(At(d_1), s) \vee \neg holds(At(d_2), s). \tag{52}$$

- T_{pos}:

$$Poss(move(d_1, d_2), s) \equiv holds(At(d_1), s) \wedge \tag{53}$$
$$[d_1 = Boat \supset holds(Arrived, s)] \wedge [d_2 = Boat \supset \neg holds(Gone, s)],$$
$$Poss(Arrive, s) \equiv holds(Gone, s) \wedge \neg holds(Arrived, s), \tag{54}$$
$$Poss(Depart, s) \equiv \neg holds(Gone, s). \tag{55}$$

- T_{ef}:

$$Poss(Arrive, s) \supset holds(Arrived, do(Arrive, s)), \tag{56}$$
$$Poss(Depart, s) \supset holds(Gone, do(Depart, s)), \tag{57}$$
$$Poss(move(d_1, d_2), s) \supset holds(At(d_2), do(move(d_1, d_2), s)). \tag{58}$$

- T_{S_0}:

$$holds(At(Fyen), S_0) \wedge \neg holds(Gone, S_0) \wedge \neg holds(Arrived, S_0). \tag{59}$$

- T_{occ}:

$$occurs_{\mathcal{T}}(move(Fyen, Landing), T_1) \wedge 99 \leq T_1 \leq 101, \tag{60}$$
$$occurs_{\mathcal{T}}(Depart, 100), \tag{61}$$
$$occurs_{\mathcal{T}}(Arrive, 110), \tag{62}$$
$$holds_{\mathcal{T}}(At(Landing), t) \wedge t < 100 \supset \tag{63}$$
$$(\exists t').occurs_{\mathcal{T}}(move(Landing, Boat), t') \wedge t < t' < 100,$$
$$holds_{\mathcal{T}}(Arrived, t) \wedge holds_{\mathcal{T}}(At(Boat), t) \supset \tag{64}$$
$$(\exists t').t \leq t' \wedge occurs_{\mathcal{T}}(move(Boat, Jutland), t').$$

State constraint (52) asserts that the motorcycle cannot be in two places simultaneously. Following the procedure indicated in section 2.4, the effect axiom:

$$Poss(move(d_1, d_2), s) \wedge d_1 \neq d_2 \supset \neg holds(At(d_1), do(move(d_1, d_2), s)), \tag{65}$$

can be generated from this state constraint and the effect axiom (58). Therefore, we obtain the following successor state axiom:

$$Poss(a, s) \supset holds(At(d), do(a, s)) \equiv \tag{66}$$
$$a = move(d_1, d\,) \vee \neg(d_1 \neq d\ \wedge a = move(d, d_1)) \wedge holds(At(d), s),$$

which means that the motorcycle is at d after moving to d, or if it is at d and did not move elsewhere. The successor state axioms for *Gone* and *Arrived* are:

$$Poss(a,s) \supset holds(Arrived, do(a,s)) \equiv a = Arrive \vee holds(Arrived, s), \tag{67}$$

$$Poss(a,s) \supset holds(Gone, do(a,s)) \equiv a = Depart \vee holds(Gone, s). \tag{68}$$

Axiom (53) says that to move anywhere from d_1 the motorcycle has to be at d_1, and if the motorcycle is in the *Boat* then the move is only possible if the *Boat* has *Arrived*. Also, if the motorcycle moves to the *Boat*, then the *Boat* must not be *Gone*. Axiom (63) says that the motorcycle will board the boat if it is at *Landing* before the boat departs. Axiom (64) says that the motorcycle leaves the boat after the boat has arrived. Axioms (61) and (62) state that *Depart* and *Arrive* occur at times 100 and 110 respectively. Notice that *Depart* and *Arrive* can each occur only once. The other axioms are self explanatory.

Using our framework, we obtain two classes of minimal models. Each class is characterized by the sets of situations that are actual in each. In order to abbreviate the sentences that we will write below, we extend the language with a new set of situation constants S_1, S_2, S_3, S_4, S_5, S'_1, S'_2, and S'_3. We fix the interpretation of these situation constants[11]:

$$S_1 = do(move(Fyen, Landing), S_0), \tag{69}$$

$$S_2 = do(move(Landing, Boat), S_1), \tag{70}$$

$$S_3 = do(Depart, S_2), \tag{71}$$

$$S_4 = do(Arrive, S_3), \tag{72}$$

$$S_5 = do(move(Boat, Jutland), S_4), \tag{73}$$

$$S'_1 = do(Depart, S_0), \tag{74}$$

$$S'_2 = do(move(Fyen, Landing), S'_1), \tag{75}$$

$$S'_3 = do(Arrive, S'_2). \tag{76}$$

As we just said, there are two classes of models. In the first class of models, we have:

$$actual(s) \equiv s = S_0 \vee s = S_1 \vee s = S_2 \vee s = S_3 \vee s = S_4 \vee s = S_5, \tag{77}$$

$$start(S_1) < start(S_2) < start(S_3) < start(S_4) < start(S_5). \tag{78}$$

Whereas the second class of models is characterized by:

$$actual(s) \equiv s = S_0 \vee s = S'_1 \vee s = S'_2 \vee s = S'_3, \tag{79}$$

$$start(S'_1) < start(S'_2) < start(S'_3). \tag{80}$$

Obviously the ordering (78) and (80) of the temporal constants is derivable from the equality relationships defined with the axioms (69)-(76).

[11] These constants are introduced as abbreviations to avoid writing very long sentences that involve *do* terms with many other nested *do* terms. Since the interpretation for these constants is fixed, its participation in the circumscription policy as fixed or variable is of no consequence.

The objective of this example is to show that the circumscriptive theory characterized by (35) entails that either the first set of sentences (77)-(78) is true, or the second set of sentences (79)-(80) is true. We do so by showing that there are minimal models that satisfy each set of sentences. Then we show that any model that does not satisfy either set of sentences cannot be minimal.

Observation 3.3 *There exists a minimal model $\mathcal{M}$ that satisfies sentences (77)-(78).*

$\mathcal{M}$ may be obtained by setting $T_1 = start(S_1) = 99$, $start(S_2) = 99.5$, $start(S_3) = 100$, $start(S_4) = 110$, $start(S_5) = 111$. Clearly, there is a model in which the actual line is such that these temporal constrains are satisfied. This model is minimal with respect to the circumscription policy. In fact, if we eliminate any element from the extension of $occurs_{\mathcal{T}}$ we violate one of the axioms.

Observation 3.4 *There exists a minimal model $\mathcal{M}'$ that satisfies sentences (79)-(80).*

For the second class of models, we set $start(S'_1) = 100$, $start(S'_2) = T_1 = 101$, $start(S'_3) = 110$. Again, it is easy to check that a model $\mathcal{M}'$ that satisfies this constraints exists. Also, this is a minimal model, since eliminating any one of the elements of the extension of $occurs_{\mathcal{T}}$ leads to a non-model.

Finally:

Observation 3.5 *Any minimal model must satisfy either (77)-(78) or (79)-(80).*

This can be easily checked by realizing that all models include the occurrences of *Depart* and $move(Fyen, Landing)$. The only ambiguity with respect to these events is which occurs first. There are only two possibilities (we assume that they may not occur concurrently). From each possibility, a different class of models arise. First, assume that $\mathcal{M}$ is an arbitrary model in which $move(Fyen, Landing)$ occurs before *Depart*. Thus, in this model $T_1 < 100$, given that the structure of time is dense and from proposition 3.2 it follows that this model satisfies:

$$(\exists\, s,t)\ occurs(move(Fyen, Landing), s) \wedge start(do(move(Fyen, Landing), s)) = T_1 \wedge$$
$$during(t, s) \wedge T_1 < t < 100. \tag{81}$$

Therefore, the left hand side of (63) is satisfied. Hence, the motorcycle moves to the *Boat*. Once in the *Boat*, no *move* actions are possible until the ferry-boat has arrived, as inferred from (53). Thus, (77)-(78) must hold in the minimal models arising from this case. Second, it is simple to check that if $\mathcal{M}'$ is a model in which $move(Fyen, Landing)$ occurs after *Depart*, then all the occurrences in the second class of models must arise. Hence, it is easy to see that the minimal models arising from this class must satisfy (79)-(80).

It is important to point out that the axiomatization of the problem is extremely simplified. For example, in a more general axiomatization, the *Poss* axioms for *move* actions should refer to the spatial interrelationships between the origin and destination. Also, the axiomatization forces the behavior of the ferry and the motor-cyclist. For example, after arriving at *Landing*, if the motor-cyclist is on time she must board. However, in a more general case, there is no guarantee that the motor-cyclist will indeed board the ferry once in *Landing*. In fact, without formalizing notions of agent's goals, desires, intentions, etc., we may not have any way to predict what will happen.

4 Continuity and Natural Events

An important drawback of the original situation calculus and some of its extensions is the impossibility of talking about properties that change continuously with time. This is the case with the discrete situation calculus, in which an induction axiom (e.g., (1)) is used to characterize the set of existing situations. For example, consider the problem of modeling the dynamic behavior of physical objects. To do so, we need a language to describe different properties of these objects. For instance, we may want to describe situations in which an object is falling. Thus, we may introduce a fluent term $falling(A)$ to denote the property *object A is falling*. Therefore, we may write $holds(falling(A), s)$ to mean that $falling(A)$ is true in some situation s. Also, assume that we want to describe the position of the object. Specifically, we may want to consider the property *object A is at height Y*. We might be tempted to write an expression like:

$$holds(height(A, Y), S),$$

as a sentence intended to mean that S is a situation in which A is at height Y. However, if A is falling, and we take height to be a real number, such a formula would be true at an instant of time only. For this to be logically correct, we would need S to span a single time point. Therefore we would need a different situation for each time point in which the object is falling. Since we take time to correspond to the non-negative real numbers, we would need a continuum of situations. This, of course, violates the induction axiom. Hence, the property of an object being in a certain position in space cannot be denoted by a fluent.

The previous problem does not arise if one gives up the notion that space-time has a continuous nature. For example, we may consider that coordinates are isomorphic to the integers, and that objects stay a positive and finite length of time at each location. In this approach, if an object falls it would pass through a finite set of points during any interval of time. However, this is unsatisfactory for several reasons. The most important one is that it introduces severe representational constraints. In fact, it does not adequately describe the physical phenomena we want to reason about. Now, assuming that we are willing to accept this representational limitation, the resulting framework is very artificial. In particular, since the ball would be changing position from one situation to the next, we would have to introduce actions to provoke these changes at every step.

