Natural Actions, Concurrency and Continuous Time in the Situation Calculus

Ray Reiter
Department of Computer Science
University of Toronto
Toronto, Canada M5S 1A4
and
The Canadian Institute for Advanced Research
email: reiter@cs.toronto.edu http://www.cs.toronto.edu/~cogrobo/

Abstract

Our focus in this paper is on natural exogenous actions (Pinto [23]), namely those which occur in response to known laws of physics, like a ball bouncing at times determined by Newtonian equations of motion. The property of such actions that we wish to capture is that they must occur at their predicted times, provided no earlier actions (natural or agent initiated) prevent them from occurring. Because several such actions may occur simultaneously, we need a theory of concurrency. Because such actions may be modeled by equations of motion, we need to represent continuous time. This paper shows how to gracefully accommodate all these features within the situation calculus, without sacrificing the simple solution to the frame problem of Reiter [25]. One nice consequence of this approach is a situation calculus specification of deductive planning, with continuous time and true concurrency, and where the agent can incorporate external natural event occurrences into her plans.

1 Introduction

For the past several years, the Cognitive Robotics Group at the University of Toronto has been exploring the feasibility of the situation calculus (McCarthy [19]) as a theoretical and computational foundation for modeling autonomous agents dwelling in dynamic environments. It is a challenging research problem to capture, in a single formal and computational framework, the full range of characteristics associated with such settings: the frame, ramification and qualification problems, exogenous and natural events, chance events and the unpredictability of action effects, complex actions and procedures and the ability of an agent to perform such actions, time, concurrency, hypothetical and counterfactual reasoning about action occurrences and time, perceptual actions and their effects on an agent's mental state, the complex relationships among reasoning, perception and action, planning, belief revision in the presence of conflicting observations, etc. The principal objective of this project is to provide just such a general theory of actions and time, and, as noted above, our formal foundation for this has been the situation calculus.

While we remain far from achieving these long-range objectives, we have had some modest success in this undertaking. Starting with a solution to the frame problem for deterministic, simple actions (Reiter [25]), we have defined and implemented a novel situation calculus-based logic programming language for defining complex agent behaviors (Levesque et al. [14]), and experimented with it in a robotics application (Lespérance et al. [12]), and for software agents (Marcu et al. [18]). Scherl and Levesque [28] have given a situation calculus account of sensing (knowledge-producing) actions, and Bacchus, Halpern and Levesque have extended this to include noisy sensors [2]. Lin [15] has extended Reiter's treatment for deterministic primitive actions to nondeterministic ones. Levesque [13] has given a situation calculus account of planning for agents which can sense their environments. Shapiro, Lespérance and Levesque [31] have formalized agent goals and rational actions in the situation calculus. Pinto [23] has proposed a situation calculus-based account of concurrency, natural actions and continuous time.

These initial results have encouraged us in our belief that the situation calculus is well suited to the general problem of providing a formal and computational account of complex dynamic domains and agent behaviors. This paper is a further step in the direction of appropriately enriching the situation calculus for this purpose. Specifically, we suitably modify, and build on, the earlier work mentioned above by Pinto [23] and also by Ternovskaia [32]. This we do by providing an axiomatization of the situation calculus to include concurrency, continuous time and two kinds of actions: (1) Those under the control of an autonomous agent with the "free will" to perform any of these actions at any time, provided their preconditions are met. (2) Natural actions – those under nature's control – which must occur at their predetermined times provided no other actions (natural or agent initiated) occur earlier to prevent them from occurring. Towards that end, we begin by proposing an axiomatization of the concurrent, temporal situation calculus. With these axioms in hand, we define the *legal* situations, which are those that respect the property of natural actions that they must occur at their predicted times, unless something happens to prevent them. We then prove some intuitively desirable properties of these legal situations, for example, that worlds which lack agents with "free will" evolve deterministically, and we establish regression-based methods for verifying that a given situation is legal (the projection problem). One nice consequence of this approach is a situation calculus theory of deductive planning, with continuous time and true concurrency, and where the agent can incorporate external natural event occurrences into her plans.

2 Formal Preliminaries

2.1 The Language of the Situation Calculus

We begin by expanding the situation calculus ontology beyond that of Lin and Reiter [16] and Reiter [26]. The situation calculus is a sorted second order language with the following sorts, function and predicate symbols:

1. There is a sort *action* of *simple* actions. Conceptually, all simple actions are instantaneous, and every function symbol of sort *action* takes a parameter (in the last argument position) denoting the time of the action's occurrence. So, $start\_meeting(person, t)$ might be the instantaneous action of *person* starting a meeting at time t. This will make the notion of concurrent actions relatively unproblematic, which is not the case when actions have durations, and therefore may overlap in complicated ways (Gelfond, Lifschitz and Rabinov [7]).
2. A sort *time* ranging over the reals.
3. A function symbol *time*: $time(a)$ denotes the occurrence time of the simple action a.
4. A function symbol *start*: $start(s)$ denotes the start time of the situation s.
5. A new sort *concurrent*; these are sets of simple actions. We do not axiomatize sets, but instead rely on the standard interpretation of sets and their operations (union, intersection, etc) and relations (membership, subset, etc). This is in the same spirit as our treatment of the sort *time*; we do not axiomatize the reals for this purpose, but instead rely on the standard interpretation of the reals and their operations (addition, multiplication etc) and relations ($<$, $\leq$, etc). To distinguish the sorts *action* and *concurrent*, we use variables $a, a', \ldots$ and $c, c', \ldots$ respectively.
6. As in the sequential situation calculus, we have a sort *situation*, but now ranging over sequences of concurrent actions.
7. A binary function symbol $do : concurrent \times situation \rightarrow situation$, and a constant S_0 denoting the initial situation.
8. The sequential situation calculus has a distinguished predicate symbol *Poss*; $Poss(a, s)$ means that simple action a can be executed in the situation s. We extend *Poss* to concurrent actions, and will write $Poss(c, s)$ to mean that the concurrent action c is possible in situation s.
9. The sequential situation calculus has a distinguished predicate symbol $<$. $s < s'$ means that one can get from situation s to situation s' by a sequence of executable (possible) simple actions. We extend $<$ to concurrent actions, so that $s < s'$ will mean that one can get from situation s to situation s' by a sequence of executable (possible) concurrent actions.
10. Finally, there are predicate symbols *natural*, *coherent*, *legal* and *lntp*, to be described later.

2.2 Foundational Axioms for the Concurrent, Temporal Situation Calculus

Lin and Reiter [16] and Reiter [26] provide foundational axioms for the sequential situation calculus. These need to be generalized to the concurrent, temporal setting, which we now do. Our assumption that all simple actions are instantaneous makes this generalization relatively unproblematic.

We begin by postulating a second order induction axiom:

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

We need the following unique names axioms for situations:

$$S_0 \neq do(c, s),^1 \tag{2}$$

$$do(c, s) = do(c', s') \supset c = c' \wedge s = s'. \tag{3}$$

The time of an action occurrence is the value of that action's temporal argument. So, for each action function $A(\vec{x}, t)$ of our situation calculus language, we need an axiom:

$$time(A(\vec{x}, t)) = t, \tag{4}$$

[1] In what follows, lower case Roman characters will denote variables in formulas, unless otherwise noted. In addition, free variables will always be implicitly universally prenex quantified.

as, for example, in

$$time(start\_meeting(person, t)) = t.$$

Following Lin and Shoham [17], Pinto [23] and others we treat concurrent actions as sets, possibly infinite, of simple actions. As we'll see later, the possibility of infinitely many actions occurring concurrently must be taken seriously, so that the obvious notation $a_1||a_2||\cdots||a_n$ cannot accommodate this possibility. Because concurrent actions are sets of simple actions, we can use the notation $a \in c$ to mean that simple action a is one of the actions of the concurrent action c.

