Embracing Causality in Specifying the Indeterminate Effects of Actions

Fangzhen Lin
Department of Computer Science
University of Toronto
Toronto, Canada M5S 3H5
email: fl@ai.toronto.edu

Abstract

This paper makes the following two contributions to formal theories of actions: Showing that a causal minimization framework can be used effectively to specify the effects of indeterminate actions; and showing that for certain classes of such actions, regression, an effective computational mechanism, can be used to reason about them.

Introduction

Much recent work on theories of actions has concentrated on primitive, determinate actions. In this paper, we pose ourselves the problem of specifying *directly* the effects of indeterminate actions,[1] like we do for the primitive, determinate ones.

There are several reasons why we think this is an important problem. First of all, there are actions whose effects, when described at a natural level, are indeterminate. Secondly, one can argue that there is no absolute defining line between determinate and indeterminate actions. The differences have a lot to do with the levels of descriptions. The effects of an action may be determinate at one level of description, but indeterminate at another. So a theory that treats determinate and indeterminate actions in fundamentally different ways will have difficulties coping with language changes. Finally, even if all the primitive actions have determinate effects, there are still needs for specifying directly the effects of complex actions that are often indeterminate. For instance, these specifications may be part of the inputs to a program synthesizer.

Our contributions in this paper are two folds. We first show that the causal minimization framework of (Lin [5]) can be used effectively to specify the effects of indeterminate actions. We then show that for certain classes of such actions, regression, an effective computational mechanism, can be used to reason about them.

[1] For the purpose of this paper, the phrases "the effects of indterminate actions" and "the indeterminate effects of actions" are considered to be synonyms.

Logical Preliminaries

We shall investigate the problem in the framework of the situation calculus [8]. Our version of it employs a many sorted second-order language. We assume the following sorts: *situation* for situations, *action* for actions, *fluent* for propositional fluents, *truth-value* for truth values $true$ and $false$, and *object* for everything else.

We use the following domain independent predicates and functions:

- The binary function do - for any action a and any situation s, $do(a, s)$ is the situation resulting from performing a in s.
- The binary predicate H - for any propositional fluent p and any situation s, $H(p, s)$ is true if p holds in s.
- The binary predicate $Poss$ - for any action a and any situation s, $Poss(a, s)$ is true if a is possible (executable) in s.
- The ternary predicate $Caused$ - for any fluent p, any truth value v, and any situation s, $Caused(p, v, s)$ is true if the fluent p is caused to have the truth value v in the situation s. For instance, $Caused(loaded, true, do(load, s))$ means that the action $load$ causes $loaded$ to be true in the resulting situation.

We shall make use some additional special predicates and functions, and will introduce them when they are needed.

We assume that all theories in this paper will include the following basic axioms:

- For the predicate $Caused$, the following two basic axioms:[2]

$$Caused(p, true, s) \supset H(p, s), \tag{1}$$
$$Caused(p, false, s) \supset \neg H(p, s). \tag{2}$$

- For the truth values, the following unique names and domain closure axiom:

$$true \neq false \wedge (\forall v)(v = true \vee v = false). \tag{3}$$

[2] We use the convention that in displayed formulas, free variables are implicitly universally quantified from the outside.

- The unique names assumptions for fluent and action names (we assume there are only finitely many of them). Specifically, if $\{F_1, ..., F_n\}$ is the set of the fluent names, then we have:

$$F_i(\vec{x}) \neq F_j(\vec{y}), \ i \text{ and } j \text{ are different,}$$
$$F_i(\vec{x}) = F_i(\vec{y}) \supset \vec{x} = \vec{y}.$$

  Similarly for action names. In the following, we shall denote this set of unique names axioms by $\mathcal{D}_{una}$.

- The set Σ of *foundational axioms* in [6] for the *discrete* situation calculus. These axioms characterize the structure of the space of situations. For the purpose of this paper, it is enough to mention that they include the following unique names axioms for situations:

$$s \neq do(a, s),$$
$$do(a, s) = do(a', s') \supset (a = a' \wedge s = s').$$

In the rest of this paper, we shall frequently make use of the following shorthand notation: If F is a fluent name of arity $object^n \rightarrow fluent$, then we define the expression $F(t_1, ..., t_n, t_s)$ to be a shorthand for the formula $H(F(t_1, ..., t_n), t_s)$, where $t_1, \cdots, t_n$ are terms of sort *object*, and t_s a term of sort *situation*. So if *white* is a fluent, then $white(s)$ is a shorthand for $H(white, s)$.