Another possibility is to consider actions that have durations. For example, the falling of a ball can be considered to be an event that results in the ball hitting the ground. Such an approach greatly complicates the structure of the tree of situations. Furthermore, in this approach it is unclear how to consider occurrences of events while other events are in progress.

Instead, we propose a solution that involves dividing the properties of the world into two different classes. First, we have the properties modeled using standard fluents, which we also call *discrete fluents*. Second, we have the properties that vary continuously with time (e.g., the position of an object), which we call *continuous parameters*. Then, the property *object A is falling* will be considered a discrete fluent. As before, actions may be performed that would make discrete fluents change from one situation to another. For example, some action or event might produce a change that would make the discrete fluent *object A is at rest* true, and the property *object A is falling* false. On the other hand,

continuous parameters, such as the *position of object A* do not only change due to actions being performed by agents, but they also change due to the object's natural *inertia*. For example, if the ball in our example is falling, its position will be changing until some action that stops its motion is performed.

A further distinction between discrete fluents and continuous parameters is that the former hold or do not hold throughout situations, whereas continuous parameters may change their value continuously within a situation. This taxonomic division of properties of the world is not new. In fact, Galton (1990) proposes a similar distinction within the framework of Allen's theory of time (Allen 1984). Galton uses the terminology *state of motion* and *state of position* for properties that correspond to our discrete fluents and continuous parameters respectively. Also, related ideas are explored by Shanahan in the framework of the event calculus in logic programming (Shanahan 1990).

The majority of the problems that deal with properties that vary continuously with time are problems in the domain of physics. Since physics is concerned with building mathematical models for natural phenomena, we have to model actions or events that are considered *natural*. We use the term *natural event* or *natural action* to refer to events whose occurrence is determined by the laws of nature (physics). As an example, imagine that we start pumping a continuous flow of air into a rubber balloon. The inflating of the balloon will cause it to pop (a natural event), unless something happens to stop the flow of air. In other words, the laws of physics prescribe that when the pressure inside the balloon reaches a certain point, the balloon ought to pop. Also, we use the term *natural process* to refer to a totally ordered set of natural actions that occur in sequence. For example, if a ball is released, the set of bounces can be considered a natural process.

In contrast to what we call natural actions, the actions that have been considered in the previous section are *performed* by some *agent*. An important difference between agent-driven actions and natural events is that the former can be performed at any time (given that the preconditions for their execution are true). On the other hand, natural events cannot happen at an arbitrary time. Rather, they occur if and only if the environmental conditions dictate that their occurrence must arise. In this section we present a characterization of natural events. Clearly, a characterization of natural phenomena is essential to build systems for reasoning about physical processes.

Our interest is to provide a formal foundation for specifying dynamic systems. To do so, we present a mechanism to integrate continuous change into the situation calculus. An important advantage of our approach is that we keep the notions of *fluent* and *situation*. Thus, we maintain the view that fluents are properties that change from situation to situation. Also, situations are *discrete*. Therefore, the same approach to deal with the frame and ramification problems is applicable in the extended language presented here.

4.1 Extensions to the language

4.1.1 An Oracle

As mentioned in section 2, we work with an interpreted theory in which $\mathcal{R}$ has a fixed standard interpretation. Also, we use some standard functions for the reals, (i.e., $+$, $-$, $\times$, $/$, etc.) along with an infinite set of *numerical* constants (e.g. 0, -1, 3.14, etc.). Furthermore, we assume standard interpretations for these functions and constants.

Another important assumption we make here, is that we have an Oracle capable of answering questions about the truth or falsity of some sentences. For example, if in the course of a proof we need to show that:

$$(\exists k,t).t > 0 \wedge k > 0 \wedge k \times t = 12,$$

we assume that our Oracle will correctly respond *true*.

It may be argued that the Oracle introduced here is not really necessary. This is due to the fact that we are working with an interpreted theory in which the sort of real numbers has a fixed interpretation. Similarly, we assume that the real function symbols (e.g., $+, \times$, etc.) are also interpreted in the standard way. The reason we introduce this Oracle is that it allows us to cleanly separate the tasks of reasoning within the logic of the situation calculus and the task of reasoning about the reals.

We appeal to the Oracle to respond to questions regarding relationships between real numbers. For instance, solving systems of equations, and determining the existence of solutions to equations or inequalities. By using such an Oracle, we deliberately gloss over many issues regarding reasoning about equations and real numbers. This simplification allows us to concentrate on the extensions that are necessary for the situation calculus to deal with continuity.

In order to characterize the Oracle, we consider that we have:

- A language $\mathcal{L}_{\mathcal{R}}$, which is a sub-language of the situation calculus that contains the single sort $\mathcal{R}$ of real numbers. A set of real variables and constants, the binary function symbols $+, -, \times, / : \mathcal{R} \times \mathcal{R} \rightarrow \mathcal{R}$, the predicate symbol $< \subseteq \mathcal{R} \times \mathcal{R}$ and the equality symbol.

- A theory $\Sigma_{\mathcal{R}}$ in the language $\mathcal{L}_{\mathcal{R}}$ that characterizes the sort $\mathcal{R}$ of real numbers, along with the real operators $+, -, \times$, and $/$. Thus, we assume that the models of $\Sigma_{\mathcal{R}}$ correspond to the standard interpretations for the real numbers and the real operators mentioned above. Furthermore, let $\Phi(\mathbf{x})$ be an arbitrary quantifier free formula, in which $\mathbf{x}$ represents a tuple of real variables. All the free variables mentioned in the formula $\Phi(\mathbf{x})$ are in $\mathbf{x}$. We require that $\Sigma_{\mathcal{R}}$ be such that either:

$$\Sigma_{\mathcal{R}} \models (\exists \mathbf{x})\ \Phi(\mathbf{x}), \tag{82}$$

or

$$\Sigma_{\mathcal{R}} \models \neg(\exists \mathbf{x})\ \Phi(\mathbf{x}). \tag{83}$$

In this work we do not address the issue of writing down the theory $\Sigma_{\mathcal{R}}$. Rather, we assume that the theory is given. Finally, we assume that the Oracle is capable of answering whether (82) is true or (83) is true for any given formula $\Phi(\mathbf{x})$ that obeys the constraints mentioned earlier. Furthermore, we assume that if (82) is true, then the Oracle provides us with an assignment of elements of $\mathcal{R}$ to the variables in $\mathbf{x}$ that makes $\Phi(\mathbf{x})$ true.

Obviously, our entire approach to deal with continuity hinges on the expressive capabilities of the language $\mathcal{L}_{\mathcal{R}}$, as well as on the existence of this Oracle. We have taken the simplest possible language $\mathcal{L}_{\mathcal{R}}$ using only the basic real

operators. On the other hand, Oracles exist that are capable of answering our questions for a wide variety of classes of existential queries of the kind (82), for example, the Maple and Mathematica software systems.

As illustrated in the examples discussed later, we need to appeal to the Oracle in order to prove or disprove that certain sentences of the form $(\exists t)\ \phi(t)$ are true.

4.1.2 Ontological Extensions

Some properties of objects in the world vary continuously with time. For example, the temperature or position of an object. In order to refer to these properties, we introduce a class of objects called *parameters*. These parameters are used to name continuously varying properties. For example, *Pos* may be a parameter that names the position of some object along the X-axis in a Cartesian space. The actual real value of the property during a situation is described using a real function of time (e.g., $y(t) = 5 + 6 \times t$). Each such function will have a name (e.g., y). If during a situation s, a property named by a parameter p is behaving according to some function of time named by y we will write an expression stating that p is evaluated through y during situation s.

Formally, we introduce two new sorts in the language. First, we add a sort $\mathcal{P}$ for parameters, and the sort $\mathcal{Y}$ for names of real functions of time. We use the letters p and P for sort $\mathcal{P}$, and y and Y for sort $\mathcal{Y}$. These letters are used with or without subscripts or other markers. Furthermore, we introduce two new functions: First, the function $q : \mathcal{Y} \times \mathcal{T} \rightarrow \mathcal{R}$; such that $q(y,t)$ denotes the real number that results from evaluating the function y at time t[12]. Second, the function $\doteq$: $\mathcal{P} \times \mathcal{Y} \rightarrow \mathcal{F}$; such that $\doteq (p,y)$, which we use in infix notation as $p \doteq y$, denotes a fluent that holds when the parameter p behaves as described by the function named by y.

As a simple example, assume that we want to state that the height of some object is given by the function of time $5 + 6 \times t$ in some state S. Then, we may introduce the constant Y to name the function of time $5 + 6 \times t$, and the constant P_h to denote the parameter height. This is stated in the language as follows:

$$(\forall t)\ q(Y,t) = 5 + 6 \times t,$$
$$holds(P_h \doteq Y, S).$$

The first sentence states that the value of the function of time denoted by Y is $5 + 6 \times t$. The second sentence states that the object's height behaves according to Y during the situation S.

An important axiom for the function $\doteq$ is:

$$holds(p \doteq y_1, s) \wedge holds(p \doteq y_2, s) \supset (\forall t)\ q(y_1,t) = q(y_2,t). \tag{84}$$

Thus, if during a situation s a parameter's value varies according to two functions of time, then these functions must agree on their values at all times.

Since the function symbol $\doteq$ denotes a fluent function, a fluent $p \doteq y$ may change from situation to situation. For example, a golf ball may be following a trajectory whose height is given by a function $2 \times t - t^2$ during S_0. If it falls

[12] In (Sandewall 1989a), Sandewall proposes a similar approach to deal with functions of time.

in a sand trap, its height will be described by some constant. Therefore, the event of the ball falling into the trap has had the effect of changing the way in which the ball's height is described (e.g. from a parabolic function of time to a constant). On the other hand, if some other event occurs, the behavior of the ball may be unaffected. Thus, in the same manner as with traditional fluents, we need to describe how actions affect the behavior of parameters. This is addressed in the following section.