We require that concurrent actions be coherent, which is to say, there is at least one action in the collection, and that all of the (instantaneous) actions in the collection occur at the same time:

$$\begin{array}{l} coherent(c) \equiv \\ \quad (\exists a)a \in c \wedge (\exists t)(\forall a')[a' \in c \supset time(a') = t]. \end{array} \quad (5)$$

We can now extend the function *time* from simple actions to concurrent ones, and we can define the function *start*, as follows:

$$\begin{array}{l} coherent(c) \supset \\ \quad [time(c) = t \equiv (\exists a)(a \in c \wedge time(a) = t)] \\ \quad \wedge start(do(c, s)) = time(c). \end{array} \quad (6)$$

Notice that we do not define the start time of S_0; this is arbitrary, and may (or may not) be specified to be any real number, depending on the application.

Not every action is executable in every situation. Accordingly, we introduce a binary predicate $Poss(c, s)$, meaning that it is possible to execute concurrent action c in situation s. What can we say in general about the preconditions of concurrent actions? At the very least, we need the following:

$$Poss(a, s) \supset Poss(\{a\}, s), \quad (7)$$

$$Poss(c, s) \supset coherent(c) \wedge (\forall a)[a \in c \supset Poss(a, s)]. \quad (8)$$

As we shall see in Section 3.3, the converse of (8) need not hold.

Finally, we need to reconsider the relation $<$ on situations as axiomatized for the sequential, non-temporal situation calculus in Lin and Reiter [16] and Reiter [26]. The intended interpretation of $s < s'$ is that situation s' is reachable from situation s by some sequence of one or more concurrent actions, each concurrent action of which is possible in that situation resulting from executing the actions preceding it in the sequence. Consider the situation

$$\begin{array}{l} do(\{collide(B_1, B_2, 4), end\_lunch(Bill, 4)\}, \\ \quad do(\{start\_meeting(Susan, 6)\}, S_0)), \end{array}$$

in which the time of the second action precedes that of the first. Intuitively, we do not want to consider such an action sequence possible, and we amend the foundational axioms for $<$ in the sequential, non-temporal case accordingly:

$$\neg s < S_0, \quad (9)$$

$$\begin{array}{l} s < do(c, s') \equiv Poss(c, s') \wedge s \leq s' \wedge \\ \quad start(s') \leq time(c). \end{array} \quad (10)$$

Here, $s \leq s'$ is an abbreviation for $s < s' \vee s = s'$. Now, $s < s'$ means that one can get to s' from s by a sequence of possible concurrent actions, and moreover, the times of those action occurrences must be nondecreasing. Notice that we are overloading the predicate $<$; it is used to order situations as well as real numbers in the temporal domain. It will always be clear from context which usage we mean. Finally, notice that the constraint $start(s') \leq time(c)$ in axiom (10) permits action sequences in which the time of an action may be the same as the time of a preceding action, without requiring that these actions occur concurrently. For example,

$$\begin{array}{l} do(\{collide(B_1, B_2, 4), end\_lunch(Bill, 4)\}, \\ \quad do(\{start\_meeting(Susan, 4)\}, S_0)), \end{array}$$

might be a perfectly good situation accessible from S_0. This situation is defined by a sequence of two concurrent actions, each of which has the same occurrence time. We allow for this possibility because often an action occurrence serves as an enabling condition for the simultaneous occurrence of another action. For example, cutting a weighted string at time t enables the action $start\_falling(t)$. Both actions occur at the same time, but conceptually, the falling event happens "after" the cutting. Accordingly, we want to allow the situation $do(\{start\_falling(t)\}, do(\{cut\_string(t)\}, S_0))$.

The axioms (1) - (10) are the foundational axioms for *the concurrent, temporal situation calculus.*

3 Axiomatizing Concurrent Worlds

Most actions (picking up a block, going from one location to another) take time. What use, then, is a theory of actions in which all actions are instantaneous? As observed by Pinto [23] and Ternovskaia [32], the trick for making this work in the situation calculus is to conceive of such actions as *processes*, represented by fluents, and to introduce durationless actions which initiate and terminate these processes. For example, in a blocks world, we might have instantaneous actions $start\_pickup(x, t)$ and $end\_pickup(x, t)$, and the process of picking up x is represented by the fluent $picking\_up(x, t, s)$. $start\_pickup(x, t)$ causes the fluent *picking_up* to be true, $end\_pickup(x, t)$ causes it to be false. In those situations s in which $picking\_up(x, t, s)$ is true, we can describe those properties of the world, for example the position of the agent's hand as a function of t, which must be true during the evolution of the process *picking_up*.

3.1 Successor State Axioms

Reiter [25], building on the ideas of Pednault [21] and of Haas [9] and Schubert [29], proposes a solution to the frame problem for deterministic, nonconcurrent actions in the absence of state constraints. This provides a systematic way of obtaining so-called *successor state axioms* from the effect axioms. We have to generalize these successor state axioms slightly, to take concurrency into account. This is quite straightforward, following the proposals of Pinto in his Ph.D. thesis [23] and Ternovskaia [32]. So, we will write formulas like:

$$Poss(c,s) \supset [picking\_up(x, do(c,s)) \equiv (\exists t) start\_pickup(x,t) \in c \vee picking\_up(x,s) \wedge \neg(\exists t) end\_pickup(x,t) \in c].$$

A more interesting example is due to James Allen [1]. Imagine a door with a spring latch. The door can be unlocked by turning the latch, but the agent must keep the latch turned, for if not, the spring loaded mechanism returns the latch to its locked position. To open the door, the agent must turn the latch, and keep it turned while she pushes on the door. The concurrent latch turning and door pushing causes the door to open. Neither action by itself will open the door. This is easy to do in the situation calculus if we view the action of turning and holding the latch open, which intuitively would have a duration, as a composite of two instantaneous actions, $turn\_latch(t)$ and $release\_latch(t)$, whose effects are to make the fluent $locked(s)$ false and true respectively. In the same spirit, we treat the action of pushing on a door, which also would intuitively have a duration, as a composite of two instantaneous actions $start\_push(t)$ and $end\_push(t)$, whose effects are to make the fluent $pushing(s)$ true and false respectively. The appropriate successor state axiom for *open* is:

$$Poss(c,s) \supset [open(do(c,s)) \equiv (\exists t)[turn\_latch(t) \in c \wedge start\_push(t) \in c] \vee pushing(s) \wedge (\exists t) turn\_latch(t) \in c \vee \neg locked(s) \wedge (\exists t) start\_push(t) \in c \vee open(s)].$$

Those for *pushing* and *locked* are:

$$Poss(c,s) \supset [pushing(do(c,s)) \equiv (\exists t) start\_push(t) \in c \vee pushing(s) \wedge \neg(\exists t) end\_push(t) \in c],$$

$$Poss(c,s) \supset [locked(do(c,s)) \equiv (\exists t) release\_latch(t) \in c \vee locked(s) \wedge \neg(\exists t) turn\_latch(t) \in c].$$

Another interesting example is due to Rob Miller.[2] Turning on the hot water faucet causes hot water to run (denoted by the fluent $hot(s)$); similarly for turning on the cold. Both the hot and cold water faucets share a common spout, so if only the hot water is running, you will burn your hand. This example is of interest because the occurrence of the action of turning on the cold water faucet *cancels* the effect of a co-occurrence of turning on the hot.

$$Poss(c,s) \supset [hot(do(c,s)) \equiv (\exists t) turnonHot(t) \in c \vee hot(s) \wedge \neg(\exists t) turnoffHot(t) \in c].$$

$$Poss(c,s) \supset [cold(do(c,s)) \equiv (\exists t) turnonCold(t) \in c \vee cold(s) \wedge \neg(\exists t) turnoffCold(t) \in c].$$

