Causality, Indirect Effects and Triggers
Preliminary Report

Javier A. Pinto*
Departamento de Ciencia de la Computación, Escuela de Ingeniería, Pontificia
Universidad Católica de Chile, Casilla 306, Santiago, CHILE
jpinto@acm.org

Abstract

Causality plays an essential role in reasoning about actions, and much of the research on the area deals with the problem of formalizing causality. In this article we pursue the idea that causality can be understood as an abstraction, and that one can write theories in which causality can be left implicit in the formalization of dynamic worlds. In particular, we identify one form of causality, which we call *triggered causality*, and illustrate it with a simple standard example taken from the literature on causation. Later, we relate our approach to causality to the one proposed by Lin in his recent research. We establish the proposition that, for a restricted class of theories based on explicit causation, it is possible to find a translation of a causal theory into a theory with embedded causation and triggers.

Introduction.

In a recent article (Pinto 1998c) we proposed to treat causation as an abstraction that can be extracted from theories of action as opposed to treat it as a primitive notion from which theories of action should be built. Furthermore, we postulated that causation is an abstraction that hides a complex or unknown reality. For instance, the statement "cigarette smoking may cause cancer," should not be taken to mean that the act of smoking sometimes has cancer as a direct consequence. Rather, it states that, by some complex process, that we do not care, or are not prepared to elaborate, cigarette smoking leads to a state in which cancer may develop.

Research in Knowledge Representation has led to the discovery of several problems that are inherent to the enterprise of building theories of action. In (Pinto 1998c) we discussed the modeling of indeterminate actions and the ramification problem. In this article, we elaborate on our discussion of the *ramification problem* (Ginsberg & Smith 1988; Lin & Reiter 1994; Kartha & Lifschitz 1994; Thielscher 1997); i.e., the problem that arises when one needs to reason about indirect effects of actions. For example, flipping a switch might turn a light off, making the room dark. However, the indirect effect, like causality, cannot be used in the reverse order. That is, making the room dark should not have the indirect effect of flipping the switch. One approach to deal with this problem is to appeal to *causation*, introduced as a primitive relation between entities (for example, see (Thielscher 1997; Lin 1995; Kartha & Lifschitz 1994)). These entities could be actions/events or fluents (dynamic world properties). For instance, *causes* might be introduced as a predicate between fluents. Then, one would write something like *causes(SwitchOn,LightOn)*, perhaps with certain qualifications, in order to state that whenever the switch is on, the light is also on. If *causes* is treated as asymmetric, then causality can be used to eliminate the possibility that changing the status of the light might change the status of the switch.

The main goal of this article is to compare our approach to an existing approach to modeling causation, as pertains to the ramification problem or the reasoning with indirect effects. In particular, we compare our proposal to Fangzhen Lin's work on causality and indirect effects (Lin 1995). We establish the result that *Causal Theories* (in Lin's style) can be translated, in some cases, into theories with implicit causation and triggers in the style that we proposed in (Pinto 1998c). The relevance of this result is twofold, it lends credibility to our approach based on implicit causation, and it provides a better understanding of the space of possible formalisms for reasoning about actions and causality. There are several approaches to write theories of action and change using explicit causality; for instance, see (Lifschitz 1987; McCain & Turner 1995; Thielscher 1997). Also, Doherty and Gustafsson, in (Doherty 1994; Gustafsson & Doherty 1996), advance an approach based on the concept of *occlusion* and a filtering method, proposed by Sandewall (Sandewall 1994). The use of occlusion in this latter approach turns out to be similar to Lin's use of the *Causes* predicate, as discussed in (Gustafsson & Doherty 1996). In future work, we would like to extend our comparison to other approaches based on explicit causality. As suggested by Lin (personal communication) such a comparison would shed some light on the issue of when one should use implicit or explicit causation.

In this article, we present the modeling approach introduced in (Pinto 1998c), we discuss one simple circuit example, used to illustrate problems that may arise in the presence of state constraints.

*This work has been partly funded by the Fondecyt project number 197-1304, supported by FCT and by the PRAXIS XXI Project 2/2.1/MAT/262/94 SitCalc.

The Modeling Approach

In this article we explore the consequences of viewing causation as an abstract notion. In particular, if one models a particular dynamic reality, and the model fails to give the expected results, the reason might be that the level of abstraction used for the model is too coarse. Instead of introducing explicit causality and minimization to differentiate between *good* and *bad* models of a theory, we change the level of description allowed by the language, and rebuild the theory accordingly. We discuss a simple example used in the research literature on the ramification problem (for example, (Kartha & Lifschitz 1994; Thielscher 1997; Grünwald 1998)). Our treatment of the switches example can also be applied to other classical problems used to illustrate the ramification problem; for instance Lin's *suitcase example* (Lin 1995) and the *stuffy room example* of Ginsberg and Smith's (Ginsberg & Smith 1988).