Also, we need to introduce the functions $+$, $-$, $\times$ and $/ : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathcal{Y}$. Once again, we overload symbols in the language and rely on the context to disambiguate them. These symbols are introduced to apply real operators to function names, with the meaning described by the following axioms:

$$q(y_1 + y_2, t) = q(y_1, t) + q(y_2, t), \tag{85}$$

$$q(y_1 - y_2, t) = q(y_1, t) - q(y_2, t), \tag{86}$$

$$q(y_1 \times y_2, t) = q(y_1, t) \times q(y_2, t), \tag{87}$$

$$q(y_1 / y_2, t) = q(y_1, t) / q(y_2, t). \tag{88}$$

Furthermore, we introduce the functions *constant*, *linear* and *quadratic* of sort $\mathcal{Y}$, with one, two and three real arguments respectively. We utilize these parameters to name real functions of time as follows:

$$q(constant(r_k), t) = r_k, \tag{89}$$

$$q(linear(r_k, r_r), t) = r_r + r_k \times t, \tag{90}$$

$$q(quadratic(r_k, r_r, r_a), t) = r_a + r_r \times t + 0.5 \times r_k \times t^2. \tag{91}$$

We also introduce *derivatives* with respect to time; we use the function $d : \mathcal{Y} \rightarrow \mathcal{Y}$, some of whose properties are:

$$d(y) = constant(0) \equiv (\exists\, r_k)\ y = constant(r_k), \tag{92}$$

$$d(y) = constant(r_k) \equiv (\exists\, r_r)\ y = linear(r_k, r_r), \tag{93}$$

$$d(y) = linear(r_k, r_r) \equiv (\exists\, r_a)\ y = quadratic(r_k, r_r, r_a). \tag{94}$$

These properties of the derivatives of polynomial functions could have been obtained from a set of general axioms about derivatives of real functions. However, this approach would take us too far afield. We prefer not to deal with these complications since they fall beyond the scope of this work. Other well known properties of the derivatives of functions could be added to the list (92)-(94). However, in our examples we only make use of these ones.

In the next section, we utilize the extension of the situation calculus that was just introduced to model natural processes. In particular, we show how to write effect axioms to describe the effects of natural events. Furthermore, we discuss how to derive successor state axioms for fluent terms of the form $p \doteq y$.

4.2 Representation of Natural Processes

4.2.1 Preliminaries

In the previous section, we extended the ontology of the situation calculus to include parameters and names for real functions of time. With these new sorts of objects we can model simple physical phenomena using equations on real quantities. We would like to incorporate those models into a logical theory based on the situation calculus extended with occurrences.

We consider only phenomena which can be modeled using deterministic mathematical models. These deterministic systems are such that, if we know the initial conditions, the complete future evolution of the system is described by the model. These mathematical models describe the behavior of *closed* systems; that is, systems in which there are no external perturbations. By integrating these models with the situation calculus we obtain a language in which we may describe the interactions between active agents and physical systems.

As an example, consider a billiards table scenario in which there are two balls. Assume that we know the initial position and parameters of each ball (i.e., their position, velocity, acceleration, etc.), and that we have an accurate mathematical model that tells us the behavior of the system at all future times. If the balls are moving towards each other, then a collision will occur and such a collision will change the way in which the balls move. When studying these systems, physicists perform "piece-wise" analysis. In the example, the trajectories of each ball would be described mathematically before and after the collision. Thus, the collision would be seen as an event that changes the parameters of the system.

Within the situation calculus framework, we follow the same approach. For example, assume that we place a Cartesian coordinate system in the billiards table. Also, assume that the X and Y axes coincide with two perpendicular sides of the table and that $X = 1$ and $Y = 1$ represent the other two sides of the table. Let X^i and Y^i denote the parameters *position of the ball i with respect to the X and Y axes* respectively. Furthermore, assume that the following are the initial conditions:

$$holds(Y^1 \doteq constant(0.5), S_0) \wedge holds(Y^2 \doteq constant(0.5), S_0),$$
$$holds(X^1 \doteq linear(1,0), S_0) \wedge holds(X^2 \doteq constant(0.5), S_0).$$

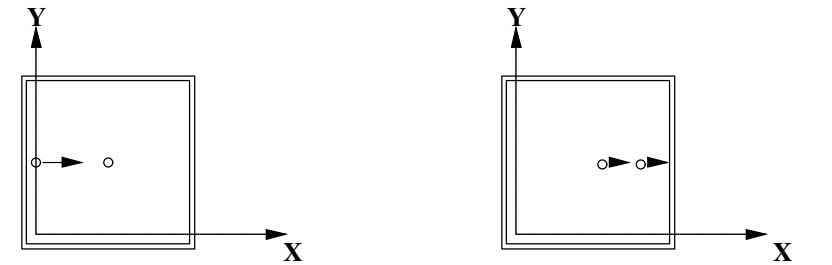
Thus, ball 1 is moving horizontally (i.e., parallel to X-axis) from position $X = 0$, $Y = 0.5$ towards ball 2, which is motionless in the middle of the table (see fig. 4a). Therefore, we would like to infer that if no agents are present in the environment, the balls will collide when ball 1 reaches the initial position of ball 2.

Furthermore, if we introduce the action constant $Collision$, then a correct axiomatization of this problem within the framework of the situation calculus would let us infer the following:

- The collision is possible in S_0 if and only if $start(do(Collision, S_0)) = 0.5$[13]. Thus, the time in which the $Collision$ may occur in S_0 is fixed.

[13] We ignore the dimension of the balls.

- After the collision in the initial situation, i.e. in situation $do(Collision, S_0)$, the motion of both balls is affected (see fig. 4b). Thus, ball 1 will no longer move according to its original trajectory, and ball 2 will start moving.



(*a*) Situation S_0. (*b*) Situation $do(Collision, S_0)$.

Figure 4: The Billiards Table Scenario.

Let us consider a second scenario in the billiards table. Assume now that the initial situation changes so that:

$$holds(Y^1 \doteq constant(0.4), S_0) \wedge holds(Y^2 \doteq constant(0.6), S_0),$$
$$holds(X^1 \doteq linear(1,0), S_0) \wedge holds(X^2 \doteq linear(2,0), S_0).$$

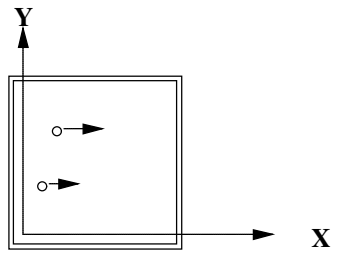


Figure 5: The Second Billiards Table Scenario.

To describe this scenario, we introduce new action constants $ReachEdge_1$ and $ReachEdge_2$.

Initially, in S_0, both balls are moving parallel to the X-axis (see figure 5). The second ball moves faster than the first one. Given that there are no agents that could introduce perturbations, we would like to infer that ball 1 reaches the edge of the table at time $t = 1$, and that ball 2 reaches it at time $t = 0.5$. Here, there are two independent events, $ReachEdge_1$ (ball 1 reaches the edge) and $ReachEdge_2$ (ball 2 reaches the edge). Let us assume that the edge absorbs all the energy of the balls and that they stop at the point of collision. In these circumstances, the evolution of the billiards world is completely determined. There is no uncertainty. Thus, we should conclude that *the only* event that is possible in S_0 is $ReachEdge_2$, and that the only event possible in $do(ReachEdge_2, S_0)$ is $ReachEdge_1$. Thus, $ReachEdge_1$ is impossible in S_0. Therefore, if we eliminate from the tree of situations all those situations that are reached by performing impossible events, we end up with a single line of possible situations. This should not be surprising, since it is a reflection of the fact that the laws of physics, along with a complete specification of the initial conditions, completely determine the evolution of the world (at least for the kind of models we are working with).

Hence, the only way in which we could obtain a branching future is by introducing incompleteness with regards to the actions that may occur. This incompleteness arises when there are *agents* that can choose from a gamut of actions. Each choice is represented by a branch in the situation tree.

Thus, in regards to this section, we understand a *natural process* to be a process whose behavior is completely determined from:

- A description of the initial conditions of the system.
- The physical laws governing the phenomenon.

All events whose occurrences are dictated by these physical laws will be called *natural*. In this section, we present an axiomatization of natural events to characterize the properties we just described. Basically, our view of natural processes is reflected in the structure of the situation calculus trees in the following way: In each situation there is at most one natural action or event possible. This follows from the axiomatization presented below.

4.2.2 Axiomatization

In this section we present an approach to axiomatize problems involving natural phenomena. We make several assumptions that greatly simplify the presentation. In particular, we assume that no natural actions occur concurrently. In general, this assumption is too strong. However, it is not difficult to add explicit concurrency in the situation calculus, thus allowing to eliminate this assumption (Pinto 1994b; Pinto 1994a; Reiter 1996).

In this axiomatization we distinguish natural events from the other events (or actions) introducing the predicate $natural \subseteq \mathcal{A}$. The most important characteristic of natural events is that when the conditions for their occurrence arise, they must occur. In this framework, a complete specification of the predicate *natural* has to be provided by the axiomatizer.

To describe some important properties of natural actions we introduce a new predicate $\Pi_t \subseteq \mathcal{A} \times \mathcal{S} \times \mathcal{T}$ which is domain dependent. $\Pi_t(a,s,t)$ is true whenever the conditions at the time $start(s)$ are such that at time t action a would occur. For example, given the initial situation in figure 4a, the literal $\Pi_t(Collision, S_0, 0.5)$ should be true. That is, in S_0 the conditions are such that if things follow the course predicted at the start of S_0, then a collision is possible at time 0.5. Furthermore, the collision can be guaranteed to occur if no other events occur before time 0.5.

In the same manner as with *natural*, we assume that we have a complete axiomatization for Π_t. Based on the predicates just introduced, we now describe some essential properties of natural events.

First, if a situation s starts at time t_s, then s would be ended by some natural action a at the first time greater than t_s for which $\Pi_t(a,s,t)$ is true. This is written as follows:

$$natural(a) \wedge \Pi_t(a,s,t) \wedge \neg(\exists t')[start(s) < t' < t \wedge \Pi_t(a,s,t')] \supset end(s,a) = t. \tag{95}$$

Also, we use the predicate Π_t to completely characterize the predicate *Poss*. Thus, unlike the case with discrete non-*natural* actions, we may write a general form for a definition of *Poss* for natural events:

$$natural(a) \supset Poss(a,s) \equiv [\Pi_t(a,s,end(s,a)) \wedge \neg(\exists a',t').natural(a') \wedge \Pi_t(a',s,t') \wedge start(s) < t' \leq end(s,a)]. \quad (96)$$

That is, natural action a is possible in state s if, given the conditions at the start of s, the action is possible[14] at the time $end(s,a)$, and no other natural action could have occurred between the start of s and $end(s,a)$. For instance, in the example of figure 5, $\Pi_t(ReachEdge_1, S_0, 1)$ should be true but $Poss(ReachEdge_1, S_0)$ should not. On the other hand, $\Pi_t(ReachEdge_2, S_0, 0.5)$ and $Poss(ReachEdge_2, S_0)$ should be true.