Minimizing Causation

The approach of (Lin [5]) to specifying the effects of actions can be summarized as follows:

1. Formalize the causal laws and constraints of the domain by a set T of axioms.
2. Circumscribe (minimize) *Caused* in $T \cup \Sigma \cup \mathcal{D}_{una} \cup \{1, 2, 3\}$ with all other predicates fixed.
3. The resulting theory, T', together with the following generic frame axiom: Unless caused otherwise, a fluent's truth value will persist:

$$Poss(a, s) \supset \{\neg(\exists v) Caused(p, v, do(a, s)) \supset [H(p, do(a, s)) \equiv H(p, s)]\}, \tag{4}$$

   will generate the appropriate frame axioms.

Lin [5] also discusses how the action preconditions can be generated. However, in this paper we shall not concern us with this problem, but assume, following (Reiter [10]), that for each action $A(\vec{x})$, we are given an action precondition axiom of the form:

$$Poss(A(\vec{x}), s) \equiv \Psi_A(\vec{x}, s),$$

where Ψ_A is a formula that does not quantify over situation variables, and does not mention any situation dependent atomic formulas except those of the form $H(t, s)$, where t is a propositional fluent term.

We shall be using the following lemma for computing the circumscription of *Caused*:

Lemma 1 *Let* $W = T \cup \Sigma \cup \mathcal{D}_{una} \cup \{1, 2, 3\}$. *Then* $Circum(W, Caused)$, *the result of circumscribing Caused in W with all other predicates fixed, is equivalent to*

$$\{Circum(T, Caused)\} \cup \Sigma \cup \mathcal{D}_{una} \cup \{1, 2, 3\}.$$

Proof: This is because the predicate *Caused* occurs only negatively in $\Sigma \cup \mathcal{D}_{una} \cup \{1, 2, 3\}$. ∎

To illustrate how to use this approach to specify the effects of indeterminate actions, consider Reiter's example of dropping a pin on a checkerboard: The pin may land inside a white square, inside a black square, or touching both.

We introduce three fluents: *white* (all or part of the pin is in a white square), *black* (all or part of the pin is in a black square), and *holding* (the agent is holding the pin); and two actions: *drop* (the agent drops the pin on the checkerboard), and *pickup* (the agent picks up the pin). We have the following action precondition axioms:[3]

$$Poss(drop, s) \equiv holding(s) \wedge \neg white(s) \wedge \neg black(s),$$
$$Poss(pickup, s) \equiv \neg holding(s) \wedge (white(s) \vee black(s)).$$

We have the following effect axioms:

$$Poss(pickup, s) \supset Caused(holding, true, do(pickup, s)), \tag{5}$$
$$Poss(pickup, s) \supset Caused(white, false, do(pickup, s)), \tag{6}$$
$$Poss(pickup, s) \supset Caused(black, false, do(pickup, s)), \tag{7}$$

$$Poss(drop, s) \supset Caused(holding, false, do(drop, s)), \tag{8}$$
$$\begin{aligned} Poss(drop, s) \supset\ & [Caused(white, true, do(drop, s)) \wedge Caused(black, false, do(drop, s))] \vee \\ & [Caused(white, false, do(drop, s)) \wedge Caused(black, true, do(drop, s))] \vee \\ & [Caused(white, true, do(drop, s)) \wedge Caused(black, true, do(drop, s))]. \end{aligned} \tag{9}$$

Suppose these are the only effect axioms, and there are no causal rules and state constraints.[4] By Lemma 1, it is easy to see that circumscribing *Caused* in

$$\{(5) - (9)\} \cup \Sigma \cup \mathcal{D}_{una} \cup \{1, 2, 3\}$$

yields:

$$Poss(a, s) \wedge Caused(p, v, do(a, s)) \supset$$

[3] Recall that as we have defined in Section , $holding(s)$, for instance, is a shorthand for $H(holding, s)$.