A common approach to the representation of knowledge about dynamic worlds recognizes actions/events and states to be primitive classes of objects in the world. States can also be associated with metric time. In this article, we work in the Situation Calculus tradition (McCarthy & Hayes 1969). In particular, we work with an extended Situation Calculus that incorporates several essential notions, like *action occurrence* and *concurrency*. Theories of action are written in the Situation Calculus taking the following notion of a *situation*:

> A situation is a sequence of actions, along with an initial situation (S_0). The state of the world in any given situation is described with a set of fluents (properties), whose truth values do not change within a situation. These fluents describe the state of the world after performing the sequence of actions that lead to the situation, starting in S_0.
>
> The fluents may be called static, because their truth values do not change within a situation. However, fluents may also describe a dynamic situation. For example, a fluent could represent the assertion "the ball is falling due to the gravitational attraction to earth." These assertions can hold throughout the state of a situation.

At the heart of the situation calculus ontology is the notion of action. The world changes from one situation to another as the result of executing actions. In particular, we consider that:

> Actions are instantaneous. Actions with durations can be modeled as pairs of instantaneous actions; one action starts the action with duration, and the other ends it. This approach to represent actions with durations is discussed in (Baier & Pinto 1998).
>
> Actions can be initiated by agents or they can be *natural*. Natural actions are *triggered* by nature. Thus, they occur when the state of the world is such that the actions must occur.

As discussed later in the article, we need to extend the ontology of actions to allow for concurrency.

Given that the language of the extended Situation Calculus is used to write theories of action, it is natural to ask whether *causation* appears in the Situation Calculus language. From our point of view, a form of causation, that we we call *elementary causation*, is implicit. Elementary causation is:

- A relation between an action and fluents. This elementary causation is specified with *effect axioms*.
- A relation between fluents, implicitly expressed in *ramification state constraints*. The ramification state constraints are *logical* statements that, along with some original set of effect axioms, imply some derived effects. For instance, switching the light off, has as a derived effect to make the room dark. The state constraint would state that the room is dark if and only if the light is off.
- A relation between the state of the world and actions (or events). For example, if the state of the world is such that the pressure inside a container exceeds the pressure capacity of the container, then the container explodes. In this case there is a causal relationship between the *high pressure* condition and an action occurrence. This type of elementary causation is modeled with natural actions, along with the constraint that natural actions occur whenever they are possible. We refer to this type of causation as *triggered causation*

We regard other forms of causation as abstractions (or compositions) of elementary causation. Thus, one way to interpret the assertion *fluent f causes fluent g to be true*, is to state that there is a process that is started when f is true, and this process leads to a state where g is true.

The Light and Switches Example

The light and switches example (see figure 1) belongs to a class of examples used to describe a form of the ramification problem. The circuit is modeled with a propositional language. The language includes propositional symbols Sw_1, Sw_2 (the propositions are true if the corresponding switches are connected) and *Light* (which is true if the light is on).

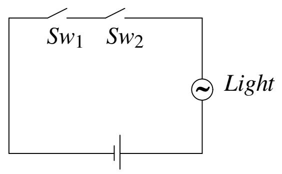

Figure 1: The Light and Switches Example

The state of the system is modeled with a truth assignment to the propositional letters. Naturally, it seems, some constraints need to be placed on the truth values of propositions. In particular, we need to state:

$$Sw_1 \wedge Sw_2 \equiv Light \tag{1}$$

Therefore, any truth assignment to the propositional symbols that is consistent with (1) can describe the state of the circuit. One problem arises when one starts with the state[1]:

$$\{\neg Sw_1, Sw_2, \neg Light\}$$

and Sw_1 is toggled. The only immediate effect of toggling the switch is that Sw_1 becomes true in the new state. However, there is an indirect effect of the toggle action. This effect is that the light comes on. Thus, the resulting state one wants is:

$$\{Sw_1, Sw_2, Light\} \tag{2}$$

Due to the symmetry of the formalization, if one uses a minimal change policy, one would end up with two candidate states, represented by (2) and:

$$\{Sw_1, \neg Sw_2, \neg Light\} \tag{3}$$

Both states have minimal change with respect to the starting state. Clearly, if one understands the behavior of electrical circuits, the state represented by (3) is not wanted. Quite obviously, the problem arises because the theory lacks certain essential information. What is missing?

One answer to the question is that we need some way to state that flipping one switch cannot affect the other switch, but that it affects the light. To incorporate this information, one can appeal to *causation*. Thus, one can say that Sw_1 **causes** *Light* **if** Sw_2. This may be taken to mean that if the switch Sw_1 is connected, then it would cause *Light* provided that Sw_2 is also connected. Similarly, (flipping switches) one would have an analogous causal rule for Sw_2 causing *Light* if Sw_1 is on. This causal information, if properly used, would allow us to compute yet another state (Thielscher 1997). After a stable state is reached, the state of the world should be one in which the light is on.