Another essential aspect of any *natural* action a is described by the following axiom:

$$[natural(a) \wedge Poss(a,s) \wedge Poss(a',s) \wedge \neg natural(a')] \supset end(s,a') < end(s,a). \quad (97)$$

Thus, if a is a *natural* action that is possible in situation s, then no other action is possible after the time at which a would occur (i.e., $end(s,a)$). For instance, in the example of figure 5, if an action is possible in S_0, it must bring S_0 to an end at a time before 0.5 (i.e., $(\forall a).Poss(a,S_0) \supset end(S_0,a) \leq 0.5$).

An immediate consequence of the above axioms is:

Observation 4.1

$$natural(a) \wedge natural(a') \wedge Poss(a,s) \wedge Poss(a',s) \supset a = a'.$$

Thus, at most one natural action is possible in a given situation.

Finally, assume that some natural event a is possible in a situation s. Furthermore, assume that nothing occurs before $end(s,a)$. Under these assumptions, the event a must occur. This is formalized as follows:

$$natural(a) \wedge Poss(a,s) \wedge \neg(\exists a')\ [occurs(a',s) \wedge end(s,a') < end(s,a)] \supset occurs(a,s). \quad (98)$$

For example, in the billiards ball scenario of figure 4, if nothing occurs prior to time 0.5, we infer that the *Collision* occurs.

The structure of logical theories for natural processes is similar to the structure for theories with occurrences described in section 3. In fact, let Σ_{nat} denote one of these theories. Σ_{nat} will contain:

- Σ_g. General axioms for the situation calculus. These are: basic axioms of section 2.1. Axioms (15)-(25) and (35) which describe the structure of occurrences. Axiom (84), which establishes the properties of the function $\doteq$. Situation independent axioms (85)-(94). Axioms (95)-(98) that constrain the structure of the situation tree and the assignment of starting times to situations.
- Σ_d, state independent axioms.
- T_{pos}. Same as the set T_{pos} for the theories studied in earlier sections, except that no *Poss* axioms are provided for natural events.

[14] As described by Π_t.

- T_{occ}. Same as the set T_{occ} for the theories studied in section (3).
- T_{Π_t}. In order to describe the conditions under which natural actions are possible, we appeal to the predicate *Poss*, which is true of a situation s and an action a whenever the preconditions for the execution of a in s hold. However, as discussed before, *Poss* may be completely characterized for natural actions after Π_t is given. Therefore, we include in T_{Π_t} a set of sentences that completely specifies Π_t.
- T_{ef}, effect axioms.
- T_{S_0}, specification of the initial conditions.
- T_{sc}, state constraints.

To illustrate the approach, we make use of the ball in the shaft example, originally proposed by Erik Sandewall (1989a). In this example, a ball is moving along a horizontal surface at a positive speed towards a shaft (see figure 6). If the ball reaches the edge of the shaft, it will fall into it with acceleration G. In this scenario, there is only one object: the ball, which is considered point-like. The vertical axis (V) is taken as pointing downwards, whereas the horizontal axis (H) points to the right. The coordinates of the relevant points are indicated in the figure as pairs (H,V). In order to write a theory to describe this scenario, we introduce the action constants $Fall$, $HitWall$ and $HitBottom$, and the parameter constants P_h and P_v.

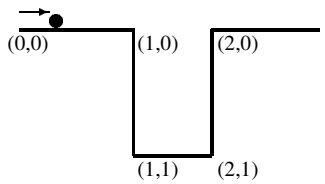


Figure 6: The ball in the shaft.

We axiomatize Sandewall's example as follows:

- Σ_d: There are only three possible events, thus[15]:

$$(\forall a).a = Fall \vee a = HitWall \vee a = HitBottom. \tag{99}$$

Also, all the actions are natural:

$$(\forall a)\ natural(a). \tag{100}$$

Furthermore, the only parameters in this problem are P_h, P_v for the position of the ball with respect to the H and V axes. Also, the parameters are distinct:

$$P_h \neq P_v.$$

[15] Remember that we assume unique names axioms for actions.

- T_{pos}. Since all actions are natural, no domain axioms about *Poss* are given.

- T_{Π_t}. The conditions for *Fall* to occur are that the vertical position of the ball be 0, and that the horizontal motion of the ball be such that an edge of the shaft is reached while moving in the direction of the shaft:

$$\Pi_t(Fall, s, t) \equiv (\exists y_h).holds(P_h \doteq y_h, s) \wedge holds(P_v \doteq constant(0), s) \wedge \\ [q(y_h, t) = 1 \wedge q(d(y_h), t) > 0 \vee q(y_h, t) = 2 \wedge q(d(y_h), t) < 0]. \tag{101}$$

We must emphasize that this axiomatization is very simplified. A better axiomatization would consider the sum of the forces on the ball as the determining factor for the ball's motion. In future research we will attempt to provide a more general theory that incorporates these views.

In (101), we appeal to the fact that if y names the function that describes the position of an object, then $d(y)$ would denote its speed.

The ball hits the wall if its vertical position is inside the shaft and its horizontal position coincides with one of the walls of the shaft and its motion is towards that wall.

$$\Pi_t(HitWall, s, t) \equiv \\ (\exists y_h, y_v).holds(P_h \doteq y_h, s) \wedge holds(P_v \doteq y_v, s) \wedge q(y_v, t) > 0 \wedge \\ [q(y_h, t) = 1 \wedge q(d(y_h), t) < 0 \vee q(y_h, t) = 2 \wedge q(d(y_h), t) > 0]. \tag{102}$$

Finally, a *HitBottom* event occurs when its coordinates coincide with some point along the bottom of the shaft and it is moving towards it.

$$\Pi_t(HitBottom, s, t) \equiv (\exists y_h, y_v).holds(P_h \doteq y_h, s) \wedge holds(P_v \doteq y_v, s) \wedge \\ [q(y_h, t) \geq 1 \wedge q(y_h, t) \leq 2 \wedge q(d(y_v), t) > 0 \wedge q(y_v, t) = 1]. \tag{103}$$

- T_{ef}. When the ball hits a wall, it bounces without losing any energy. Thus, its vertical motion is unaffected, while its horizontal motion changes direction. This effect is modeled by changing the sign of the ball's speed, while maintaining the other parameters:

$$Poss(HitWall, s) \wedge (\exists y_h)\ [holds(P_h \doteq y_h, s) \wedge d(y'_h) = constant(-1) \times d(y_h) \wedge \\ q(y'_h, end(s, HitWall)) = q(y_h, end(s, HitWall))] \supset holds(P_h \doteq y'_h, do(HitWall, s)).$$

The *Fall* does not affect the horizontal motion of the ball. However, it starts a free fall, as reflected by a change in its vertical motion:

$$Poss(Fall, s) \supset holds(P_v \doteq quadratic(0, 0, G), do(Fall, s)).$$

In this effect axiom, we set the arguments for y_v assuming that the initial conditions for the fall are that the height and vertical velocity are zero. A more general effect axiom would state that these initial conditions are determined by the value of the parameters in the previous situation.

When the ball hits the bottom of the shaft (*HitBottom*), all its energy is absorbed. Thus, as a result, motion in both directions stops.

$$Poss(HitBottom, s) \wedge (\exists y_h)\ [holds(P_h \doteq y_h, s) \wedge$$
$$y'_h = constant(q(y_h, end(s, HitBottom)))] \supset holds(P_h \doteq y'_h, do(HitBottom, s)).$$
$$Poss(HitBottom, s) \wedge (\exists y_v)\ [holds(P_v \doteq y_v, s) \wedge$$
$$y'_v = constant(q(y_v, end(s, HitBottom)))] \supset holds(P_v \doteq y'_v, do(HitBottom, s)).$$

These effect axioms can be easily rewritten to conform to the standard form of positive (8) and negative effect axioms (9). In fact, we can combine them to obtain a single positive effect axiom of the form (8) as described later.

- T_{S_0}:

$$holds(P_h \doteq linear(1, 0), S_0) \wedge holds(P_v \doteq constant(0), S_0). \tag{104}$$

- T_{sc} In the example, we do not require domain specific state constraints. However, to illustrate their use, assume that we introduce parameters to denote the speed of the ball with respect to each axis (e.g. $\dot{P}_v$). In this case, the following would be a state constraint:

$$holds(\dot{P}_v \doteq d(y), s) \equiv holds(P_v \doteq y, s).$$

4.2.3 Natural Processes and the Frame and Ramification Problems

So far, we have presented an extension of the situation calculus that allows us to write theories about natural processes. A question that arises is whether or not this extension can be adequately integrated with the solution to the frame and ramification problems studied in section 2. To argue that this is indeed the case, we simply show that axiomatizations of natural processes are a special case of the theories of action with occurrences discussed in 3.

First, the set Σ_d of state independent domain axioms are present in both types of theories. The general axioms (85)-(94), can be considered part of Σ_d.

In Σ_{nat}, the axioms in $T_{pos} \cup T_{\Pi_t}$ along with general axiom (96) play the role of T_{pos} in the theories of action with occurrences. That is, these axioms specify the predicate *Poss*.

The axioms in the sets T_{occ} of occurrence axioms, T_{ef} of effect axioms for natural events, and T_{S_0} of initial conditions have the same form as in previously studied theories of action. The only difference, is that we include general axiom (98) as an occurrence axiom.

The axioms in T_{sc} are as before. Therefore, if the state constraints are of the kind discussed in section 2.4, we will be able to derive successor state axioms from T_{sc} and T_{ef}. Also, axiom (84) is included in the set T_{sc}. Fortunately, the contrapositive form of this axiom corresponds to a binary state constraint:

$$[\neg(\forall t)\ q(y_1, t) = q(y_2, t)] \supset \neg holds(p \doteq y_1, s) \vee \neg holds(p \doteq y_2, s). \tag{105}$$

Clearly, this axiom is a constraint of the form (11). Therefore, it can be used to obtain new effect axioms following the procedure of section 2.4.

Finally, axioms (95) and (97) establish constraints on the temporal relationships between situations in the situation calculus tree, and can be considered part of the basic axioms of the theories of action.