[4] Notice that the state constraint $(\forall s)\neg[holding(s) \wedge (white(s) \vee black(s))]$ has been built into the action effect and precondition axioms.

$$
\begin{aligned}
a = pickup \wedge [&(p = holding \wedge v = true) \vee \\
&(p = white \wedge v = false) \vee \\
&(p = black \wedge v = false)] \vee \\
a = drop \wedge [&(p = holding \wedge v = false) \vee \\
&p = white \vee p = black].
\end{aligned}
$$

From this and the generic frame axiom (4), we can deduce the following *successor state axiom* (Reiter [10]) for *holding*:

$$
\begin{aligned}
&Poss(a, s) \supset holding(do(a, s)) \equiv \\
&\quad a = pickup \vee (holding(s) \wedge a \neq drop).
\end{aligned}
$$

We don't get successor state axioms for *white* and *black*. But we have the following explanation closure axioms:

$$
\begin{aligned}
&Poss(a, s) \wedge \neg[white(s) \equiv white(do(a, s))] \supset \\
&\quad (a = pickup \vee a = drop), \\
&Poss(a, s) \wedge \neg[black(s) \equiv black(do(a, s))] \supset \\
&\quad (a = pickup \vee a = drop).
\end{aligned}
$$

These axioms, together with the effect axioms, yield the following disjunction of successor state axioms:

$$
\begin{aligned}
&Poss(a, s) \supset \\
&\{[white(do(a, s)) \equiv \\
&\quad (a = drop \vee (white(s) \wedge a \neq pickup))] \wedge \\
&\ [black(do(a, s)) \equiv \\
&\quad (black(s) \wedge a \neq pickup \wedge a \neq drop))]\} \vee \\
&\{[white(do(a, s)) \equiv \\
&\quad (white(s) \wedge a \neq pickup \wedge a \neq drop))] \wedge \\
&\ [black(do(a, s)) \equiv \\
&\quad (a = drop \vee (black(s) \wedge a \neq pickup))]\} \vee \\
&\{[white(do(a, s)) \equiv \\
&\quad (a = drop \vee (white(s) \wedge a \neq pickup))] \wedge \\
&\ [black(do(a, s)) \equiv \\
&\quad (a = drop \vee (black(s) \wedge a \neq pickup))]\}.
\end{aligned}
$$

Notice the correspondences between the three cases and those in *drop*'s effect axiom for *white* and *black*.

Classes of Indeterminate Actions

The indeterminate effects of *drop* are *inclusive* in that the pin may land on a white square, a black square, or both. To see how such inclusive indeterminate effects can be represented succinctly, notice first that under the two general axioms (1) and (2) about *Caused*, the effect axiom (9) is equivalent to the following three axioms:

$$
\begin{aligned}
&Poss(drop, s) \supset \{Caused(white, true, do(drop, s)) \vee \\
&\quad Caused(black, true, do(drop, s))\}, \\
&Poss(drop, s) \supset \{Caused(white, true, do(drop, s)) \vee \\
&\quad Caused(white, false, do(drop, s))\}, \\
&Poss(drop, s) \supset \{Caused(black, true, do(drop, s)) \vee \\
&\quad Caused(black, false, do(drop, s))\}.
\end{aligned}
$$

Notice that under the domain closure and unique names axiom (3) for truth values, the last axiom is equivalent to

$$
Poss(drop, s) \supset (\exists v) Caused(black, v, do(drop, s)).
$$

This axiom is like the *releases* propositions in the action description language of [3]. Notice here the necessity of something like the predicate *Caused*. The corresponding sentence in terms of H:

$$
\begin{aligned}
&Poss(drop, s) \supset \{H(black, do(drop, s)) \vee \\
&\quad \neg H(black, do(drop, s))\}
\end{aligned}
$$

is just a tautology.