The following successor state axiom captures the conditions for burning oneself:

$$\begin{aligned} Poss(c,s) \supset [burn(do(c,s)) \equiv \\ &\neg cold(s) \wedge (\exists t) turnonHot(t) \in c \wedge \neg(\exists t) turnonCold(t) \in c \\ &\vee \neg hot(s) \wedge cold(s) \wedge (\exists t) turnonHot(t) \in c \wedge (\exists t) turnoffCold(t) \in c \\ &\vee hot(s) \wedge cold(s) \wedge \neg(\exists t) turnoffHot(t) \in c \wedge (\exists t) turnoffCold(t) \in c \\ &\vee burn(s)].\end{aligned}$$

3.2 Action Precondition Axioms

The approach of Reiter [25] to axiomatizing dynamic worlds in the situation calculus relies on a collection of *action precondition axioms*, one for each action type, and we also rely on such axioms here. These specify necessary and sufficient conditions under which the action is possible. For example, in a blocks world, we might have the following action precondition axiom for the action $pickup(x)$, by a one-handed robot:

$$Poss(pickup(x), s) \equiv [(\forall y) \neg holding(y,s)] \wedge clear(x,s) \wedge \neg heavy(x,s).$$

In general, an action precondition axiom will have the syntactic form:

$$Poss(A(\vec{x},t), s) \equiv \Phi(\vec{x},t,s). \qquad (11)$$

Here, $\Phi(\vec{x},t,s)$ is any first order formula with free variables among $\vec{x}, t$ and s whose only term of sort *situation* is s.

3.3 The Precondition Interaction Problem

Our approach to axiomatizing actions appeals to *action precondition axioms* for specifying the preconditions of simple actions. These have the syntactic form (11). As pointed out by Pelavin [22] and Pinto [23], in the case of action preconditions for concurrent actions, the converse of (8) need not hold. Two simple actions may each be possible, their action preconditions may be jointly consistent, yet intuitively they should not be concurrently possible. Pinto calls this the *precondition interaction problem*. Here is a simple example, after a similar example of Pelavin [22]:

$$Poss(start\_move\_left(t), s) \equiv \neg moving(s).$$

$$Poss(start\_move\_right(t), s) \equiv \neg moving(s).$$

[2] Personal communication.

Intuitively,
$Poss(\{start\_move\_left(t), start\_move\_right(t)\}, s)$ should be false. With reasonable successor state axioms, we should be able to derive something like:

$$Poss(\{start\_move\_left(t), start\_move\_right(t)\}, s) \supset moving\_right(do(\{start\_move\_left(t), start\_move\_right(t)\}, s)),$$

and

$$Poss(\{start\_move\_left(t), start\_move\_right(t)\}, s) \supset moving\_left(do(\{start\_move\_left(t), start\_move\_right(t)\}, s)).$$

So, in the presence of a reasonable state constraint like:

$$\neg[moving\_right(s) \wedge moving\_left(s)],$$

we could derive

$$\neg Poss(\{start\_move\_left(t), start\_move\_right(t)\}, s).$$

These are complicated issues which need to be looked at more closely; see Pinto [23] for a more extensive discussion. We shall ignore them here, except to note that the ideas in the rest of this paper require no commitment, one way or another, to the form of a solution to the precondition interaction problem.

3.4 Infinitely Many Actions Can Co-Occur

Nothing prevents one from writing:

$$Poss(A(x,t), s) \equiv t = 1,$$

in which case $A(x, 1)$ can co-occur, for all x. So if x ranges over the natural numbers (or the reals, or ...) we get lots of possible co-occurrences.

4 Natural Actions

Our focus in this paper is on natural exogenous actions (Pinto [23]), namely those which occur in response to known laws of physics, like a ball bouncing at times determined by Newtonian equations of motion. These laws of physics will be embodied in the action precondition axioms, in the style of Pinto's PhD thesis [23], but in a somewhat more natural form:

$$Poss(bounce(t), s) \equiv is\_falling(s) \wedge \{height(s) + vel(s)[t - start(s)] - 1/2g[t - start(s)]^2 = 0\}.$$

Here, $height(s)$ and $vel(s)$ are the height and velocity, respectively, of the ball at the start of situation s.

Notice that the truth of $Poss(bounce(t), s)$ *does not* mean that the bounce action must occur in situation s, or even that the bounce action must eventually occur. It simply means that the bounce is physically possible at time t in situation s; a *catch* action occurring before t should prevent the bounce action.

We introduce a predicate symbol *natural*, with which the axiomatizer can declare suitable actions to be natural, as, for example, $natural(bounce(t))$.

4.1 Natural Actions and Legal Situations

In the space of all possible situations, we want to single out the legal situations, i.e. those which respect the property of natural actions that they must occur at their predicted times, provided no earlier actions (natural or agent initiated) prevent them from occurring. We capture these legal situations with the following definition:

$$legal(s) \equiv [S_0 \leq s \wedge (\forall a, c, s').natural(a) \wedge Poss(a, s') \wedge do(c, s') \leq s \supset a \in c \vee time(c) < time(a)]. \quad (12)$$

This definition may initially be a bit difficult to understand; the following provides a more intuitive inductive characterization of the legal situations.

Lemma 1 *The foundational axioms imply that the definition (12) is equivalent to the conjunction of the following two sentences:*

$$legal(S_0).$$

$$legal(do(c, s)) \equiv [legal(s) \wedge Poss(c, s) \wedge start(s) \leq time(c) \wedge (\forall a).natural(a) \wedge Poss(a, s) \supset a \in c \vee time(c) < time(a)].$$

Proof:
$\Rightarrow$ Straightforward.
$\Leftarrow$ Use the induction axiom (1), with the definition (12) as induction hypothesis.

4.2 An Example: Enabling Actions

In the discussion following the presentation of axiom (10), we noted the possibility of situations containing two or more concurrent actions with the same occurrence times. We now provide an example where this is a desirable feature of our axiomatization. Consider a scenario in which an agent is holding an object. At some time she releases the object, enabling it to start falling. The *start_falling* action is a natural action, which is to say, it must occur immediately after the release action. For simplicity, assume that once the object starts to fall, it continues falling forever.

$$start(S_0) = 0, \quad holding(S_0), \quad \neg falling(S_0).$$

$$natural(a) \equiv (\exists t)a = start\_falling(t),$$

$$Poss(release(t), s) \equiv holding(s) \wedge start(s) \leq t,$$

$$Poss(start\_falling(t), s) \equiv \neg holding(s) \wedge \neg falling(s) \wedge start(s) \leq t,$$

$$Poss(c, s) \supset [falling(do(c, s)) \equiv (\exists t)start\_falling(t) \in c \vee falling(s)],$$

$$Poss(c, s) \supset [holding(do(c, s)) \equiv (\exists t)catch(t) \in c \vee holding(s) \wedge \neg(\exists t)release(t) \in c].$$

Then, the following is a legal situation:

$$do(\{start\_falling(1)\}, do(\{release(1)\}, S_0)).$$

The following is *not* a legal situation:

$$do(\{start\_falling(2)\}, do(\{release(1)\}, S_0)).$$

4.3 Zeno's Paradox

Legal situations admit infinitely many distinct action occurrences over a finite time interval. Consider the natural action A:

$$Poss(A(t), s) \equiv t = (1 + start(s))/2,$$

with $start(S_0) = 0$. Then for any $n \geq 1$, the situation $do([A(1/2), \ldots, A(1 - 1/2^n)], S_0)$ is legal.[3] This means that if B is another action, natural or not, with $Poss(B(t), s) \equiv t = 1$, then $B(1)$ never gets to be part of any legal situation; it never happens! This is arguably the right intuition, given the idealization of physical reality involved in the axiomatization of A. There does not appear to be any simple way to prevent Zeno's paradox from arising in temporal axiomatizations like ours. Of course, this is not really a paradox, in the sense that such examples do not introduce any inconsistencies into the axiomatization. See E. Davis [4] for a deeper discussion of these issues.