The Ball in the Shaft Here, we show how the solution to the frame and ramification problems is applied to the ball in the shaft example.

First, we rewrite the effect axioms as a single positive effect axiom like (8):

$$Poss(a,s) \wedge \gamma^{+}_{p \doteq y}(a,s) \supset holds(p \doteq y, do(a,s)), \tag{106}$$

where $\gamma^{+}_{p \doteq y}(a,s)$ denotes the formula:

$$\begin{aligned}
& a = HitWall \wedge (\exists y_h)\ [holds(P_h \doteq y_h, s) \wedge d(y) = constant(-1) \times d(y_h) \wedge \\
& \qquad q(y, end(s, HitWall)) = q(y_h, end(s, HitWall))] \vee \\
& a = Fall \wedge p = P_v \wedge y = quadratic(0,0,G) \vee \\
& a = HitBottom \wedge (\exists y_h)\ [p = P_h \wedge holds(P_h \doteq y_h, s) \wedge \\
& \qquad y = constant(q(y_h, end(s, HitBottom)))] \vee \\
& a = HitBottom \wedge (\exists y_v)\ [p = P_v \wedge holds(P_v \doteq y_v, s) \wedge \\
& \qquad y = constant(q(y_v, end(s, HitBottom)))].
\end{aligned}$$

Also, from (105) and (106), we obtain the effect axiom:

$$Poss(a,s) \wedge \gamma^{+}_{p \doteq y}(a,s) \wedge \neg(\forall t)\ q(y,t) = q(y',t) \supset \neg holds(p \doteq y, do(a,s)).$$

With respect to the ball in the shaft example, this means that the ball may not follow two different trajectories. To obtain the set of successor state axioms for each of the fluents, we follow the same procedure outlined in section 2.4. We obtain:

$$\begin{aligned}
Poss(a,s) \supset holds(p \doteq y, do(a,s)) \equiv \gamma^{+}_{p \doteq y}(a,s) \vee holds(p \doteq y, s) \wedge \\
\neg[\gamma^{+}_{p \doteq y}(a,s) \wedge \neg(\forall t)\ q(y,t) = q(y',t)].
\end{aligned} \tag{107}$$

Given that in the ball in the shaft example we assume that there are no external agents, we should be able to determine what happens with the ball at all times. In fact, we can show that:

Proposition 4.1 *In the theory Σ_{nat} for the ball in the shaft example, it follows that:*

$$occurs(Fall, S_0), \tag{108}$$

$$occurs(HitBottom, s) \supset \neg(\exists a)\ occurs(a, do(HitBottom, s)), \tag{109}$$

$$(\forall s).occurs(a,s) \wedge s > S_0 \supset a = HitWall \vee a = HitBottom. \tag{110}$$

4.2.4 Relationship to Sandewall's approach

Conceptually, the work presented in this section is related to Sandewall's (1989a, 1989b). Sandewall proposes to combine logic and differential equations for describing *real world* systems. In Sandewall's words:

> The method applies to *piecewise continuous* physical systems, i.e. systems whose possible histories over time may contain a number of significant time-points called *breakpoints*, and where all parameters of the system are assumed to be continuous in the intervals between breakpoints. (Sandewall 1989a, p.412)

Sandewall's significant time-points correspond to the times at which events occur. However, in our approach we do not require all parameters to be continuous in the intervals between breakpoints. Rather, we require that the parameters be fully described by some function of time.

Another essential difference between Sandewall's approach and ours is that he appeals to non-monotonic reasoning in order to study the relationship between parameters before and after the breakpoints. In fact, the problem of determining what parameters are discontinuous at a breakpoint is a special form of the frame problem. Sandewall's solution to the frame problem is a variant of Yoav Shoham's chronological minimization (Shoham 1988) which is called *Chronological Minimization of Discontinuities*.

Unfortunately, as shown by Rayner (1991), Sandewall's approach fails to sanction the intuitively correct answers in some simple problems. Sandewall's approach is such that if two models coincide in the changes that are produced up to some time point t, then the model that is preferred is the one that has a minimal set of discontinuous parameters at t (regardless of what the models say about the parameters at times after t, hence the term chronological). The problem arises when there are multiple minimal models. Rayner's example is a variation of the ball in the shaft problem studied earlier. In his example, in the bottom half of the shaft there is a radiation field. The radiation field changes the pattern of heat exchange between the ball and its environment. Thus, if the ball enters the radiation field, then the parameter describing the temperature of the ball will suffer a discontinuous change. The problem is that when the ball reaches the field's boundary it may either bounce (changing its motion but not the nature of the heat exchange), or it may cross the field (changing the nature of the heat exchange but not its motion). Thus, the chronological minimization of discontinuities cannot distinguish between the two models. However, the intended conclusion corresponds to the model in which the ball enters the radiation field.

It is important to point out that the deficiency that Rayner found in Sandewall's approach is due to an inadequate solution to the frame problem and not, we believe, to the conceptual underpinnings of his approach. In fact, if we use Reiter's solution to the frame problem, Rayner's problem does not arise. Basically, each breakpoint is associated with some event, whose effects are described using *effect axioms*. Thus, in the previous example, the event of *reaching the boundaries of the radiation field* will have effects on the thermal parameters of the ball but not on its motion. Hence, the successor state axioms will not relate such an event to a change in the ball's motion.

4.3 Discussion

In this section we have presented a novel approach to deal with continuity within the situation calculus. Also, we presented an axiomatization to model natural phenomena utilizing the concept of occurrences introduced in an earlier section. The greatest advantage of the approach presented is that it builds upon a solid foundation based on the situation calculus. In particular, we make use of Reiter's solution to the frame problem, as well as of Lin and Reiter's approach to the ramification problem.

The enriched situation calculus is an attempt to provide a formal foundation for the specification of dynamic systems with continuous variables. Also, we believe that this language can be used to address many of the representational concerns in the area of qualitative reasoning. We have not addressed these issues directly and they are left for future investigation.

There are a number of issues that we have not addressed and which merit some further research. We briefly discuss some of these below.

First, an essential limitation of our approach is that the set of times at which events or actions occur have to be discrete. Otherwise, the induction axiom for situations would be violated. Furthermore, the approach presented is limited by the power of the Oracle and the expressive power of the interaction language. It would be clearly advantageous to study the power of existing systems for dealing with real equations and inequalities. For instance, until we properly characterize the behavior of the Oracle, it is not possible to describe the computational properties of the task of reasoning with continuous systems.

In particular, recall that when the Oracle is asked whether or not an expression of the type (82):

$$\Sigma_{\mathcal{R}} \models (\exists \mathbf{x})\ \Phi(\mathbf{x})\,.$$

is true, the Oracle answers "false" or it responds with an assignment for $\mathbf{x}$ that makes the expression "true". However, in order to predict the future behavior of a dynamic system, we are interested in obtaining an answer to the following question: *given that*

$$(\exists t)\ \phi(t) \tag{111}$$

is true, what is the smallest non-negative real number that satisfies (111)? Such a question arises when we use axiom (96) to determine what is the next time that some natural action is expected to occur. However, the query language that we have defined for the Oracle is not powerful enough to express such a request. Given these expressive limitations, it turns out that there is no bound in the number of queries necessary to determine when the next action may occur. This arises if (111) has an infinite number of solutions over a finite interval of time.

The evaluation of the predicate $Poss(a,s)$ for natural actions using (96) involves finding the earliest time that satisfies a sentence of the form (111) along with evaluating some fluents in the state s. Thus, the evaluation of *Poss* for a given natural action in a situation whose state is known takes a time proportional to the time it takes to find the earliest t for which a formula of the form (111) is satisfied. Furthermore, given the state of a situation, determining the next natural action that is possible takes time that is proportional to the the number of distinct natural actions in

the domain, times the time it takes to evaluate *Poss* for some natural action in a known state. However, caching results from one situation to the next would greatly simplify the computational process.

On the surface, it appears that there are many problems that cannot be addressed using the extension of the situation calculus studied here. For example, consider the problem of modeling the bouncing of a rubber ball. If the behavior of the ball is modeled in such a way that Zeno's paradox arises (Davis 1992), we may face a contradictory theory. In fact, if the ball bounces an infinite number of times in a finite time interval, and if we model each bounce of the ball as a natural event, then there will be a line of situations that contains an infinite number of situations lapsing a finite time. Problems arise if we state that some situation exists after the ball stops bouncing. In fact, the infinite number of bounces (each one represented as an occurrence), coupled with an existing situation after the ball stops, is reflected in a structure of situations that is not consistent with the induction axiom (1).

A possible way around the problem mentioned above, is to model the bouncing of the ball with a single function of time and without treating the *bounce* as an event. Thus, we would have a real function of time completely describing the behavior of the ball with its many bounces. Furthermore, we would state that there is a single situation in which the ball follows the trajectory dictated by this function. Clearly, this solution is not general. In fact, a function to describe the complete behavior of the system may not be available. Furthermore, one of the reasons to model natural phenomena as sequences of natural events is to avoid having to specify real functions to model the entire behavior of the dynamic system.

Unfortunately, even if we model the behavior of the system as described above, we still need to ensure that the axioms for natural processes do not introduce inconsistencies with the induction axiom for situations. In many cases this will require extra-logical proofs that infinite series of natural events in finite time do not arise.

Another issue that we did not address has to do with inherent uncertainty. In fact, we stated that we only deal with processes whose mathematical models are completely determinate. Thus, we assumed that, given a complete specification of the initial situation, there is only one way in which the world may evolve. As we saw before, this is reflected in the situation tree by the fact that, in the absence of non-natural actions, all possible events line up in a single branch. This reflects a view of the world in which there is no uncertainty. However, if we want to model situations in which there is uncertainty with regards to which natural events may arise, we can simply allow for more than one natural action to be possible in some situations.

There are several problems regarding continuous properties that we have not addressed. For instance, we may want to provide a taxonomy of properties of objects that must obey certain continuity properties. For example, we may want to state that the derivative of the function that describes the position of an object in space must always exist, except in break points (e.g., points in which some action abruptly modifies the behavior of the parameter). Problems of these nature are analyzed by Sandewall (1989a) in the framework of another temporal logic. A problem that arises is that we might need to introduce inter-state constraints, which we have not considered. For example, we might want to state that for any action a and situation s, the position of an object at the end of s must be the same as the position at the start of $do(a,s)$. Thus, adding a more general description of physical phenomena may require us to re-study the solution to the frame and ramification problems in the presence of inter-state constraints.