In general, if the action α has inclusive indeterminate effects on the fluent terms $P_1, ..., P_n$, i.e. causes at least one of them to be true and the rest of them to be false, under the context γ, then we have the following causal laws:

$$
\begin{aligned}
&Poss(\alpha, s) \wedge \gamma \supset \{Caused(P_1, true, do(\alpha, s)) \vee \cdots \vee \\
&\quad Caused(P_n, true, do(\alpha, s))\}, \\
&Poss(\alpha, s) \wedge \gamma \supset \{Caused(P_i, true, do(\alpha, s)) \vee \\
&\quad Caused(P_i, false, do(\alpha, s))\},
\end{aligned}
$$

where $1 \leq i \leq n$.

The number of indeterminate effects need not be finite. If, under the context γ, the action α has the inclusive indeterminate effects on $F(x)$ for those x that satisfies φ, then we have the following causal laws:

$$
\begin{aligned}
&Poss(\alpha, s) \wedge \gamma \wedge (\exists x)\varphi(x) \supset \\
&\quad (\exists x)[\varphi(x) \wedge caused(F(x), true, do(\alpha, s))], \\
&Poss(\alpha, s) \wedge \gamma \supset (\forall x)\{\varphi(x) \supset \\
&\quad [Caused(F(x), true, do(\alpha, s)) \vee \\
&\quad\ Caused(F(x), false, do(\alpha, s))]\}.
\end{aligned}
$$

For instance, playing loud rock and roll music will make some of the nearby people (including the person who plays it) happy, and the rest of them unhappy: let γ be true, $\varphi(x)$ be $nearby(x, s)$, and $F(x)$ be $happy(x)$.

Contrast to the inclusive indeterminate effects, we have the exclusive ones. For instance, flipping a coin causes exactly one of $\{head, tail\}$ to be true. Generally, if the action α has *exclusive* indeterminate effects on the fluent terms $P_1, ..., P_n$, i.e. causes exactly one of them to be true and the rest of them to be false, under the context γ, then we have the following causal laws:

$$
\begin{aligned}
&Poss(\alpha, s) \wedge \gamma \supset \{Caused(P_1, true, do(\alpha, s)) \dot{\vee} \cdots \dot{\vee} \\
&\quad Caused(P_n, true, do(\alpha, s))\}, \\
&Poss(\alpha, s) \wedge \gamma \supset \{Caused(P_i, true, do(\alpha, s)) \vee \\
&\quad Caused(P_i, false, do(\alpha, s))\},
\end{aligned}
$$

where $1 \leq i \leq n$, and $\dot{\vee}$ is the exclusive or operator:

$$
\varphi_1 \dot{\vee} \cdots \dot{\vee} \varphi_k \equiv (\varphi_1 \vee \cdots \vee \varphi_k) \wedge \bigwedge_{1 \leq i \neq j \leq k} \neg(\varphi_i \wedge \varphi_j).
$$

Again, the number of indeterminate effects need not be finite. If, under the context γ, the action α has the exclusive indeterminate effects on $F(x)$ for those x that satisfies φ, then we have the following causal laws:

$$
\begin{aligned}
&Poss(\alpha,s) \wedge \gamma \wedge (\exists x)\varphi(x) \supset\\
&\quad (\exists! x)[\varphi(x) \wedge caused(F(x), true, do(\alpha,s))],\\
&Poss(\alpha,s) \wedge \gamma \supset\\
&\quad (\forall x)\{\varphi(x) \supset [Caused(F(x), true, do(\alpha,s)) \vee\\
&\qquad\qquad Caused(F(x), false, do(\alpha,s))]\},
\end{aligned}
$$

where $(\exists! x)$ means there is a unique x. For instance, picking a ball from a bag causes one to hold one and only one of the balls in the bag: let γ be $true$, $\varphi(x)$ be $in\text{-}bag(x,s)$, and $F(x)$ be $holding(x)$.