However, there are other ways to deal with this problem. First, observe that the constraint (1) is not really faithful to the reality we are modeling. Rather, it is a simplification. Light bulbs take time to heat up and emit light. Therefore, there will be states of the world where two switches and a light, connected as depicted by figure 1, contradict constraint (1). The key here, is that the passage of time introduces an aspect not accounted for in the propositional model. This is the notion of *state*. After toggling the switch, the circuit changes state to a new state (perhaps unstable). In the new state the light might be off and the two switches might be connected. In these circumstances, the light comes on and a new state is generated.

Therefore, we take the causal relations between the switches and the light to be an abstraction. However, instead of explicitly introducing this abstraction to overcome shortcomings of our models, we can improve the modeling language and decrease the granularity. If a model fails to capture reality, we can attempt to decrease the level of abstraction and be more faithful to the reality we want to capture.

In the next section we propose a different way to model the circuit example, were we show that the type of ramification where causation seemed necessary can be modeled by introducing intervening, perhaps unstable, states. In a later section, we show how this same problem is solved using Lin's methodology (Lin 1995).

[1] This means an assignment where *false* is assigned to the propositional symbols Sw_1 and *Light*, and *true* is assigned to Sw_2.

A Detailed View of the Light Switches Example

In this section we introduce a new formalization of the light switches examples. We make use of the concurrent Situation Calculus (Pinto 1994; Reiter 1996), although one could use other specification languages instead (e.g., the Circumscriptive Event Calculus(Shanahan 1995)). Concurrency is not necessary to model this example, but it is necessary in a general setting, as evidenced in a later section, where we compare Lin's causal approach to the one described here.

We assume some familiarity with the Situation Calculus. The concurrent Situation Calculus is a sorted language with sorts $\mathcal{A}, \mathcal{C}, \mathcal{S}, \mathcal{D}$ for simple actions, concurrent actions, situations and domain objects. The sort of the variables used in the examples should be obvious from the context. Furthermore there is a distinguished initial situation denoted with the constant S_0, and a function *do* that, given an action and a situation, yields another situation. Fluents are predicates that take one argument (its last) of type situation. To model a dynamic domain we describe how the world changes from one situation to another as the result of performing some concurrent action.

The foundational axioms for the concurrent Situation Calculus are[2]:

$$(\forall \varphi).[\varphi(S_0) \wedge (\forall s, c)\ (\varphi(s) \supset \varphi(do(c, s)))] \supset (\forall s)\ \varphi(s), \tag{4}$$

$$do(c_1, s_1) = do(c_2, s_2) \supset c_1 = c_2, \tag{5}$$

$$\neg s < S_0, \tag{6}$$

$$s < do(c, s') \equiv Poss(c, s') \wedge s \leq s', \tag{7}$$

$$Poss(c, s) \equiv (\forall a)[a \in c \supset Poss(a, s)]. \tag{8}$$

Axioms (4) and (5) characterize the space of situations, and axioms (6) and (7) define the relation $<$ between situations. Conceptually, $S < S'$ is true whenever there is a sequence of concurrent actions that, when performed in S, lead to S'. Furthermore, all the actions performed must be possible in the situations in which they are performed.

Concurrent actions are treated as sets of primitive actions (as in (Reiter 1996)). Thus, $\in$ is used as a relation between simple actions and concurrent actions. We write $a \in c$ to mean that action a is a part of the concurrent action c. Axiom (8) embodies the assumption that actions can be performed concurrently if and only if all the component actions can be performed individually. In general, this is not correct, since there are actions that can be performed concurrently, but not individually. A more general treatment of concurrency falls outside the scope of this article.

In addition to the axioms (4)-(8) we make use of the notion of *natural action*. Natural actions are not regular *agent actions*. Instead, they are actions that arise due to the nature

[2] All free variables that appear in formulas are assumed to be universally quantified with maximum scope.

of the world. Thus, these actions are *naturally triggered* by the state of the world. One characteristic of these actions is that they must occur whenever they are possible. Thus, we write:

$$Poss(a,s) \wedge natural(a) \supset occurs(a,s). \tag{9}$$

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

$$\begin{aligned} legal(s) \equiv\ & s \geq S_0 \wedge \\ & (\forall s', c, a)[s' < s \wedge occurs(a,s') \wedge \\ & \qquad do(c,s') \leq s \supset a \in c]. \end{aligned} \tag{10}$$

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

To model the circuit of figure 1, we introduce constants Sw_1 and Sw_2. We assume that these constants denote distinct individuals. Thus:

$$Sw_1 \neq Sw_2. \tag{11}$$

Also, we use the fluent *connected* to describe the state of a switch. Thus $connected(Sw, s)$ would be true of a situation in which the switch were connected. Similarly, $Light(s)$ is meant to state that in a given situation the light is on. We use the fluent $flowing$, which is true of situations where there is current flowing in the circuit. We also add new actions *Start_flow* and *End_flow*. These are the natural actions in this domain:

$$natural(a) \equiv a = Start\_flow \vee a = End\_flow. \tag{12}$$

The basic toggle action is modeled with the function *toggle* that takes one argument (the switch to be toggled).