5 Conclusion

5.1 Summary

The main objective of this work is to design a logical language to represent knowledge about dynamic domains. We chose to use the situation calculus as the basic logical language. This choice is partly motivated by the expressive power of the language. However, a stronger motivation is the availability of solutions to the frame problem within the framework. Also, we benefited from the work of Lin and Reiter to approach the ramification problem within the situation calculus. As emphasized in this paper, an important contribution of our work has been to extend the situation calculus language to deal with a wide variety of representational aspects of dynamic worlds. The contribution of our work to the problem of representing knowledge about dynamic worlds can be summarized as:

- Extend Lin and Reiter's work on the ramification problem by providing mechanisms to deal with theories of action that include binary state constraints.
- Extend Pinto and Reiter's approach to represent determinate knowledge about the future within the situation calculus. This extension allows us to infer that natural processes go on unless some intervening actions are performed. These actions are non-monotonically assumed not to arise.
- Present an enriched ontology of the situation calculus that allows the representation of knowledge about continuous properties of the world.

This work is part of a larger enterprise that seeks to create a logical framework to represent knowledge about agent populated dynamic domains within the situation calculus. This enterprise started with the pioneering work of McCarthy (1968) and McCarthy and Hayes (1969). The situation calculus has been one of the standard languages to study well known problems in knowledge representation. For instance, a great deal of research on the frame problem has been done using the situation calculus as the theoretical framework (e.g., (Lifschitz 1987; Hanks and McDermott 1987; Lifschitz and Rabinov 1989; Baker 1991) and many others). However, it has been during this decade that serious efforts have been made to integrate, within the situation calculus, the knowledge necessary to model agent populated dynamic worlds. Furthermore, the advances in this enterprise are making possible the use of extended versions of the situation calculus in a variety of domains, ranging from databases (Reiter 1992a; Bertossi and Ferreti 1994), Software Engineering (Borgida, Mylopoulos, and Reiter 1993), Business Process Modeling (Gruninger and Pinto 1995; Yu, Mylopoulos, and Lespérance 1996) to robotics (Levesque, Reiter, Lespérance, Lin, and Scherl 1996; Lespérance, Levesque, Lin, Marcu, and Reiter 1995).

5.2 Further Research

This work can be extended in many ways. In particular:

- Multiple Agents: We have presented an approach that allows one to express occurrences of external events. Along with current work to allow for concurrency, we would like to explicitly introduce the notion of *agent*. Thus, we would probably introduce a new sort for agents, with actions indexed by the agent that performs them.

- Specification of Dynamic Processes. The ability to deal with continuous change allows us to represent physical processes that are commonly encountered in industry. Thus, with the extended situation calculus described in this work, we provide a tool to specify these processes utilizing what is commonly called a *hybrid* representational mechanism. Thus, all processes that can be modeled using standard mathematical models can be integrated with the models of actions and effects of the discrete situation calculus. The extensions to the situation calculus presented in sections 3 and 4 provide the foundations for such applications.

- Planning with External Events. The planning approaches built around the situation calculus normally consider single-agent worlds, in which all actions are the responsibility of the single agent. Thus, we can investigate extensions of this research to problems in which there are external events which may be the result of other agents present in the world, the effects of nature, or simply delayed effects resulting from actions performed by the same agent.

- Reasoning with the Representation. We have developed a logic programming approach to do temporal projection for simple occurrence theories (Pinto and Reiter 1993). We would like to extend this work to reason about continuous change as well. Furthermore, we would like to investigate a theorem proving approach along the lines of Bertossi (Bertossi and Ferreti 1994; Bertossi, Pinto, Saez, Kapur, and Subramaniam 1996).

- Assessment of Applicability. We have proposed a solution to the frame and ramification problems for some families of theories. It is essential that any such proposal be assessed for correctness. Currently, there are two independent formal frameworks to perform these assessments: Sandewall's (1994, 1995) and Gelfond and Lifschitz's (1993, 1993). Unfortunately, neither framework considers continuous change. Thus, before we can attempt to perform an assessment, we need to extend one of the aforementioned frameworks. This is part of our current work.

- Hypothetical Reasoning. Our work builds upon the work of Pinto and Reiter (1993), in which occurrences are formalized using the notion of an *actual line* of situations. A shortcoming of this approach is that it does not adequately supports hypothetical reasoning about occurrences. Therefore, this approach inherits this limitation. Currently, we are developing an alternative framework in which occurrences are not formalized in terms of actual lines. Instead, occurrences are a primitive notion. Preliminary results are reported in (Pinto 1996b; Pinto 1996a). Our modeling of continuous change does not directly depend on the formalization of occurrences as actions in the actual line. Therefore, our approach is equally applicable to this new formulation of occurrences.

5.3 Related Work

One of the earliest to formulate a temporal logic with branching continuous time was Drew McDermott (1982). Also, DeJong (1994) proposes an approach based on learning to do planning in continuously changing worlds. He focuses on the representation of continuous actions that are controlled by the planning agent. DeJong's work uses a domain representation language similar to Forbus's qualitative process theory (Forbus 1984). On the other hand, Davis (Davis 1994), presents a language to model the behavior of autonomous agents in terms of continuous control and choice, in which time is modeled as a branching structure. Unlike the other approaches, Davis shows how discrete theories can be embedded in the continuous branching time structure.

There is a great deal of conceptual overlap between our model of time and the models proposed in the aforementioned literature. Perhaps the most essential difference is that we build upon a solution to the frame and ramification problems, which can be extended to incorporate concurrency (Pinto 1994a). Indeed, an important contribution of this article is that we extend Reiter's solution to the frame problem to domains in which there is continuous change. An important feature of our approach is that continuous change is integrated with discrete actions allowing us to use existing planning techniques. Planning in our framework is a subject for future research.

Another important work in this area is the one of Miller (1996), where he uses a conceptual approach similar to ours, but based upon Baker's circumscriptive solution to the frame problem (Baker 1991). Miller's work extends the language of the situation calculus and incorporates linguistic constructs to add differential equations. Furthermore, Miller and Shanahan (1996) apply Miller's strategy within the framework of the circumscriptive event calculus (Shanahan 1995).

In an approach related to ours, Reiter (Reiter 1996) takes a situation calculus with concurrent actions as a starting point and models continuous motion using Newtonian equations of motion. Reiter also shows how these equations can be integrated within the discrete situation calculus. However, he does not provide a mechanism to prefer one development over another using some preferential ordering. Finally, Kelley (1996) has shown how to model a steam boiler using the language of the situation calculus extended to accommodate natural actions.

Acknowledgements

Many people have influenced the research that I report in this paper. I would like to thank Leopoldo Bertossi, Alvaro Campos, Hector Levesque, Fangzhen Lin, Michael Gruninger, Todd Kelley, and Sheila McIlraith for feedback on this work. Special thanks to Raymond Reiter for many hours of inspiring discussions. Also, thanks to the anonymous reviewers who carefully read a long manuscript and provided important feedback.

This work has been partly funded by Fondecyt under grant # 1950885, and by the School of Engineering of the Catholic University of Chile.

A Structural Properties of Situations

Here, we prove some interesting properties of the structure of situations in the discrete situation calculus. These properties are used in the next section in order to prove the various results presented in the text.

Proposition A.1 *Some consequences of axioms (1)-(6) are:*

- *All situations are reachable from S_0:*

$$(\forall s)\,.\,S_0 \leq s. \tag{112}$$

Let $\varphi(s) = [S_0 \leq s]$. Clearly $\varphi(S_0)$ is identically true. Furthermore, from (3) it follows that $S_0 \leq s \supset S_0 \leq do(a,s)$. Therefore, the antecedent of (1) is true for our choice of φ. Hence, (112) follows.

-

$$(\forall a, s)\,.\,\neg\, s < s. \tag{113}$$

By contradiction using (4) with $s_1 = s_2 = s$.

-

$$(\forall s)\,.\,s \neq S_0 \supset (\exists a, s')\,.\,s = do(a,s'). \tag{114}$$

Let $\varphi(s) = [s \neq S_0 \supset (\exists a, s')\,.s = do(a,s')]$. $\varphi(S_0)$ holds trivially. Since $\varphi(do(a,s))$ is identically true,

$$(\forall a, s)\,.\varphi(s) \supset \varphi(do(a,s))$$

follows. Therefore $(\forall s)\,.\varphi(s)$ holds, thus (114) holds as well.

-

$$(\forall s_1, s_2, s_3)\,.\,s_1 < s_2 \wedge s_2 < s_3 \supset s_1 < s_3. \tag{115}$$

Again, we use our induction axiom for situations. Let

$$\varphi(s) = (\forall s_1, s_2)\,.\,s_1 < s_2 \wedge s_2 < s \supset s_1 < s.$$

$\varphi(S_0)$ holds trivially ($s_2 < S_0$ is identically false). We have to show that $\varphi(s) \supset \varphi(do(a,s))$. In fact, From (3), it follows that $s_1 < s \supset s_1 < do(a,s)$. Therefore:

$$\varphi(s) \supset (\forall s_1, s_2)\ [s_1 < s_2 \wedge s_2 < s \supset s_1 < do(a,s)]. \tag{116}$$

Now, from (3) and (113) it follows that:

$$s_2 < do(a,s) \wedge s_2 \neq s \equiv s_2 < s.$$

Which allows us to rewrite (116) as:

$$\varphi(s) \supset (\forall s_1, s_2)\ [s_1 < s_2 \wedge s_2 < do(a,s) \wedge s_2 \neq s \supset s_1 < do(a,s)]. \tag{117}$$

From (3):

$$(\forall s_1, s_2).s_1 < s_2 \supset s_1 < do(a,s_2).$$

Thus:

$$(\forall s_1, s_2).s_1 < s_2 \wedge s_2 < do(a,s) \wedge s_2 = s \supset s_1 < do(a,s).$$

Therefore, we can conclude that:

$$\varphi(s) \supset (\forall s_1, s_2)\ [s_1 < s_2 \wedge s_2 < do(a,s) \supset s_1 < do(a,s)]. \tag{118}$$

In other words $\varphi(s) \supset \varphi(do(a,s))$

- $$(\forall a, s).s < do(a,s). \tag{119}$$

  Trivially follows from (3).