There are, of course, actions with indeterminate effects that are neither inclusive or exclusive. In general, if the number of the indeterminate effects of an action $A(\vec{x})$ is finite, then its effect axioms can be written of the following forms:

$$
\begin{aligned}
&Poss(A(\vec{x}),s) \supset\\
&\quad (\forall p,s)[\varphi(\vec{x},p,v,s) \supset Caused(p,v,do(A(\vec{x}),s))],
\end{aligned}
\tag{10}
$$

$$
\begin{aligned}
&Poss(A(\vec{x}),s) \supset\\
&\quad \{(\forall p,v)[\varphi_1(\vec{x},p,v,s) \supset Caused(p,v,do(A(\vec{x}),s))]\\
&\quad \vee \cdots \vee\\
&\quad (\forall p,v)[\varphi_n(\vec{x},p,v,s) \supset Caused(p,v,do(A(\vec{x}),s))]\},
\end{aligned}
\tag{11}
$$

where φ and φ_i's are formulas that do not quantify over situation variables, and do not mention any other situation dependent atomic formulas except those of the form $H(t,s)$.

For instance, the two effect axioms about $drop$ can be rewritten as:

$$
\begin{aligned}
&Poss(drop,s) \supset (\forall p,s)\{p = holding \wedge v = false \supset\\
&\qquad Caused(p,v,do(drop,s))\},
\end{aligned}
\tag{12}
$$

$$
\begin{aligned}
&Poss(drop,s) \supset\\
&\quad (\forall p,v)\{[p = white \wedge v = true \vee\\
&\qquad p = black \wedge v = false] \supset\\
&\qquad\quad Caused(p,v,do(drop,s))\} \vee\\
&\quad (\forall p,v)\{[p = white \wedge v = false \vee\\
&\qquad p = black \wedge v = true] \supset\\
&\qquad\quad Caused(p,v,do(drop,s))\} \vee\\
&\quad (\forall p,v)\{[p = white \wedge v = true \vee\\
&\qquad p = black \wedge v = true] \supset\\
&\qquad\quad Caused(p,v,do(drop,s)).
\end{aligned}
\tag{13}
$$

Notice that (10) and (11) can be combined into a single axiom of the latter form. But as we shall see later, it is beneficial to have a separate axiom for determinate effects.

Computing Successor State Axioms

We now consider how to reason with the theories of the actions whose effects are specified by axioms of the forms (10) and (11).

Let T_{ea} be a given set of the effect axioms of the forms (10) and (11). Then the conjunction of the sentences in T_{ea} is *separable* (Lifschtz [4]) w.r.t. $Caused$. Therefore, according to a result in [4], $Circum(T_{ea}, Caused)$, the circumscription of $Caused$ in T_{ea}, is computable by a first-order sentence. In general, this sentence, together with $\mathcal{D}_{una}$, will yield a disjunction of successor state axioms, which is often large and cumbersome to reason with. In particular, it is not clear how to compute *regression*, a computationally effective mechanism for tasks such as planning and temporal projection [11, 9, 10], w.r.t. such disjunctions.

A Transformation

To overcome this, we introduce a new ternary predicate $Case$ of the arity $object \times action \times situation$, and a distinguished constant 0 and a unary function $succ$ over sort *object*. We use the convention that if a natural number n occurs as an object term in a formula, then it is considered to be a shorthand for the term obtained from 0 by applying n times the function $succ$. For instance, in $Case(2,a,s)$, the number 2 is a shorthand for $succ(succ(0))$.

For now we shall consider $Case$ to be an auxiliary predicate introduced for computational purposes. Later, we shall consider some possible interpretations of this predicate.

Using $Case$, we transform the indeterminate effect axiom (11) into the following sentences that have the form of a determinate effect axiom:

$$
\begin{aligned}
&Poss(A(\vec{x}),s) \wedge Case(1,A(\vec{x}),s) \supset\\
&(\forall p,v)[\varphi_1(\vec{x},p,v,s) \supset Caused(p,v,do(A(\vec{x}),s))],
\end{aligned}
\tag{14}
$$

$$\vdots$$

$$
\begin{aligned}
&Poss(A(\vec{x}),s) \wedge Case(n,A(\vec{x}),s) \supset\\
&(\forall p,v)[\varphi_n(\vec{x},p,v,s) \supset Caused(p,v,do(A(\vec{x}),s))],
\end{aligned}
\tag{15}
$$

together with the following constraints on $Case$:

$$Case(1,A(\vec{x}),s)\ \dot{\vee}\ \cdots\ \dot{\vee}\ Case(n,A(\vec{x}),s), \tag{16}$$