Below, we specify the sets of axioms structured according to the style of axiomatization proposed by Reiter (Reiter 1991).

Unique Names Axioms Uniqueness of names is assumed for actions. In the example:

$$Start\_flow \neq End\_flow, \tag{13}$$
$$toggle(x) \neq Start\_flow \wedge toggle(x) \neq End\_flow, \tag{14}$$
$$toggle(x) = toggle(y) \supset x = y. \tag{15}$$

Action precondition axioms The action precondition axioms are necessary and sufficient conditions for actions to be executable. We have:

$$Poss(toggle(x), s) \equiv x = Sw_1 \vee x = Sw_2, \tag{16}$$
$$\begin{aligned} Poss(Start\_flow, s) \equiv\ & connected(Sw_1, s) \wedge \\ & connected(Sw_2, s) \wedge \neg flowing(s), \end{aligned} \tag{17}$$
$$\begin{aligned} Poss(End\_flow, s) \equiv\ & \neg(connected(Sw_1, s) \wedge \\ & connected(Sw_2, s)) \wedge flowing(s). \end{aligned} \tag{18}$$

Axiom (16) states that only the two switches can be toggled. Axioms (17) and (18) state the circumstances under which the flow of current may start and may end.

Effect axioms To describe the effect of toggling a switch, we add the *effect axioms*:

$$\begin{aligned} Poss(c,s) \wedge toggle(x) \in c \wedge connected(x,s) \supset\ & \\ \neg connected(x, do(c,s)), & \end{aligned} \tag{19}$$
$$\begin{aligned} Poss(c,s) \wedge toggle(x) \in c \wedge \neg connected(x,s) \supset\ & \\ connected(x, do(c,s)). & \end{aligned} \tag{20}$$

The effects of End_flow and $Start\_flow$ are described as:

$$Poss(c,s) \wedge End\_flow \in c \supset \neg flowing(do(c,s)), \tag{21}$$
$$Poss(c,s) \wedge Start\_flow \in c \supset flowing(do(c,s)). \tag{22}$$

State constraints The state constraint:

$$Light(s) \equiv flowing(s). \tag{23}$$

states that the light is on if and only if there is current flowing. This state constraint is a ramification state constraint. Thus, it establishes a truth condition of all states. When an action affects the $flowing$ or the $Light$ fluents, the action also has, as ramification, an effect on the other fluent.

This state constraint does not accurately model the behavior of the light. Assuming that the light works properly, it is not generally true that the flowing of current instantaneously turns the light on. Instead, the light bulb will take some time to heat up and emit light. This behavior can also be modeled with the introduction of a natural event, that results from a continuous process (Pinto 1994; Kelley 1996; Reiter 1996). Therefore, state constraint (23) is a simplification, which, for the purposes of the example, yields the expected results.

Following (Pinto 1994), we can derive new effect axioms from the state constraint (23) and the effect axioms (21) and (22):

$$Poss(c,s) \wedge End\_flow \in c \supset \neg Light(do(c,s)), \tag{24}$$
$$Poss(c,s) \wedge Start\_flow \in c \supset Light(do(c,s)). \tag{25}$$

Successor state axioms The successor state axioms embody completeness assumptions on the domain description. These axioms are derived from all the effect axioms for the domain. Therefore, for this example, we replace axioms (19)-(25) with:

$$Poss(c,s) \supset Light(do(c,s)) \equiv [Start\_flow \in c \vee Light(s) \wedge End\_flow \notin c], \tag{26}$$

$$Poss(c,s) \supset flowing(do(c,s)) \equiv [Start\_flow \in c \vee flowing(s) \wedge End\_flow \notin c], \tag{27}$$

$$Poss(c,s) \supset connected(sw,do(c,s)) \equiv [toggle(sw) \in c \wedge \neg connected(sw,s) \vee toggle(sw) \notin c \wedge connected(sw,s)]. \tag{28}$$