- $$do(a_1,s_1) < do(a_2,s_2) \supset s_1 < s_2. \tag{120}$$

  Assume:

  $$do(a_1,s_1) < do(a_2,s_2).$$

  From (3):

  $$do(a_1,s_1) \leq s_2.$$

  From (119) $s_1 < do(a,s_1)$*, which along with (115) leads to:*

  $$s_1 < s_2.$$

  So that (120) holds.

- $$(\forall a, s_1, s_2).do(a,s_1) = do(a,s_2) \supset s_1 = s_2. \tag{121}$$

This can be shown by contradiction. Let's extend the language with new constants A, S_1, and S_2 such that:

$$do(A,S_1) = do(A,S_2) \wedge S_1 \neq S_2.$$

Obviously, $S_1 < do(A,S_1)$, since $do(A,S_1) = do(A,S_2)$, we get $S_1 < do(A,S_2)$. From (3), we obtain $S_1 \leq S_2$; since $S_1 \neq S_2$, we get $S_1 < S_2$. By symmetry, we also obtain $S_2 < S_1$, which contradicts (4). Therefore, (121) follows.

B Proofs

Proposition 2.1. This proposition establishes that from axioms (1-4) and (7) it follows that:

$$s_1 < s_2 \wedge holds(f,s_1) \wedge \neg holds(f,s_2) \supset \\ (\exists s,a)\,(s_1 < do(a,s) \leq s_2) \wedge ab(f,a,s). \tag{122}$$

The proof is by contradiction. We extend the language with new constants S_1, S_2, and F, such that:

$$S_1 < S_2 \wedge holds(F,S_1), \tag{123}$$

$$\neg holds(F,S_2), \tag{124}$$

and:

$$(\forall s,a)\,(S_1 < do(a,s) \leq S_2) \supset \neg ab(F,a,s). \tag{125}$$

We will show that (123) and (125) contradict (124). The contradiction arises by applying the law of inertia (7) and the induction axiom (1) with:

$$\varphi(s) = (S_1 \leq s \leq S_2) \supset holds(F,s).$$

$\varphi(S_0)$ is immediately true, since $(S_1 \leq S_0 \leq S_2) \supset S_0 = S_1$ and we assumed that $holds(F,S_1)$. We now prove that $\varphi(s) \supset \varphi(do(a,s))$. We do this by cases:

- We prove that:

  $$(\forall a,s)\,\neg(S_1 \leq s \leq S_2) \wedge \varphi(s) \supset \varphi(do(a,s)).$$

  Assume that S is an arbitrary situation such that $\neg(S_1 \leq S \leq S_2)$. If $S_1 \leq do(a,S) \leq S_2$ is false, then trivially $\varphi(do(a,S))$ is true. Now, if $S_1 \leq do(a,S) \leq S_2$ is true, then $do(a,S) = S_1$, otherwise $(S_1 \leq S \leq S_2)$ would hold. Since $\neg ab(F,a,S_1)$ follows from (125), we infer, using the law of inertia, that $holds(F,do(a,S))$. Hence, $\varphi(do(a,s))$ also holds in this case.

- Now we prove that:

  $$(\forall a,s)\,(S_1 \leq s \leq S_2) \wedge \varphi(s) \supset \varphi(do(a,s)).$$

Assume that S is an arbitrary situation such that $(S_1 \leq S \leq S_2) \wedge \varphi(S)$ is true. Obviously $holds(F,S)$ follows; this along with (125) ensures that $\varphi(do(a,S))$ must also be true. (Notice that if $(S_1 \leq S \leq S_2) \supset do(a,S) \neq S_1$).

Therefore, it follows that $(\forall a,s)\varphi(s) \supset \varphi(do(a,s))$. So, applying (1) we infer that (124) is contradicted. Hence, the proposition must be true.

Proposition 2.2. As mentioned in the text of the proposition, we show this by utilizing induction. Thus, we assume that $T_{sc_2}(S_0)$ is true and use the induction axiom (1) with:

$$\varphi(s) = S_0 \preceq s \supset T_{sc_2}(s).$$

Obviously, $\varphi(S_0)$ is true, given that we are assuming that $T_{sc_2}(S_0)$ is true. Hence, we need to show that:

$$\varphi(s) \supset \varphi(do(a,s)).$$

Which can be rewritten as:

$$(S_0 \preceq s \supset T_{sc_2}(s)) \supset (S_0 \preceq do(a,s) \supset T_{sc_2}(do(a,s))). \tag{126}$$

We know that $\neg S_0 \preceq s \supset \neg S_0 \preceq do(a,s)$. If $S_0 \preceq s$ is false, then (126) is trivially true. If $S_0 \preceq s$ is true, then (126) translates to:

$$S_0 \preceq s \wedge T_{sc_2}(s) \wedge S_0 \preceq do(a,s) \supset T_{sc_2}(do(a,s)).$$

This is proven by contradiction. Thus, assume that:

$$S_0 \preceq s \wedge T_{sc_2}(s) \wedge S_0 \preceq do(a,s) \wedge \neg T_{sc_2}(do(a,s)).$$

If $\neg T_{sc_2}(do(a,s))$ holds, then there must be a state constraint in T_{sc_2} that is not satisfied. Thus, let

$$\alpha(s) = \neg\rho(\mathbf{x}_r) \supset holds(f_1(\mathbf{x}_1),s) \vee holds(f_2(\mathbf{x}_2),s),$$

be an arbitrary such a constraint[16]. Since $\rho(\mathbf{x}_r)$ is state independent, it follows that:

$$\alpha(s) \wedge \neg\alpha(do(a,s)) \supset \neg\rho(\mathbf{x}_r).$$

Thus, it also follows that:

$$holds(f_1(\mathbf{x}_1),s) \vee holds(f_2(\mathbf{x}_2),s), \tag{127}$$

$$\neg(holds(f_1(\mathbf{x}_1),do(a,s)) \vee holds(f_2(\mathbf{x}_2),do(a,s))). \tag{128}$$

Notice that the only way for this situation to arise is that the set T'_{ef} contain effect axioms like:

$$Poss(a,s) \wedge \gamma^-_{f_1}(a,s) \supset \neg holds(f_1(\mathbf{x}_1),do(a,s)), \tag{129}$$

$$Poss(a,s) \wedge \gamma^-_{f_2}(a,s) \supset \neg holds(f_2(\mathbf{x}_2),do(a,s)). \tag{130}$$

Each of these effect axioms either belong to the original set of effect axioms T_{ef} or are a consequence of $T_{sc_2} \cup T_{ef}$. In either case it follows that the original theory Σ_{sc_2} is inconsistent. Thus, the result follows.

[16]Notice that the argument that follows works whether or not the *holds* literals are positive (as in this case) or negative.

Theorem 2.2. This theorem is an immediate consequence of theorem 2.1 and proposition 2.2.

Proposition 3.1. This proposition is proved by appealing to the induction axiom (1). The first part of the proposition follows by taking:

$$\varphi(s) = (\forall s').s' \leq s \supset [actual(s) \supset actual(s')],$$

and the second one, by taking

$$\varphi(s) = (\forall s').actual(s) \wedge actual(s') \supset s < s' \vee s' < s \vee s = s'.$$

Straightforward manipulation leads to:

$$(\forall a, s).\varphi(s) \supset \varphi(do(a, s)).$$

Which directly implies the results.

Proposition 3.2. It follows by using the induction axiom (1) with:

$$\varphi(s) = (\forall s').s' < s \supset start(s') < start(s).$$

Proposition 3.4. Let O_s and O_t denote the left and right side of the equivalence. First we show that $O_s \supset O_t$:

From proposition 3.2 and using a simple induction on the length of $\mathcal{O}_<$ it follows that:

$$\mathcal{O}_<(s_1, \ldots, s_n) \supset \mathcal{O}_<(start(s_1), \ldots, start(s_n)). \tag{131}$$

In which the ordering $<$ in the left side refers to the order relation between situations, and the $<$ on the right side refers to the order relation between the reals.

From (120), (21) and propositions 3.1 and 3.2 it follows that:

$$s_1 < s_2 \wedge occurs(a_1, s_1) \wedge occurs(a_2, s_2) \supset do(a_1, s_1) < do(a_2, s_2). \tag{132}$$

Thus, from (132) and a simple induction on the length of $\mathcal{O}_<$ we obtain:

$$O_s \supset \mathcal{O}_<(do(A_1, S_1), \ldots, do(A_n, S_n)). \tag{133}$$

From (131), (133) and (22) we obtain:

$$\begin{aligned}(\exists s_1, \ldots, s_n).[&occurs(A_1, s_1) \wedge \ldots \wedge occurs(A_n, s_n) \wedge \mathcal{O}_<(s_1, \ldots, s_n) \\ &\supset \\ &occurs_{\mathcal{T}}(A_1, start(do(A_1, s_1))) \wedge \ldots \wedge occurs_{\mathcal{T}}(A_n, start(do(A_1, s_n))) \wedge \\ &\mathcal{O}_<(start(do(A_1, s_1)), \ldots, start(do(A_1, s_n)))].\end{aligned}$$

Thus, the we have proven that $O_s \supset O_t$.

Now we show that $O_t \supset O_s$:

From (22) we obtain:

$$(\exists t_1,\ldots,t_n).occurs_{\mathcal{T}}(A_1,t_1)\wedge\ldots\wedge occurs_{\mathcal{T}}(A_n,t_n)\wedge\mathcal{O}_<(t_1,\ldots,t_n) \tag{134}$$
$$\supset$$
$$(\exists s_1,\ldots,s_n).occurs(A_1,s_1)\wedge\ldots\wedge occurs(A_n,s_n)\wedge$$
$$\mathcal{O}_<(start(do(A_1,s_1)),\ldots,start(do(A_1,s_n))).$$

From (120) and propositions 3.1 and 3.2 it follows that:

$$start(do(a_1,s_1))<start(do(a_2,s_2))\wedge actual(do(a_1,s_1))\wedge actual(do(a_2,s_2))\supset s_1<s_2. \tag{135}$$

From (134) and (135) it immediately follows that $O_t \supset O_s$.