4.4 The Natural World Condition

This is the sentence:

$$(\forall a) natural(a). \qquad (NWC)$$

The Natural World Condition restricts the domain of discourse to natural actions only.

Lemma 2 *The following is a consequence of the foundational axioms and the definition (12):*

$$legal(do(c, s)) \wedge legal(do(c', s)) \wedge NWC \supset c = c'.$$

Proof:
Suppose, for fixed c, c', s, that $legal(do(c, s))$, $legal(do(c', s))$, and NWC. Then, by Lemma 1, $Poss(c, s)$, and $Poss(c', s)$, and therefore, by (8), $coherent(c)$ and $coherent(c')$.

1. First we prove $a \in c \supset a \in c'$. Suppose, to the contrary, that there is some α, with $\alpha \in c$, but $\alpha \notin c'$. By NWC, $natural(\alpha)$. Since $Poss(c, s)$, we have $Poss(\alpha, \sigma)$ by (8). Hence, since $legal(do(c', s))$, we conclude, with the help of Lemma 1, that

   $$time(c') < time(\alpha). \qquad (13)$$

   Since $coherent(c')$, c' is nonempty by (5), so there exists β such that $\beta \in c'$. Since $Poss(c', s)$, by (8), we conclude $Poss(\beta, s)$. Since $natural(\beta)$, we conclude, by Lemma 1, that

   $$\beta \in c \vee time(c) < time(\beta). \qquad (14)$$

   But, since $coherent(c)$ and $coherent(c')$, we have $time(\alpha) = time(c)$ and $time(\beta) = time(c')$, and these, together with (13), imply $time(\beta) < time(c)$. This, together with (14), implies $\beta \in c$. Since $coherent(c)$, $time(\beta) = time(c)$, which contradicts $time(\beta) < time(c)$.

2. The proof that $a \in c' \supset a \in c$ is entirely symmetric to the previous one.

Combining 1 and 2, we conclude $c = c'$.

□

Intuitively, the above lemma tells us that natural worlds are *deterministic*: If there is a legal successor situation, it is unique. The following theorem extends Lemma 2 to *histories*: When there are only natural actions, the world evolves in a unique way, if it evolves at all.

Theorem 1 *The foundational axioms and the definition (12) entail the following:*

$$legal(s) \wedge legal(s') \wedge NWC \supset S_0 \leq s \leq s' \vee S_0 \leq s' \leq s.$$

Proof:
The proof is by induction on s, using the induction axiom (1). The case $s = S_0$ is immediate, so assume the result for s, and suppose $legal(do(c, s)) \wedge legal(s') \wedge NWC$. We must prove $S_0 \leq do(c, s) \leq s' \vee S_0 \leq s' \leq do(c, s)$. Since $legal(do(c, s))$, then by Lemma 1, $legal(s)$ and $Poss(c, s)$. Hence, by the induction hypothesis, we conclude $S_0 \leq s \leq s' \vee S_0 \leq s' \leq s$.

Case 1: $S_0 \leq s \leq s'$.

Case 1.1: $s = s'$. Then, because $Poss(c, s)$, $s' < do(c, s)$ by axiom (10), so, $S_0 \leq s' \leq do(c, s)$.

Case 1.2: $s < s'$.

We require the following two results, each of which is provable by induction on s'.

$$(\forall s, s').s < s' \supset (\exists c) do(c, s) \leq s', \qquad (15)$$

$$(\forall s, s').legal(s') \wedge s \leq s' \supset legal(s). \qquad (16)$$

Now, by (15), $do(c', s) \leq s'$ for some c'. Moreover, because $legal(s')$, we have, by (16), $legal(do(c', s))$. Since also $legal(do(c, s))$, then by Lemma 2, $c = c'$, and we conclude that $S_0 \leq do(c, s) \leq s'$.

Case 2: $S_0 \leq s' \leq s$.

Since $Poss(c, s)$, then by axiom (10), $s' < do(c, s)$, and by the transitivity of $\leq$ (provable by induction), we conclude $S_0 \leq s' \leq do(c, s)$.

4.5 Least Natural Time Points

The following definition plays a central role in theorizing about natural actions:

$$\begin{aligned} &lntp(s, t) \equiv \\ &(\exists a)[natural(a) \wedge Poss(a, s) \wedge time(a) = t] \wedge \\ &(\forall a)[natural(a) \wedge Poss(a, s) \supset time(a) \geq t]. \end{aligned} \qquad (17)$$

[3] $do([a_1, \ldots, a_n], s)$ abbreviates the situation reached from s by performing the actions $a_1, \ldots, a_n$ in sequence.

Intuitively, the *least natural time point* is the earliest time during situation s at which a natural action can occur.

Remark 1 *(17) entails the following:*
$$lntp(s,t) \wedge lntp(s,t') \supset t = t'.$$

So, when it exists, the least natural time point is unique. The least natural time point need not exist, for example, when $(\forall a).natural(a) \equiv (\exists x,t)a = B(x,t)$, where x ranges over the nonzero natural numbers, and $Poss(B(x,t),s) \equiv t = start(s) + 1/x$.

Lemma 3 *Our situation calculus axioms entail:*
$natural(a) \wedge legal(do(c,s)) \wedge a \in c \supset lntp(s, time(a))$.

Proof:
Assume that $natural(a)$, $legal(do(c,s))$ and $a \in c$, for fixed a, c, s. Since $legal(do(c,s))$, we know, by Lemma 1, that $Poss(c,s)$, and therefore, by (8) that $Poss(a,s)$ and $coherent(c)$. We must prove $lntp(s, time(a))$. By the definition of $lntp$, it is sufficient to prove $(\forall a').natural(a') \wedge Poss(a',s) \supset time(a') \geq time(a)$. So, for fixed a', assume $natural(a') \wedge Poss(a',s)$. We prove $time(a') \geq time(a)$. Since $legal(do(c,s))$, we know, by Lemma 1, that
$$a' \in c \vee time(c) < time(a'). \qquad (18)$$
Moreover, since $coherent(c)$ and since $a \in c$, we conclude from (5) that $time(a) = time(c)$. (18) gives two cases:
Case 1: $time(c) < time(a')$. Since $time(a) = time(c)$, we have $time(a') > time(a)$.
Case 2: $a' \in c$. Since $coherent(c)$, we infer $time(a') = time(c)$. Since $time(a) = time(c)$, then $time(a') = time(a)$.

□

So, whenever $do(c,s)$ is legal, c's natural actions all co-occur at the least natural time point of s. All the actions that must co-occur first, according to the "laws of motion", actually do co-occur.