The initial situation The only constraints on the possible values assigned to the fluents in the initial situation are the state constraints. Therefore, one should add the state constraints (particularized to S_0) to the theory. We omit this step, and assume that in the initial situation, S_0, the state is one where the second switch is connected, the first switch is disconnected and the light and current flow are off. Thus:

$$\neg connected(Sw_1,S_0) \wedge connected(Sw_2,S_0) \wedge \neg flowing(S_0) \wedge \neg Light(S_0). \tag{29}$$

Now, we have the following proposition:

Proposition 1 *Let Σ denote the theory with axioms (4)-(18) and (26)-(29). Then:*

$$\Sigma \models (\forall c_1,c_2,s).\ legal(s) \wedge s = do(c_2,do(c_1,S_0)) \wedge toggle(Sw_1) \in c_1 \wedge toggle(Sw_2) \notin c_1 \supset Start\_flow \in c_2 \wedge Light(s). \tag{30}$$

This proposition states that, after toggling Sw_1 in S_0, the current starts flowing and the light comes on.

In this example we show that we are able to obtain the results that we expect from the model. That is, the *Light* is on after the circuit becomes stable. To obtain this result, we explicitly incorporated the dynamic nature of the behavior of the circuit using a sequence of states. Thus, unless we explicitly toggle Sw_2, toggling Sw_1 leads to a state where the *Light* is on. We are able to obtain much more than that. For instance, we can derive explanations, if we observe that the *Light* is on some time after S_0, then there must be an intermediate situation where Sw_1 is toggled.

The result can also be stated in another form, by explicitly introducing the notion of stability. We say that a situation is stable if no natural action is possible in that situation. That is, we introduce the predicate $stable \subseteq \mathcal{S}$ with the definition:

$$stable(s) \equiv \neg(\exists a).natural(a) \wedge Poss(a,s). \tag{31}$$

Thus, a stable situation is one in which there are no natural actions that can be triggered.

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

$$leads(c,s,s') \equiv s < do(c,s) \leq s' \wedge stable(s') \wedge ((\forall s_i) s < s_i < s' \supset \neg stable(s_i)) \wedge ((\forall c_i,s_i,a_i)[do(c,s) < do(c_i,s_i) \leq s' \wedge a \in c_i] \supset natural(a)). \tag{32}$$

This latter predicate is intended to relate a situation s and an action a to the *stable* situation that results after all natural actions triggered by a in s are performed. The second conjunct of the definition requires that all intermediate situations be unstable. The last conjunct represents the requirement that the resulting stable situation be reached by performing only natural actions.

A different way of formulating the result that the light is on after toggling the switch is:

$$\Sigma \models (\forall s).leads(\{toggle(Sw_1)\},S_0,s) \supset Light(s). \tag{33}$$

This means that, after the first switch is toggled in the initial situation, the next stable state is one in which the light is on. Notice that this formulation is weaker than the one in (30) because the definition for *leads* requires that all the actions that are performed to reach the final stable state be natural.

Lin's Approach to Causation

Lin's approach is also based on the situation calculus, with some non-trivial differences. For instance, Lin uses a situation calculus with reified fluents. Thus, one needs to introduce a new sort $\mathcal{F}$ for fluents along with a distinguished predicate constant $holds \subseteq \mathcal{F} \times \mathcal{S}$. Thus, instead of writing $connected(x,s)$ one should write $holds(connected(x),s)$. However, as Lin, we write $connected(x,s)$ as shorthand for $holds(connected(x),s)$. The reason for the use of fluents as objects is to allow them to be arguments for the *Causes* predicate, whose extension is assumed to be minimal.

Another difference is that Lin does not have a sort for concurrent actions, and retains the original situation calculus ontology for actions.

Also, the situation calculus is extended with a sort $\mathcal{V}$ for truth values, which includes the objects *True* and *False* (which we use as constants that denote themselves).

The predicate $Caused \subseteq \mathcal{F} \times \mathcal{V} \times \mathcal{S}$ is introduced to mean that a fluent is caused to take a truth value at a situation.

Along with the foundational axioms for the situation calculus (similar to (4)-(7), but trivially modified to refer to non-concurrent actions), the following axioms are necessary (we hope that the sorts for the variables and constants are clear from the context):

$$Caused(p,True,s) \supset Holds(p,s), \tag{34}$$

$$Caused(p,False,s) \supset \neg Holds(p,s), \tag{35}$$

$$True \neq False \wedge (\forall v)(v = True \vee v = False). \tag{36}$$

To model the circuit example in Lin's framework, we write:

Direct Effect Axioms

$$Poss(toggle(x),s) \supset [connected(x,s) \wedge v = False \supset Caused(connected(x),v,do(toggle(x),s))],$$

$$Poss(toggle(x),s) \supset [\neg connected(x,s) \wedge v = True \supset Caused(connected(x),v,do(toggle(x),s))].$$