□

Proposition 4.1. (108) states that the first occurrence is a *Fall* event. The proof is as follows: From (101) and (104) we obtain:

$$\Pi_t(Fall,S_0,t)\equiv[q(linear(1,0),t)=1\wedge q(d(linear(1,0)),t)>0\vee \tag{136}$$
$$q(linear(1,0),t)=2\wedge q(d(linear(1,0)),t)<0].$$

From (90) we know that $linear(1,0)=t$, and from (93) $d(linear(1,0))=constant(1)$. Thus, it follows that we can rewrite (136) as:

$$\Pi_t(Fall,S_0,t)\equiv[t=1\wedge q(constant(1),t)>0\vee t=2\wedge q(constant(1),t)<0].$$

Which can be rewritten as:

$$\Pi_t(Fall,S_0,t)\equiv[t=1\wedge 1>0\vee t=2\wedge 1<0].$$

At this point in the proof, we need to ask our Oracle whether:

$$(\exists t).t=1\wedge 1>0\vee t=2\wedge 1<0.$$

To which the Oracle will reply *true* and give an assignment $t=1$. Then we may ask whether:

$$(\exists t).t\neq 1\wedge[t=1\wedge 1>0\vee t=2\wedge 1<0].$$

To which the Oracle would reply *false*. Thus, we conclude that:

$$\Pi_t(Fall,S_0,t)\equiv t=1.$$

Also, from (95), we conclude that:

$$end(S_0,Fall)=1.$$

Now, from (102):

$$\Pi_t(HitWall, s, t) \equiv q(constant(0), t) > 0 \wedge [t = 1 \wedge 1 < 0 \vee t = 2 \wedge 1 > 0].$$

By consulting the Oracle we know that $0 > 0$ is false, therefore $\Pi_t(HitWall, S_0, t)$ is also false. A similar proof leads us to conclude that $\Pi_t(HitBottom, S_0, t)$ is false. Therefore, we arrive at the conclusion that:

$$\Pi_t(a, S_0, t) \equiv a = Fall \wedge t = 1.$$

From (96), (99) and (100), we infer that $Poss(a, S_0) \equiv a = Fall$. Furthermore, from (98), it follows that $occurs(Fall, S_0)$.

To prove (109), it is not difficult to show (from T_{Π_t}) that after a $HitBottom$ occurs in s:

$$\Pi_t(a, do(HitBottom, s), t) \equiv false.$$

Finally, to prove (110) we use the induction axiom (1) with:

$$\varphi(s) = occurs(a, s) \wedge s > S_0 \supset a = HitWall \wedge$$
$$[holds(y_v, do(HitWall, s)) \equiv holds(y_v, do(Fall, S_0))] \vee a = HitBottom.$$

$\varphi(S_0)$ is vacuously true. Assuming that for an arbitrary S, $\varphi(S)$ is true, we need to show that $\varphi(do(a', S))$ is also true, for an arbitrary a'. This is trivial to show by using the effect axioms and appealing to the Oracle.

□

References

Allen, J. and G. Ferguson (1993, January). Action in Interval Temporal Logic. In *The Second Symposium on Logical Formalizations of Commonsense Reasoning*, pp. 12–22.

Allen, J. F. (1984). Towards a General Theory of Action and Time. *Artificial Intelligence 23*, 123–154.

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.

Baker, A. (1991). Nonmonotonic Reasoning in the Framework of the Situation Calculus. *Artificial Intelligence 49*, 5–23.

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 Brazilean AI Society (SBIA 95)*, Number 991 in LNAI, pp. 59–68. Springer.

Bertossi, L. and J. Ferreti (1994). SCDBR: A Reasoner for Specifications in the Situation Calculus of Database Updates. In *Proceedings of the First Intenational Conference on Temporal Reasoning*.

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.), *Lecture Notes in Artificial Intelligence: Proceedings of the 9th International Symposium on Methodologies for Intelligent Systems*, pp. 212–222. Springer-Verlag.

Borgida, A., J. Mylopoulos, and R. Reiter (1993, May). ... And Nothing Else Changes: The Frame Problem in Procedure Specifications. In *Proceedings of the Fifteenth International Conference on Software Engineering*.

Davis, E. (1992). Infinite loops in finite time: Some observations. In C. R. B. Nebel and W. Swartout (Eds.), *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning (KR'92)*, pp. 47–58. Morgan Kaufmann Publishers, Inc.

Davis, E. (1994, July). Branching Continuous Time and the Semantics of Continuous Action. Technical Report TR1994-666, New York University, 251 Mercer St. New York.

DeJong, G. (1994). Learning to Plan in Continuous Domains. *Artificial Intelligence 65*(1), 71–141.

Forbus, K. (1984). Qualitative Process Theory. *Artificial Intelligence 24*.

Galton, A. (1990). A Critical Examination of Allen's Theory of Action and Time. *Artificial Intelligence 42*(2-3), 159–188.

Gelfond, M. and V. Lifschitz (1993). Representing Action and Change by Logic Programs. *The Journal of Logic Programming 17*, 301–322.

Georgeff, M. P. (1987, April). Many Agents are Better than One. In F. M. Brown (Ed.), *The Frame Problem in Artificial Intelligence*, Los Altos, California, pp. 58–76. Morgan Kaufmann Publishers, Inc.

Gruninger, M. and J. Pinto (1995, March). A Theory of Complex Actions for Enterprise Modelling. In C. Boutilier and M. Goldszmidt (Eds.), *Symposium Extending Theories of Action: Formal Theory and Practical Applications*, Number SS-95-07 in 1995 AAAI Symposium, Stanford, California, pp. 94–99. Stanford University: AAAI Press. ISBN 0-929280-90-3.

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

Hanks, S. and D. McDermott (1987). Nonmonotonic Logic and Temporal Projection. *Artificial Intelligence 33*(3), 379–412.

Kautz, H. (1986). The Logic of Persistence. In *Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI-86)*, pp. 401–405.

Kelley, T. G. (1996). Modeling Complex Systems in the Situation Calculus: A Case Study Using the Dagstuhl Steam Boiler Problem. In *Proceedings KR'96*. Morgan Kaufmann. To appear.

Kowalski, R. and M. Sergot (1986). A Logic-based Calculus of Events. *New Generation Computing 4*, 67–95.

Lespérance, Y., H. Levesque, F. Lin, D. Marcu, and R. Reiter (1995, March). Foundations of a Logical Approach to Agent Programming. In *Actes des Troisièmes Journées Francophones sur l'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents*, Chambery-St-Badolph, France.

Levesque, H. J., R. Reiter, Y. Lespérance, F. Lin, and R. B. Scherl (1996). GOLOG: A Logic Programming Language for Dynamic Domains. *The Journal of Logic Programming*. To appear.

Lifschitz, V. (1985). Computing Circumscription. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-85)*, pp. 121–127.

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

Lifschitz, V. (1993, January). A Language for Describing Actions. In *The Second Symposium on Logical Formalizations of Commonsense Reasoning*, pp. 103–113.

Lifschitz, V. and A. Rabinov (1989). Miracles in Formal Theories of Action. *Artificial Intelligence 38*, 225–237.

Lin, F. and R. Reiter (1993, January). State Constraints Revisited. In *The Second Symposium on Logical Formalizations of Commonsense Reasoning*, pp. 114–121.

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

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

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

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

McCarthy, J. (1980). Circumscription a form of Non-Monotonic Reasoning. *Artificial Intelligence 13*, 27–39.

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

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

McDermott, D. (1982). A Temporal Logic for Reasoning About Processes and Plans. *Cognitive Science 6*, 101–155.

Miller, R. (1996). A Case Study in Reasoning about Actions and Continuous Change. In *Proceedings ECAI*. John Wiley & Sons.

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

Miller, R. and M. Shanahan (1996). Reasoning about Dicontinuities in the Event Calculus. In *Proceedings KR'96*. Morgan Kaufmann. To appear.

Pinto, J. (1994a). Concurrent Events: Synergy and Cancellation of Effects. In *European Conference on Artificial Intelligence, Workshop on Logic and Change*, pp. 105–110.

Pinto, J. (1994b, 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/~cogrob/jpThesis.ps.Z.

Pinto, J. (1996a). Incorporating Observations in Theories of Action. In G. Antoniou and M. Truszczynski (Eds.), *PRICAI'96 Workshop on Reasoning with Incomplete and Changing Information. Research Report CIT-96-06, School of Computing and Information Technology, Griffith University*, pp. 79–88.

Pinto, J. (1996b). Occurrences in the Hypothetical Worlds of the Situation Calculus: Extended Abstract. In U. C. Sigmund and M. Thielscher (Eds.), *Reasoning about Actions and Planning in Complex Environments, FAPR'96 Workshop. Technical Report AIDA-96-11, Technische Hochschule Darmstadt, Fachbereich Informatik*, pp. VII–1 – VII–8.

Pinto, J. (1997). State constraints and epistemological completeness. In progress.

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. and R. Reiter (1995, September). Reasoning about Time in the Situation Calculus. *Annals of Mathematics and Artificial Intelligence 14*(2-4), 251–268.

Rayner, M. (1991). On the applicability of nonmonotonic logic to formal reasoning in continuous time. *Artificial Intelligence 49*, 345–360.

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. (1992a, June). Formalizing database evolution in the situation calculus. In *Proceedings of the International Conference on Fifth Generation Computer Systems*, pp. 600–609.

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

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

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.

Sandewall, E. (1989a, May). Combining Logic and Differential Equations for Describing Real-World Systems. In H. L. R. Brachman and R. Reiter (Eds.), *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, Toronto, Ontario, Canada, pp. 412–420. Morgan Kaufmann Publishers, Inc.

Sandewall, E. (1989b). Filter Preferential Entailment for the Logic of Action in almost Continuous Worlds. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-89)*, Detroit, MI, pp. 894–899.

Sandewall, E. (1991, August). Features and Fluents. Technical Report LiTH-IDA-R-91-29, Linköping University.

Sandewall, E. (1994). The range of applicability of some nonmonotonic logics for strict inertia. *Journal of Logic and Computation 4*(5).

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. (1990). Representing Continuous Change in The Event Calculus. In *Proceedings ECAI*.

Shanahan, M. (1993). Explanation in the Situation Calculus. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-93)*, pp. 160–165.

Shanahan, M. (1995, September). A Circumscriptive Calculus of Events. *Artificial Intelligence 77*(2), 249–284.

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

Stein, L. and L. Morgenstern (1994). Motivated Action Theory: A Formal Theory of Causal Reasoning. *Artificial Intelligence 71*, 1–42.

Yu, E. K., J. Mylopoulos, and Y. Lespérance (1996, August). Modelling the Organization: New Concepts and Tools for Re-Engineering,. *IEEE Expert,* special issue on Artificial Intelligence Applications in Enterprise Modeling.