In the case of a domain closure assumption on natural actions, we can give an explicit formula for $lntp(s,t)$. So, suppose we have the following domain closure axiom:
$$natural(a) \equiv (\exists \vec{x},t)a = A_1(\vec{x},t) \vee \cdots \vee (\exists \vec{z},t)a = A_n(\vec{z},t), \qquad (19)$$
together with the associated declarations (4):
$$time(A_1(\vec{x},t)) = t, \quad \vdots \quad time(A_n(\vec{z},t)) = t. \qquad (20)$$

Lemma 4 *(17), (19) and (20) entail the following:*
$$lntp(s,t) \equiv [(\exists \vec{x})Poss(A_1(\vec{x},t),s) \vee \cdots \vee (\exists \vec{z})Poss(A_n(\vec{z},t),s)] \wedge (\forall \vec{x},t')[Poss(A_1(\vec{x},t'),s) \supset t' \geq t] \wedge \cdots \wedge (\forall \vec{z},t')[Poss(A_n(\vec{z},t'),s) \supset t' \geq t].$$

The Least Natural Time Point Condition

In view of the possibility of "pathological" axiomatizations, for which the least natural time point may not exist (see comments following Remark 1), we introduce the following sentence:
$$(\forall s).(\exists a)[natural(a) \wedge Poss(a,s)] \supset (\exists t)lntp(s,t). \qquad (LNTPC)$$
Normally, it will be the responsibility of the axiomatizer to prove, usually by induction, that his axioms entail $LNTPC$.

Theorem 2 *Our situation calculus axioms entail:*
$$LNTPC \supset legal(do(c,s)) \equiv \{legal(s) \wedge Poss(c,s) \wedge start(s) \leq time(c) \wedge [(\forall a).natural(a) \wedge time(a) \leq time(c) \supset [a \in c \equiv Poss(a,s) \wedge lntp(s,time(a))]]\}.$$

Proof:
$\Rightarrow$
Assume $LNTPC$, and for fixed c, s, $legal(do(c,s))$. By Lemma 1, we conclude $legal(s)$, $Poss(c,s)$ and $start(s) \leq time(c)$. So we must prove:
$$(\forall a).natural(a) \wedge time(a) \leq time(c) \supset [a \in c \equiv Poss(a,s) \wedge lntp(s,time(a))].$$
So, for fixed a, assume $natural(a) \wedge time(a) \leq time(c)$; we prove: $a \in c \equiv Poss(a,s) \wedge lntp(s,time(a))$.

1. Assume $a \in c$; we prove $Poss(a,s) \wedge lntp(s,time(a))$. Since $Poss(c,s)$, we know $Poss(a,s)$ by (8). By Lemma 3, we know $lntp(s,time(a))$.
2. Assume $Poss(a,s) \wedge lntp(s,time(a))$; we prove $a \in c$. Since $legal(do(c,s))$, then by Lemma 1: $a \in c \vee time(c) < time(a)$. Since $time(a) \leq time(c)$, we conclude $a \in c$.

$\Leftarrow$
Assume $LNTPC$, and for fixed c, s, $legal(s)$, $Poss(c,s)$, $start(s) \leq time(c)$, and
$$(\forall a).natural(a) \wedge time(a) \leq time(c) \supset [a \in c \equiv Poss(a,s) \wedge lntp(s,time(a))]. \qquad (21)$$
By Lemma 1, we must prove: $(\forall a').natural(a') \wedge Poss(a',s) \supset a' \in c \vee time(c) < time(a')$. So, for fixed a', assume $natural(a') \wedge Poss(a',s) \wedge time(a') \leq time(c)$. We must prove that $a' \in c$. By (21), it is sufficient to prove $Poss(a',s) \wedge lntp(s,time(a'))$. We already know that $Poss(a',s)$, so we must prove $lntp(s,time(a'))$. By $LNTPC$, we have $lntp(s,\tau)$ for some τ, so there is an α such that $natural(\alpha) \wedge Poss(\alpha,s) \wedge time(\alpha) = \tau$. We prove that $time(a') = time(\alpha)$, from which the desired conclusion $lntp(s,time(a'))$ follows. Now, because $Poss(a',s)$ and $natural(a')$, we have $time(a') \geq time(\alpha)$. Therefore, because $time(a') \leq time(c)$, $time(\alpha) \leq time(c)$,

so by (21), $\alpha \in c$. Since $Poss(c,s)$, by (8) and (5), $time(\alpha) = time(c)$. Since $time(a') \geq time(\alpha) = time(c)$, and since also $time(a') \leq time(c)$, we conclude $time(a') = time(\alpha)$. Hence, we have proved $a' \in c$.

Theorem 3 *Our situation calculus axioms entail the following:*

$$\begin{aligned}&LNTPC \wedge NWC \supset\\ &\quad legal(do(c,s)) \equiv \{legal(s) \wedge Poss(c,s) \wedge\\ &\qquad start(s) \leq time(c) \wedge\\ &\qquad\quad (\forall a)[a \in c \equiv Poss(a,s) \wedge lntp(s, time(a))]\}.\end{aligned}$$

Proof:
The $\Leftarrow$ direction follows immediately from Theorem 2. To prove the $\Rightarrow$ direction, assume that $LNTPC$ and NWC, and, for fixed c and s, that $legal(do(c,s))$. We must prove: $(\forall a).a \in c \equiv Poss(a,s) \wedge lntp(s, time(a))$. By Lemma 1, we have $Poss(c,s)$, and

$$(\forall a).Poss(a,s) \supset a \in c \vee time(c) < time(a). \quad (22)$$

1. Suppose $a \in c$. Since $Poss(c,s)$, then by (8) we know that $Poss(a,s)$. It remains to prove $lntp(s, time(a))$. This follows immediately from Lemma 3.
2. Assume $Poss(a,s) \wedge lntp(s, time(a))$. We must prove $a \in c$. Since $Poss(c,s)$, then by (8), $coherent(c)$ and hence c is nonempty. So, there exists α such that $\alpha \in c$. Since $Poss(c,s)$, then by (8), we know that $Poss(\alpha,s)$ and $coherent(c)$, so by (5), $time(\alpha) = time(c)$. Since $lntp(s, time(a))$, we have that $time(\alpha) = time(c) \geq time(a)$. So, by (22), we conclude $a \in c$.

□

This theorem informs us that for natural worlds satisfying $LNTPC$, we obtain the next legal situation from the current one by assembling into c all the possible actions occurring at the least natural time point of the current situation, provided this collection of natural actions is possible, and the least natural time point is greater than or equal to the start time of the current situation. Intuitively, this is as it should be for natural worlds. Theorem 3 provides the theoretical foundation for a situation calculus-based simulator for physical systems (Kelley [10]).

The Concurrent Natural Actions Assumption

This is the following sentence:

$$\begin{aligned}&coherent(c) \wedge (\forall a)[a \in c \supset natural(a) \wedge Poss(a,s)]\\ &\qquad \supset Poss(c,s). \qquad (CNAA)\end{aligned}$$

This says that a coherent collection of natural actions is possible if each individual action in the collection is possible. In other words, the precondition interaction problem does not arise for co-occurring natural actions. This seems to be an assumption about the accuracy with which the physics of the world has been modeled by "equations of motion", in the sense that if these equations predict a co-occurrence, then this co-occurrence really happens in the physical world, so that in our situation calculus model of that world, this co-occurrence should be possible.