$$
\begin{aligned}
&\{(\forall p,v)[\varphi_i(\vec{x},p,v,s) \supset (\varphi(\vec{x},p,v,s) \vee \varphi_j(\vec{x},p,v,s))] \wedge\\
&\neg(\forall p,v)[\varphi_j(\vec{x},p,v,s) \supset (\varphi(\vec{x},p,v,s) \vee \varphi_i(\vec{x},p,v,s))]\}\\
&\supset \neg Case(i,A(\vec{x}),s),
\end{aligned}
\tag{17}
$$

for any $1 \le i \ne j \le n$.

Notice the exclusive or in (16). This is because when $Casued$ is circumscribed, the logical or in (11) will become exclusive. The intuitive meaning of (17) is that if the extension of $(\lambda p,v)\varphi_i \vee \varphi$ strictly contains that of $(\lambda p,v)\varphi_j \vee \varphi$, then the conjunct corresponding to $Case(i,A(\vec{x}),s)$ cannot be minimal, so $Case(i,A(\vec{x}),s)$ must not hold. These constraints are best understood in lights of the following Theorem 1 which will establish the correctness of the above transformation.

Notice also that this transformation applies only to the indeterminate effect axiom (11). This is why it is beneficial to put as much information as possible into (10).

In the following, we shall denote by T'_{ea} the set of axioms obtained from T_{ea} by replacing every indeterminate effect axiom in it of the form (11) by the axioms (14) - (15). We shall denote by $\mathcal{D}_{case}$ the set of constraints (16) and (17). Notice that this set is also depended on T_{ea}.

Given two theories T_1 and T_2 such that T_1's language is T_2's augmented by a new predicate P, we say that these two theories are equivalent with respect to T_2's language if T_1 is a conservative extension of T_2: a structure is a model of T_2 iff it can be extended into a model of T_1. As it turns out, this is the same as saying that T_2 is the result of *forgetting* P in T_1 according to (Lin and Reiter [7]), and according to a result there, when T_1 is finite, this is the same as saying that T_2 is logically equivalent to the sentence $(\exists P). \bigwedge T_1$, where $\bigwedge T_1$ is the conjunction of the sentences in T_1[5].

We have:

Theorem 1 *Under the unique names axioms $\mathcal{D}_{una}$, the result of circumscribing Caused in $T'_{ea} \cup \mathcal{D}_{case}$ is a conservative extension of the result of circumscribing Caused in T_{ea}:*

$$\mathcal{D}_{una} \models Circum(T_{ea}, Caused) \equiv (\exists Case) Circum(T'_{ea} \cup \mathcal{D}_{case}, Caused).$$

Corollary 1.1 *Under the unique names assumptions, for any formula φ that does not mention Case, $Circum(T_{ea}, Caused) \models \varphi$ iff $Circum(T'_{ea} \cup \mathcal{D}_{case}, Caused) \models \varphi$.*

Computing Successor State Axioms

Having established the correctness of the above transformation, we now proceed to show how to generate successor state axioms from the resulting axioms.

Notice first that the sentence (10) can be rewritten into an axiom of the following form:

$$Poss(A(\vec{x}), s) \wedge \varphi(\vec{x}, p, v, s) \supset Caused(p, v, do(A(\vec{x}), s)).$$

Similarly, we can do the same for axioms of the form (14) - (15). Now from these axioms in T'_{ea}, we can generate, for each fluent F, two axioms of the following forms:

$$Poss(a, s) \wedge \gamma^+_F(\vec{x}, a, s) \supset Caused(F(\vec{x}), true, do(a, s)), \quad (18)$$

$$Poss(a, s) \wedge \gamma^-_F(\vec{x}, a, s) \supset Caused(F(\vec{x}), false, do(a, s)), \quad (19)$$

where γ^+_F and γ^-_F do not quantify over situation variables, and the only situation dependent atomic formulas in them are either of the form $H(t, s)$ or of the form $Case(t_1, t_2, s)$.