Explicit Qualifications We assume that the qualification problem is solved, and that it yields the axioms:

$$Poss(toggle(x),s) \equiv x = Sw_1 \vee x = Sw_2,$$
$$Poss(Start\_flow,s) \equiv connected(Sw_1,s) \wedge connected(Sw_2,s) \wedge \neg flowing(s),$$
$$Poss(End\_flow,s) \equiv \neg(connected(Sw_1,s) \wedge connected(Sw_2,s)) \wedge flowing(s).$$

Causal Rules The following rules are used to derive indirect effects:

$$connected(Sw_1,s) \wedge connected(Sw_2,s) \supset Caused(Light,True,s),$$
$$\neg connected(Sw_1,s) \vee \neg connected(Sw_2,s) \supset Caused(Light,False,s),$$

Domain Knowledge

$$Sw_1 \neq Sw_2.$$

Unique Names Axioms Uniqueness of names is assumed for fluent and action names.

Minimization of Caused As pointed out by Lin, the minimization of *Caused* can be computed with Clark's completion when the causal theory (direct effect axioms along with the causal rules) is stratified. In this case, the minimization yields:

$$\begin{aligned}&Caused(connected(x),v,s) \equiv\\ &\quad v = False \wedge (\exists s')[s = do(toggle(x),s') \wedge connected(x,s') \wedge Poss(toggle(x),s')] \vee\\ &\quad v = True \wedge (\exists s')[s = do(toggle(x),s') \wedge \neg connected(x,s') \wedge Poss(toggle(x),s')].\end{aligned}$$

and

$$\begin{aligned}&Caused(Light,v,s) \equiv\\ &\quad [v = True \wedge connected(Sw_1,s) \wedge connected(Sw_2,s)] \vee\\ &\quad [v = False \wedge (\neg connected(Sw_1,s) \vee \neg connected(Sw_2,s))].\end{aligned}$$

Using the *pseudo successor state axiom*:

$$\begin{aligned}&Poss(a,s) \supset\\ &\quad (holds(p,do(a,s)) \equiv\\ &\qquad Caused(p,True,do(a,s)) \vee\\ &\qquad holds(p,s) \wedge \neg Caused(p,False,do(a,s))),\end{aligned} \tag{37}$$

we can derive the successor state axiom for connected:

$$\begin{aligned}&Poss(a,s) \supset connected(x,do(a,s)) \equiv\\ &\quad a = toggle(x) \wedge \neg connected(x,s)\\ &\quad \vee connected(x,s) \wedge a \neq toggle(x).\end{aligned} \tag{38}$$

Analogously, we can obtain:

$$\begin{aligned}&Poss(a,s) \supset Light(do(a,s)) \equiv\\ &\quad connected(Sw_1,do(a,s)) \wedge connected(Sw_2,do(a,s)) \vee\\ &\quad [Light(s) \wedge connected(Sw_1,do(a,s)) \wedge connected(Sw_2,do(a,s))].\end{aligned}$$

which reduces to:

$$Poss(a,s) \supset Light(do(a,s)) \equiv connected(Sw_1,do(a,s)) \wedge connected(Sw_2,do(a,s)).$$

By using regression (Reiter 1991) and the successor state axiom for connected, we obtain the final successor state axiom for *Light*:

$$\begin{aligned}&Poss(a,s) \supset Light(do(a,s)) \equiv\\ &\quad [a = toggle(Sw_1) \wedge \neg connected(Sw_1,s)\\ &\quad \vee connected(Sw_1,s) \wedge a \neq toggle(Sw_1)] \wedge\\ &\quad [a = toggle(Sw_2) \wedge \neg connected(Sw_2,s)\\ &\quad \vee connected(Sw_2,s) \wedge a \neq toggle(Sw_2)].\end{aligned}$$

which can be rewritten as:

$$\begin{aligned}&Poss(a,s) \supset Light(do(a,s)) \equiv\\ &\quad [a = toggle(Sw_1) \wedge \neg connected(Sw_1,s) \wedge connected(Sw_2,s)] \vee\\ &\quad [a = toggle(Sw_2) \wedge \neg connected(Sw_2,s) \wedge connected(Sw_1,s)] \vee\\ &\quad [connected(Sw_1,s) \wedge connected(Sw_2,s) \wedge a \neq toggle(Sw_2) \wedge a \neq toggle(Sw_2)].\end{aligned} \tag{39}$$

Given the successor state axioms (38) and (39), it is straightforward to see that using Lin's method for modeling causation yields the right conclusions.