Using this and (8), we obtain the following:

Lemma 5 *Our situation calculus axioms entail:*

$$\begin{aligned}&CNAA \wedge (\forall a)[a \in c \supset natural(a)] \supset\\ &\quad Poss(c,s) \equiv coherent(c) \wedge (\forall a)[a \in c \supset Poss(a,s)].\end{aligned}$$

Corollary 1 *Our situation calculus axioms entail:*

$$\begin{aligned}&LNTPC \wedge NWC \wedge CNAA \supset\\ &\quad legal(do(c,s)) \equiv \{legal(s) \wedge (\exists a)a \in c \wedge\\ &\qquad start(s) \leq time(c) \wedge\\ &\qquad\quad (\forall a)[a \in c \equiv Poss(a,s) \wedge lntp(s, time(a))]\}.\end{aligned}$$

Proof:
By Theorem 3, it is sufficient to prove that

$$\begin{aligned}&LNTPC \wedge NWC \wedge CNAA \supset\\ &\quad (\exists a)a \in c \ \wedge start(s) \leq time(c) \wedge\\ &\qquad (\forall a)[a \in c \equiv Poss(a,s) \wedge lntp(s, time(a))]\\ &\quad \equiv\\ &\quad Poss(c,s) \wedge start(s) \leq time(c) \wedge\\ &\qquad (\forall a)[a \in c \equiv Poss(a,s) \wedge lntp(s, time(a))].\end{aligned}$$

So, assume $LNTPC \wedge NWC \wedge CNAA$. The $\Leftarrow$ direction is straightforward, using Lemma 5, and the definition (5) of *coherent*. To prove the $\Rightarrow$ direction, assume, for fixed a, c, and s, that $(\exists a)a \in c$, and $(\forall a)[a \in c \equiv Poss(a,s) \wedge lntp(s, time(a))]$. It is sufficient to prove $Poss(c,s)$. By Lemma 5 and NWC, it is sufficient to prove $coherent(c) \wedge (\forall a)[a \in c \supset Poss(a,s)]$. The second conjunct follows from NWC, so we must prove $coherent(c)$, which is equivalent, by (5), to $(\exists a)a \in c \wedge (\exists t)(\forall a)[a \in c \supset time(a) = t]$. The first conjunct is given by assumption, so we are left with proving $(\exists t)(\forall a)[a \in c \supset time(a) = t]$. Suppose, for the purposes of deriving a contradiction, that $(\forall t)(\exists a).a \in c \wedge time(a) \neq t$. By assumption, $(\exists a)a \in c$, so, for fixed α, suppose $\alpha \in c$. Then we must have $(\exists a).a \in c \wedge time(a) \neq time(\alpha)$, so for some fixed α', we have $\alpha' \in c \wedge time(\alpha') \neq time(\alpha)$. Because $\alpha \in c$ and $\alpha' \in c$, we infer $lntp(s, time(\alpha))$ and $lntp(s, time(\alpha'))$, so by Remark 1, $time(\alpha') = time(\alpha)$, contradiction.

5 Some Consequences of this Approach

5.1 Planning with Concurrent and Natural Actions

The classical specification of the planning task is by Green [8], and concerns a single agent in complete control of all actions that can be performed in the world being modeled. A ground situation term σ is a *plan* for G iff

$$Axioms \models S_0 \leq \sigma \wedge G(\sigma).$$

Here, *Axioms* provide the relevant background theory.

In view of the approach of this paper, we can now generalize Green's definition to the case of a single agent with the "free will" to perform a repertoire of actions under her control, and a complementary set of natural actions under nature's control: A ground situation term σ is a *plan* for G iff

$$Axioms \models legal(\sigma) \wedge G(\sigma).$$

Here, *Axioms* includes the foundational axioms and the associated definitions of this paper. It will also include action precondition and successor state axioms for the actions under consideration, unique names axioms for actions, and axioms specifying the initial world situation.

This means we now have a situation calculus specification of deductive planning, with continuous time and true concurrency, and where the agent can incorporate external natural event occurrences into her plans.

With the exception of Levesque's work on planning for agents with perceptual actions [13], this appears to be the first significant generalization of Green's classical formulation of deductive planning.

5.2 Regression

Lemma 1 provides a basis for establishing legality by regression (Waldinger [33], Pednault [21], Reiter [25]). When $LNTPC$ holds, Theorem 2 provides a better regression mechanism, and when also NWC is true, we can use Theorem 3.

While our focus in the previous section was on *specifying* what counts as a plan for agents in concurrent worlds with natural actions, we note that a regression-style planning algorithm could be based on Theorem 2, at least in the case when $LNTPC$ holds.

5.3 Example

We consider a generalization of an example that Pinto used in his Ph.D. thesis [23], which involves two natural actions and an agent's "free will". Two perfectly elastic balls, B_1 and B_2, are rolling parallel to each other on a frictionless floor, between two parallel walls. Their motions are orthogonal to the walls, so we can expect them to bounce indefinitely between the two walls, unless the agent catches one or both of them, which he is free to do. Take the first wall to be the y-axis, the second wall to be distance $W > 0$ to the right of the first wall, and the balls start their motion towards the right, beginning at the first wall. Initially, B_2 has twice the velocity of B_1.

Initial Situation

$W > 0,\ pos(B_1, S_0) = pos(B_2, S_0) = 0,\ vel(B_1, S_0) > 0,$
$vel(B_2, S_0) = 2 * vel(B_1, S_0),\ start(S_0) = 0,\ B_1 \neq B_2,$
$natural(a) \equiv (\exists t).a = bounce(B_1, t) \vee a = bounce(B_2, t).$

Action Precondition Axioms

$$\begin{aligned}
&Poss(bounce(b,t),s) \equiv \\
&\quad [b = B_1 \vee b = B_2] \wedge vel(b,s) \neq 0 \wedge \\
&\quad [vel(b,s) > 0 \supset t = start(s) + \tfrac{W - pos(b,s)}{vel(b,s)}] \wedge \\
&\quad [vel(b,s) < 0 \supset t = start(s) - \tfrac{pos(b,s)}{vel(b,s)}].
\end{aligned}$$

$$\begin{aligned}
&Poss(catch(b,t),s) \equiv \\
&\quad vel(b,s) \neq 0 \wedge \neg Poss(bounce(b,t),s).
\end{aligned}$$

Successor State Axioms

$$\begin{aligned}
&Poss(c,s) \supset pos(b, do(c,s)) = pos(b,s) + \\
&\quad vel(b,s) * (time(c) - start(s)).
\end{aligned}$$

$$\begin{aligned}
&Poss(c,s) \supset vel(b, do(c,s)) = \\
&\quad \textbf{if } (\exists t) catch(b,t) \in c \textbf{ then } 0 \\
&\quad \textbf{else if } (\exists t) bounce(b,t) \in c \textbf{ then } -vel(b,s) \\
&\quad \textbf{else } vel(b,s).
\end{aligned}$$

Least Natural Time Points

Using Lemma 4, and induction, we can show that:

$$\begin{aligned}
&lntp(s,t) \equiv Poss(bounce(B_2,t),s) \vee \\
&\quad Poss(bounce(B_1,t),s) \wedge (\forall t')\neg Poss(bounce(B_2,t'),s).
\end{aligned}$$

It follows that $LNTPC$ holds. Notice that we have not proved that $(\forall s)(\exists t) lntp(s,t)$. In fact, this is false; A $\{catch(B_1,t), catch(B_2,t)\}$ concurrent action could intervene in some situation. This would prevent any *bounce* action from occurring in the resulting situation, so this resulting situation would have no least natural time point.

For $n = 1, 2, \ldots$ define $\tau_n = \frac{n * W}{2 * vel(B_1, S_0)}$. The τ_i are the times at which ball B_2 will bounce, assuming no $catch(B_2, t)$ actions occur. Then the following sequence of concurrent actions leads to a legal situation, provided the two actions in the concurrent actions are jointly possible:
$\{bounce(B_2, \tau_1)\}, \{bounce(B_1, \tau_2), bounce(B_2, \tau_2)\},$
$\{bounce(B_2, \tau_3)\}, \{bounce(B_1, \tau_4), bounce(B_2, \tau_4)\},$
$\{catch(B_2, [\tau_4 + \tau_5]/2)\}, \{bounce(B_1, \tau_6)\},$
$\{bounce(B_1, \tau_8)\}, \{catch(B_1, \tau_9)\}.$
This could be proved by regression, using Theorem 2, but doing so by hand would be too tedious here.