Given these two effect axioms, we generate the following successor state axiom for F:

$$Poss(a, s) \supset F(\vec{x}, do(a, s)) \equiv \gamma^+_F(\vec{x}, a, s) \vee (F(\vec{x}, s) \wedge \neg\gamma^-_F(\vec{x}, a, s)). \quad (20)$$

Now let $\mathcal{D}_{ss}$ be the set of successor state axioms, one for each fluent, so generated. Our claim is that, under some reasonable conditions, $\mathcal{D}_{ss}$ captures all the information about the truth values of the fluents in $Cirum(T'_{ea}, Caused) \cup \{1, 2, 4\}$. More precisely, we have:

Theorem 2 *Under the assumption that the following* consistency condition *[10] is satisfied for each fluent F:*

$$\mathcal{D}_{una} \cup \mathcal{D}_{ap} \cup \mathcal{D}_{case} \models (\forall \vec{x}, a, s). Poss(a, s) \supset \neg(\gamma^+_F(\vec{x}, a, s) \wedge \gamma^-_F(\vec{x}, a, s)),$$

the theory

$$\Sigma \cup \mathcal{D}_{una} \cup \mathcal{D}_{ap} \cup \{Circum(T'_{ea}, Casued)\} \cup \mathcal{D}_{case} \cup \{1, 2, 3, 4\}$$

is a conservative extension of the theory

$$\Sigma \cup \mathcal{D}_{una} \cup \mathcal{D}_{ap} \cup \mathcal{D}_{ss} \cup \mathcal{D}_{case} \cup \{3\}.$$

Corollary 2.1 *Under the assumptions in the theorem, for any formula φ that does not mention Caused,*

$$\Sigma \cup \mathcal{D}_{una} \cup \mathcal{D}_{ap} \cup \{Circum(T'_{ea}, Casued)\} \cup \mathcal{D}_{case} \cup \{1, 2, 3, 4\} \models \varphi$$

iff

$$\Sigma \cup \mathcal{D}_{una} \cup \mathcal{D}_{ap} \cup \mathcal{D}_{ss} \cup \mathcal{D}_{case} \cup \{3\} \models \varphi.$$

Corollary 2.2 *Under the assumptions in the theorem, for any formula φ that does not mention Caused and Case,*

$$\Sigma \cup \mathcal{D}_{una} \cup \mathcal{D}_{ap} \cup \{Circum(T_{ea}, Caused)\} \cup \{1, 2, 3, 4\} \models \varphi$$

iff

$$\Sigma \cup \mathcal{D}_{una} \cup \mathcal{D}_{ap} \cup \mathcal{D}_{ss} \cup \mathcal{D}_{case} \cup \{3\} \models \varphi.$$

Proof: Apply Theorem 1 and Theorem 2. ■

Theorem 2 informs us that if we are only concerned with the truth values of fluents, then the original effect axioms as well as the basic axioms about *Caused* can all be discarded. In particular, this is the case with the projection problem.

Technically, the consistency conditions are needed because without these conditions, the successor state axiom (20) may not entail the formula

$$Poss(a, s) \supset \gamma^-_F(\vec{x}, a, s) \supset \neg F(\vec{x}, do(a, s)),$$

which is a consequence of the effect axiom (19) and the two basic axioms (1) and (2) about causality.

[5] Since P is a predicate constant, strictly speaking, we cannot quantify over it in a formula. However, we can consider $(\exists P)\varphi$ as a shorthand for $(\exists p)\varphi'$, where p is a predicate variable of the same arity as P, and φ' is the result of substituting P in φ by p.

Example 1 Consider again our checkerboard example. We shall consider only the successor state axioms for *white* and *black*. The indeterminate effect axiom (13) of *drop* is translated into:

$$\begin{aligned}
&Poss(drop,s) \wedge Case(1,drop,s) \supset \\
&\quad [Caused(white,true,do(drop,s)) \wedge \\
&\quad\quad Caused(black,false,do(drop,s))], \\
&Poss(drop,s) \wedge Case(2,drop,s) \supset \\
&\quad [Caused(white,false,do(drop,s)) \wedge \\
&\quad\quad Caused(black,true,do(drop,s))], \\
&Poss(drop,s) \wedge Case(3,drop,s) \supset \\
&\quad [Caused(white,true,do(drop,s)) \wedge \\
&\quad\quad Caused(black,true,do(drop,s))].
\end{aligned}$$