Comparing Both Approaches to Modeling Causality

Let T be a theory of action in Lin's style, that includes the foundational axioms, axioms (34)-(36), along with a set of direct effect axioms, explicit qualifications, domain knowledge, causal rules and unique names axioms. We assume that the causal rules are stratified. Also, for simplicity, we assume that T includes a complete description of the initial situation (that is, for every fluent there is an explicit axiom stating whether or not the fluent holds in S_0). Furthermore, let $L(T)$ denote the theory that results from extending T with: (1) a circumscription axiom to minimize *Caused* and leaving everything else fixed; (2) successor state axioms for each fluent in the language, obtained by using the *pseudo successor state axiom* (37). We want to define a translation $P(T)$ from Lin's causal theories to our situation calculus theories that incorporate natural actions and triggers.

To perform the translation from T to $P(T)$, we take the direct effect axioms and re-reformulate them in the standard approach (without *Caused*). The explicit qualifications and domain knowledge is retained in the same form[3]. At the heart of the matter is the translation of the causal rules, which, along with the direct effect axioms in T, form a stratified theory. Given that the causal theory is stratified, we have in T causal rules of the form:

$$\Phi(s) \wedge Caused(p_1,V_1,s) \wedge \ldots \wedge Caused(p_n,V_n,s) \supset Caused(f(x),V,s), \tag{40}$$

[3]The reader has to keep in mind the different treatment of fluents as objects in the causal approach, and as predicates in the noncausal approach.

where f is a fluent, and $\Phi(s)$ is a formula simple in s[4]. Also, we assume that the $\mathcal{V}$ terms V_i and V are constants (either *True* or *False*).

For each causal rule r of the form (40), we introduce a new natural action A_r, and an *enabling* fluent F_r, along with the following:

Natural action:

$$natural(A_r). \tag{41}$$

Precondition axiom

$$Poss(A_r, s) \equiv F_r(s). \tag{42}$$

State Constraint

$$\Phi(s) \wedge p'_1(s) \wedge \ldots \wedge p'_n(s) \equiv F_r(s), \tag{43}$$

where the $p'_i(s)$ literals correspond to $p_i(s)$ if the causal rule has the literal $Caused(p_i, True, s)$, and $\neg p_i(s)$ if the causal rule has the literal $Caused(p_i, False, s)$.

Effect Axioms

$$Poss(A_r, s) \wedge A_r \in c \supset \neg F_r(do(c,s)), \tag{44}$$

$$Poss(A_r, s) \wedge A_r \in c \supset f'(do(c,s)). \tag{45}$$

where the $f'(do(c,s))$ literal is either $f(do(c,s))$ or $\neg f(do(c,s))$ depending upon the value of V in (40).

Given these new axioms, we now need to derive successor state axioms for the new fluents F_r. The state constraints of the form (43) correspond to what we called *stratified definitions* (Pinto 1994), where we showed a mechanism, also based on the use of Reiter's regression operator (Reiter 1991), to obtain successor state axioms for the defined fluents. Let $P(T)$ denote the theory that results from replacing the new effect axioms (definitions) with the corresponding successor state. The result that we want is the following:

Proposition 2 *Let T, $L(T)$ and $P(T)$ be as described above. Also, let f be any fluent, and A be an arbitrary action in the language of T. Then:*

$$L(T) \models f(do(A, S_0)), \tag{46}$$

if and only if

$$P(T) \models leads(\{A\}, S_0, s) \wedge f(s) \tag{47}$$

Proof Intuitively, the theory $P(T)$ is built by taking the *causal rules* and replacing them with sets of natural actions. Given that the causal rules in $L(T)$ are stratified, they represent a causal behavior that can be analyzed in stages. In each stage, one takes consequences from one stratum and infer new consequences for the next. The natural actions that are defined to replace these causal rules mimic this deductive process. In the theory $P(T)$, each stratum is related to a situation. The first stratum corresponds to the situation where the direct effects of action A hold. In this stratum some natural actions might be possible. If so, the natural actions occur and lead to the next situation, which represents another stratum. Thus, the set of unstable situations that are the result of perform action A correspond to the strata of the causal theory.

More formally, we first need to define a minimal stratification. A stratification is an assignment of natural numbers (representing the strata) to the fluents of the language such that:

- The fluents that appear as direct effects of actions belong to the first stratum, number 0.
- If a fluent appears on the left side of a causal rule, then its stratum must be less than the stratum of the fluent in the right hand side of the rule.
- The strata assigned to each fluent, must be minimal.

Also, we define the size of a stratification as the number that corresponds to the highest stratification. The highest stratum [5], is called final.

Given this minimal stratification, the proof is done by induction on the stratum level. We prove by induction that for each non-final level of stratification in $L(T)$, there is a corresponding unstable situation in $P(T)$ where the natural actions that are possible lead to a successor situation associated to the next stratum. Furthermore, at each level, the changes that are provoked to the next situation carry over to the situation representing the final level. The last statement is obviously correct given that the causal rules are stratified (i.e., no fluent is changed twice in the stratification).