Notice that the following action, performed in S_0 may, or may not lead to a legal situation: $\{catch(B_1, \tau_1/2), catch(B_2, \tau_1/2\}$. That depends on whether or not the two catch actions are jointly possible. If the agent is a one-handed robot, then any axiomatization of the agent's abilities will include

$$\begin{aligned}
&(\forall c, s).Poss(c,s) \supset \\
&\quad \neg(\exists x, y, t).x \neq y \wedge catch(x,t) \in c \wedge catch(y,t) \in c.
\end{aligned}$$

This is another instance of Pinto's precondition interaction problem. Notice that all the results of this paper are independent of any assumptions about this problem.

5.4 Discrete Time

Nothing in the previous discussion requires time to be continuous. We consider here the consequences of relaxing this assumption. Specifically, we imagine the time line to be the integers (positive and negative), so that the sort *time* now ranges over these. Notice that Zeno's paradox cannot arise in this setting. When time is discrete, we have the following:

Lemma 6 *Suppose:*

1. *The time line ranges over the integers.*
2. *The domain closure axiom (19) holds for natural actions.*
3. *Each natural action $A_i(\vec{x}, t)$ has an action precondition axiom logically equivalent to one of the form:*

$$Poss(A_i(\vec{x}, t), s) \equiv \Phi_i(\vec{x}, t, s) \wedge start(s) \leq t, \quad (23)$$

where $\Phi_i(\vec{x}, t, s)$ is a first order formula with free variables among $\vec{x}, t, s$.

Then the least natural time point condition is satisfied: Our situation calculus axioms entail $LNTPC$.

Proof: (Slightly informal)
Suppose that $\mathcal{M}$ is model of our axioms in which the *time* sort ranges over the integers, and let σ be a variable assignment such that

$$\mathcal{M}, \sigma \models natural(a) \wedge Poss(a, s).$$

We prove that $\mathcal{M}, \sigma \models (\exists t) lntp(s, t)$. By axiom (4), domain closure (19), and the assumption (23),

$$\mathcal{M}, \sigma \models start(s) \leq time(a).$$

So, for any natural action a, $time(a)$ is bounded from below in $\mathcal{M}$ by $start(s)$. Since time is discrete, there is a least $t \geq start(s)$ for which $\mathcal{M}, \sigma \models (\exists a).natural(a) \wedge Poss(a, s) \wedge time(a) = t$. Hence, $\mathcal{M} \models LNTPC$.

□

The above lemma is actually more general than it initially appears to be. To begin, without some kind of domain closure assumption on natural actions, it is impossible to prove the legality of any situation. Secondly, it is quite natural to impose the temporal constraint $start(s) \leq t$ on action precondition axioms, as in (23), or, as we did in the bouncing balls example, omit this constraint from the axioms when it is known from the problem description that the time variable t necessarily satisfies this constraint.

6 Discussion and Conclusions

By basing it on the language $\mathcal{A}$ of Gelfond and Lifschitz [6], Baral and Gelfond [3] provide a semantic account of concurrency which, although not formulated in the situation calculus, has many similarities with ours. The principal difference is that Baral and Gelfond focus exclusively on concurrency, so their ontology does not include time or natural actions. Moreover, $\mathcal{A}_C$, their action representation language, is propositional; while it would be possible to translate $\mathcal{A}_C$ theories into the situation calculus, the resulting sentences would be in the *monadic* situation calculus, and therefore would be less general than the logical theories to which our approach applies.

There have been a few earlier papers on formalizing natural actions and continuous time. Shanahan's approach [30] is embedded in the *event calculus* (Kowalski and Sergot [11]); Sandewall [27] relies on a temporal logic. Accordingly, these proposals are difficult to compare with ours, based as it is on the situation calculus. Below, we provide a comparison along one dimension: abductive planning, which seems to be required by these proposals, and the deductive planning approach of the situation calculus.

The approaches of Pinto [23] and Pinto and Reiter [24], and of Miller and Shanahan [20] come closest to that of this paper in that they also rely on the situation calculus. These all differ from us in proposing something like an "actual" path in the tree of situations, corresponding to the way in which the world actually evolves. Both proposals suffer from what might be called the "premature minimization problem", which amounts to the assumption that all action occurrences (natural as well as those under the free will of an agent) are either specified as part of the axiomatization, or are inferable from it. Closure, in the form of the minimization of action occurrences, is enforced by suitably circumscribing these axioms. This means that any agent initiated action not deducible from the axioms is assumed not to have happened. Therefore, if A is an action which the agent could have, but did not initiate according to the axioms, then the "actual" path of world actions will not include A. In other words, *hypothetical* world evolutions, in which all possible agent initiated actions are permitted, are excluded from the minimized axioms. The only legal situations (in the sense of this paper) under these approaches are those on the actual path. Now one of the great advantages of the situation calculus is that all possible world histories are explicitly available in the language, as object language terms.[4] It is precisely this feature which permits a *deductive* approach to hypothetical reasoning about possible world futures. This, in turn, pro-

[4]That is the role of the function symbol *do*. Like *cons* in LISP, it creates sequences, in this case terms representing world histories, i.e. sequences of actions.

vides for a deductive account for planning. Unfortunately, by prematurely minimizing action occurrences, the above approaches preclude a deductive approach to hypothetical reasoning and planning; analogous versions of the definition of planning of Section 5.1 cannot be given. Instead, an *abductive* account must be used. Intuitively, one can see why this must be so. Since, after circumscribing the axioms, there is just one actual path, the possibility of other actual paths can be considered only by hypothetically postulating other free will action occurrences, closing the axioms with respect to these hypothetical occurrences, and testing whether the resulting axioms are satisfiable and entail the goal condition.

This phenomenon of planning by abduction is quite widespread; it is used, for example, in the event calculus [5] and in Allen's temporal logic [1]. In fact, it is the only way to do planning in logics which do not provide for branching futures. Unfortunately, abductive planning suffers from a number of drawbacks, when compared with the deductive approach:

1. It is a metalevel task; the planner must leave the object language to generate a candidate collection of atoms of the form $occurs(A, T)$, test the consistency of these atoms with respect to the object level axiomatization of the domain, then return to the object level to prove the goal sentence relative to the enlarged axiom set.
2. Because of the above consistency test, abductive plans are not even recursively enumerable for first order axiomatizations, in contrast to the deductive case.
3. Even if we ignore the noncomputability of the consistency test, from a computational point of view there are at least two theorem proving tasks for abductive planners: the consistency test (which normally must be performed several times), together with the goal-entailment proof.
4. It is not difficult to imagine settings where a robot agent needs to establish that a plan does *not* exist. In the deductive case, this amounts to establishing that $Axioms \models \neg(\exists s)G(s)$, and, at least formally, is no more problematic than the planning problem. For abductive planners, it is not at all obvious what such a proof might look like, or how it could be constructed; the robot must show, again at the metalevel, that there is no finite set of atoms of the form $occurs(A, T)$, consistent with the background axioms, which entails the goal.
5. As observed by Pelavin [22], for concurrent actions, abductive planning can yield incorrect plans in the presence of partial world descriptions.

Pelavin [22] addresses the formalization of concurrent actions by extending the ontology of Allen's linear time logic [1] to include histories to represent branching futures, and suitable modal operators semantically characterized with respect to these histories. This allows a deductive account of planning within a temporal logic, but at the expense of a rather complicated logic.