The basis step corresponds to level 0, represented by $do(\{A\}, S_0)$ in P(T), in that level, all changes at the first stratum are changed in the same manner as in $L(T)$. It is trivial to see that for each applicable causal rule in $L(T)$ there will be a natural action that is triggered in $P(T)$. The actions so triggered lead to the next situation, representing stratum 1. For the induction step, we assume that we reached a situation representing the n^{th} stratum, it is immediate that, by the construction of the natural actions, that this level will create the conditions for the situation representing the $n+1$ stratum.

□

The formalization that we used for the light and switches example, does not correspond to the translation mechanism described here. However, the translation results in a very similar theory. The theory that we used previously was written with the physical reality in mind. The translation, on the other hand, results from a syntactic manipulation. Therefore, the resulting theory might include natural actions which do not necessarily have analogues in reality. We believe that this should in general be the case. Whether this turns out to be true is a question of pragmatics.

Conclusion

In (Pinto 1998c) we proposed to treat causality as an abstraction defined in terms of primitive causation. The level of

[4] I.e., a formula that has s free and that does not mention any other situation term other than s; it does not mention *Caused*, nor *Poss*, nor does it mention any $\mathcal{V}$ terms.

[5] We assume finite axiomatizations.

granularity of the axiomatization is defined by primitive causation, expressed with direct action effects, state constraints, and natural occurrences.

One question that Sandewall raised regarding this approach[6] has to do with the generality of the approach, since it was originally justified in terms of a small set of examples. Here, we have attempted to address this concern, by showing that, for a class of causal theories, we can do as well as the method proposed by Lin. However, this is a rather technical result which does not necessarily shed any light on the more philosophical question: which approach is more natural? When we write theories of action (for examples, see (Pinto 1998a)), we attempt to describe reality at a level of abstraction that would allow an automated reasoner to reach the correct answers for given reasoning tasks. For our description of a dynamic reality, we define the objects of discourse (fluents, situations, actions and domain objects) and relations between them, perhaps at different moments in time (or different situations).

One essential aspect of our approach to writing theories of action is the use of *natural* actions, which can be seen as defining dynamic processes that follow a natural course that might be disturbed by agent's actions. The need for integrating these processes into theories of action has arisen in other frameworks as well (e.g., (Thielscher 1997; Lin 1998)).

In our future research, we would like to extend this work (which we consider preliminary) by completing the comparison between our framework and Lin's framework, which has been extended in (Lin 1998). We are also interested in establishing comparisons with other approaches already mentioned in the article.

Acknowledgements

I would like to thank many people that contributed to the ideas presented in this article. Among them, I would like to give explicit thanks to Leo Bertossi for his support and guidance. Also, thanks to one of the anonymous reviewers for a very careful review, and for pointing out shortcomings of the original submission. Any shortcomings left in the article are, of course, my own fault.

References

Baier, J., and Pinto, J. 1998. Non-instantaneous Actions and Concurrency in the Situation Calculus. To be presented at the ESSLLI-98 Workshop on Reasoning about Actions: Foundations and Applications.

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

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

Grünwald, P. 1998. Ramifications and Sufficient Causes. In Miller, R., and Shanahan, M., eds., *Common-Sense '98*, 42–55. Queen Mary and Westfield College, University of London. Also found at URL = http://www.ida.liu.se/ext/etai/nj/fcs-98/042/paper.ps.

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

Kartha, G. N., and Lifschitz, V. 1994. Actions with Indirect Effects (Preliminary Report). In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR'94)*.

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

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

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

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

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

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

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

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

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

Pinto, J. 1994. *Temporal Reasoning in the Situation Calculus.* Ph.D. Dissertation, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada. URL = ftp://ftp.cs.toronto.edu/~cogrob/jpThesis.ps.Z.

Pinto, J. 1998a. Integrating Discrete and Continuous Change in a Logical Framework. *Computational Intelligence* 14(1).

[6] See URL = http://www.ida.liu.se/ext/etai/nj/fcs-98/349/aip.html#001.

Pinto, J. 1998b. Occurrences and Narratives as Constraints in the Branching Structure of the Situation Calculus. *Journal of Logic and Computation*. To appear.

Pinto, J. A. 1998c. Causality in Theories of Action. In Miller, R., and Shanahan, M., eds., *Common-Sense '98*, 349–364. Queen Mary and Westfield College, University of London. Also found at `URL = http://www.ida.-liu.se/ext/etai/nj/fcs-98/349/paper.ps`.

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

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

Sandewall, E. 1994. Features and fluents, a systematic approach to the representation of knowledge about dynamical systems. Technical report, Department of Computer and Information Science, University of Linköping, Sweden. To be published as a book (Oxford University Press).

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

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