Acknowledgements:

Thanks to the other members of the University of Toronto Cognitive Robotics Group (Yves Lespérance, Hector Levesque, Fangzhen Lin, Daniel Marcu and Richard Scherl) for their comments and suggestions. This paper was strongly influenced by Javier Pinto's approach to concurrency and natural actions in his Ph.D. thesis. I have also benefited from extensive discussions about temporal reasoning with Javier, and with Mikhail Soutchanski and Eugenia Ternovskaia. Rob Miller provided useful comments on an earlier draft of this paper, and Rob and Murray Shanahan helped me better understand their approach to natural actions in the situation calculus. Todd Kelley first pointed out to me the importance of accommodating enabling actions; this motivated the current form of the foundational axiom (10) which now allows such actions to be easily represented. Hesham Khalil carefully read an earlier draft of this paper, and made many useful suggestions; in particular, he noted the need for the foundational axiom (7). This research was supported by grants from the Natural Sciences and Engineering Research Council of Canada, the Institute for Robotics and Intelligent Systems of the Government of Canada, and the Information Technology Research Centre of the Government of Ontario.

References

[1] J.F. Allen. Temporal reasoning and planning. In J.F. Allen, H.A. Kautz, R.N. Pelavin, and J.D. Tenenberg, editors, *Reasoning about Plans*, pages 1–68. Morgan Kaufmann Publishers, San Francisco, CA, San Mateo, CA, 1991.

[2] F. Bacchus, J.Y. Halpern, and H.J. Levesque. Reasoning about noisy sensors in the situation calculus. In *Proc. IJCAI'95*, pages 1933–1940, 1995.

[3] C. Baral and M. Gelfond. Reasoning about effects of concurrent actions. *Journal of Logic Programming*, 1996. to appear.

[4] Ernest Davis. Infinite loops in finite time. Technical report, Department of Computer Science, New York University, February, 1992.

[5] K. Eshghi. Abductive planning with event calculus. In *Proceedings of the Fifth International Conference on Logic Programming*, pages 562–579. MIT Press, 1988.

[6] M. Gelfond and V. Lifschitz. Representing actions in extended logic programs. In *Proc. Joint Int. Conf. and Symp. on Logic Programming*, pages 559–573, 1992.

[7] M. Gelfond, V. Lifschitz, and A. Rabinov. What are the limitations of the situation calculus? In *Working Notes, AAAI Spring Symposium Series on the Logical Formalization of Commonsense Reasoning*, pages 59–69, 1991.

[8] C.C. Green. Theorem proving by resolution as a basis for question-answering systems. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 183–205. American Elsevier, New York, 1969.

[9] A. R. Haas. The case for domain-specific frame axioms. In F. M. Brown, editor, *The frame problem in artificial intelligence. Proceedings of the 1987 workshop*, pages 343–348, Los Altos, California, 1987. Morgan Kaufmann Publishers, San Francisco, CA.

[10] T.G. Kelley. Modeling complex systems in the situation calculus: A case study using the Dagstuhl steam boiler problem. In L.C. Aiello, J. Doyle, and S.C. Shapiro, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifth International Conference (KR'96)*. Morgan Kaufmann Publishers, San Francisco, CA, 1996.

[11] R.A. Kowalski and M.J. Sergot. A logic-based calculus of events. *New Generation Computing*, 4:267, 1986.

[12] Y. Lespérance, H.J. Levesque, F. Lin, D. Marcu, R. Reiter, and R. Scherl. A logical approach to high-level robot programming – a progress report. In *Control of the Physical World by Intelligent Systems, Working Notes of the 1994 AAAI Fall Symposium*, November, 1994. New Orleans, LA.

[13] H.J. Levesque. What is planning in the presence of sensing? In *Proceedings of the National Conference on Artificial Intelligence*, 1996. To appear.

[14] H.J. Levesque, R. Reiter, Y. Lespérance, F. Lin, and R. Scherl. GOLOG: a logic programming language for dynamic domains. *Journal of Logic Programming, Special Issue on Actions*, 1996. To appear.

[15] F. Lin. Specifying the effects of indeterminate actions. In *Proceedings of the National Conference on Artificial Intelligence*, 1996. To appear.

[16] F. Lin and R. Reiter. State constraints revisited. *J. of Logic and Computation, special issue on actions and processes*, 4:655–678, 1994.

[17] F. Lin and Y. Shoham. Concurrent actions in the situation calculus. In *Proceedings of the National Conference on Artificial Intelligence*, pages 590–595, 1992.

[18] D. Marcu, Y. Lespérance, H. Levesque, F. Lin, R. Reiter, and R. Scherl. Foundations of a logical approach to agent programming. In M. Wooldridge, J.P. Muller, and M. Tambe, editors, *Intelligent Agents Volume II – Proceedings of the 1995 Workshop on Agent Theories, Architectures, and Languages (ATAL-95)*, pages 331–346. Springer-Verlag, Lecture Notes in Artificial Intelligence, 1996. To appear.

[19] J. McCarthy. Situations, actions and causal laws. Technical report, Stanford University, 1963. Reprinted in Semantic Information Processing (M. Minsky ed.), MIT Press, Cambridge, Mass., 1968, pp. 410-417.

[20] R. Miller and M. Shanahan. Narratives in the situation calculus. *The Journal of Logic and Computation (Special Issue on Actions and Processes)*, 4:513–530, 1994.

[21] E.P.D. Pednault. ADL: Exploring the middle ground between STRIPS and the situation calculus. In R.J. Brachman, H. Levesque, and R. Reiter, editors, *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning (KR'89)*, pages 324–332. Morgan Kaufmann Publishers, San Francisco, CA, 1989.

[22] R.N. Pelavin. Planning with simultaneous actions and external events. In J.F. Allen, H.A. Kautz, R.N. Pelavin, and J.D. Tenenberg, editors, *Reasoning about Plans*, pages 127–211. Morgan Kaufmann Publishers, San Francisco, CA, San Mateo, CA, 1991.

[23] J.A. Pinto. *Temporal Reasoning in the Situation Calculus*. PhD thesis, University of Toronto, Department of Computer Science, 1994.

[24] J.A. Pinto and R. Reiter. Adding a time line to the situation calculus. In *Second Symposium on Logical Formalizations of Commonsense Reasoning*, pages 172–177, Austin, Texas, Jan. 11-13, 1993.

[25] R. Reiter. The frame problem in the situation calculus: a simple solution (sometimes) and a completeness result for goal regression. In Vladimir Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*, pages 359–380. Academic Press, San Diego, CA, 1991.

[26] R. Reiter. Proving properties of states in the situation calculus. *Artificial Intelligence*, 64:337–351, 1993.

[27] E. Sandewall. Combining logic and differential equations for describing real-world systems. In R.J. Brachman, H. Levesque, and R. Reiter, editors, *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning (KR'89)*, pages 412–420. Morgan Kaufmann Publishers, San Francisco, CA, 1989.

[28] R. Scherl and H.J. Levesque. The frame problem and knowledge producing actions. In *Proc. AAAI-93*, pages 689–695, Washington, DC, 1993.

[29] L.K. Schubert. Monotonic solution of the frame problem in the situation calculus: an efficient method for worlds with fully specified actions. In H.E. Kyberg, R.P. Loui, and G.N. Carlson, editors, *Knowledge Representation and Defeasible Reasoning*, pages 23–67. Kluwer Academic Press, 1990.

[30] M.P. Shanahan. Representing continuous change in the event calculus. In *Proceedings ECAI 90*, pages 598–603, 1990.

[31] S. Shapiro, Y. Lespérance, and H.J. Levesque. Goals and rational action in the situation calculus – a preliminary report. In *Working Notes of the AAAI Fall Symposium on Rational Agency: Concepts, Theories, Models and Applications*, Cambridge, MA, 1995.

[32] E. Ternovskaia. Interval situation calculus. In *Proc. of ECAI'94 Workshop W5 on Logic and Change*, pages 153–164, Amsterdam, August 8-12, 1994.

[33] R. Waldinger. Achieving several goals simultaneously. In E. Elcock and D. Michie, editors, *Machine Intelligence 8*, pages 94–136. Ellis Horwood, Edinburgh, Scotland, 1977.