Together with the original determinate effect axioms, we have:

$$\begin{aligned}
&Poss(a,s) \supset \\
&\quad [a = drop \wedge (Case(1,drop,s) \vee Case(3,drop,s))] \supset \\
&\quad\quad Caused(white,true,do(a,s)), \\
&Poss(a,s) \supset \\
&\quad [a = pickup \vee (a = drop \wedge Case(2,drop,s))] \supset \\
&\quad\quad Caused(white,false,do(a,s)).
\end{aligned}$$

Thus we have the following successor state axiom for *white*:

$$\begin{aligned}
&Poss(a,s) \supset \{white(do(a,s)) \equiv \\
&\quad [a = drop \wedge (Case(1,drop,s) \vee Case(3,drop,s))] \vee \\
&\quad white(s) \wedge \\
&\quad\quad \neg[a = pickup \vee (a = drop \wedge Case(2,drop,s))]\}.
\end{aligned}$$

A similar successor axiom can be obtained for *black*. It can be seen that the consistency conditions are satisfied for both *white* and *black*.

We shall not get into details regarding the accompanied constraints about *Case*, but note that for this example, all constraints of the form (17) are logical consequence of the unique names assumptions. So the following is the only nontrivial constraint about *Case*:

$$Case(1,drop,s) \dot\vee Case(2,drop,2) \dot\vee Case(3,drop,3).$$

Regression and Some of Its Properties

Once we have a successor state axiom for each fluent, regression becomes syntactic substitutions [10]: for any formula $\varphi(s)$ that does not quantify over situation, and action α, the regression of a formula $\varphi(s)$ over α, written $\mathcal{R}(\varphi(s),\alpha)$, is the result of replacing in $\varphi(s)$ every atomic formula of the form $H(F(\vec{t}),s)$ by $\Phi_F(\vec{t},\alpha,s)$, where

$$Poss(a,s) \supset [F(\vec{x},do(a,s)) \equiv \Phi_F(\vec{x},a,s)]$$

is the successor state axiom for F.

The following result is immediate:

Lemma 2 *Let $\mathcal{D}_{ss}$ be a set of successor state axioms, one for each fluent. We have:*

$$\mathcal{D}_{ss} \models (\forall s).Poss(\alpha,s) \supset [\varphi(s) \equiv \mathcal{R}(\varphi,\alpha)].$$

In the rest of this section, we assume that we're given an action theory of the form:

$$\mathcal{D} = \Sigma \cup \mathcal{D}_{una} \cup \mathcal{D}_{ap} \cup \mathcal{D}_{ss} \cup \mathcal{D}_{case} \cup \mathcal{D}_{S_0},$$

where $\mathcal{D}_{case}$ is a set of *Case* constraints of the form (16) or of the form (17), and $\mathcal{D}_{S_0}$ is a set of sentences that do not mention any other situation term except S_0, and do not mention *Poss*, *Caused*, and *Case*. The other components of $\mathcal{D}$ have the usual meaning.

Our main concern is the soundness and completeness of regression for doing temporal projection with respect to the initial database. Our first positive result is about *Case* independent temporal projections:

Theorem 3 *Let $\varphi(s)$ be a formula that does not quantify over situation variable, does not mention any other situation term except s, and does not mention Poss, Caused, and Case. Let α be an action term. If, under $\mathcal{D}_{una}$, $\mathcal{R}(\varphi,\alpha)$ is equivalent to a formula that does not mention Case, then*

$$\mathcal{D} \models \varphi(do(\alpha,S_0))$$

iff

$$\mathcal{D}_{S_0} \cup \mathcal{D}_{una} \models \Psi(S_0) \wedge \mathcal{R}(\varphi,\alpha)(S_0),$$

where $\mathcal{D}_{ap} \models Poss(\alpha,S_0) \equiv \Psi(S_0)$, $\mathcal{R}(\varphi,\alpha)(S_0)$ is obtained from $\mathcal{R}(\varphi,\alpha)$ by replacing s by S_0, and $\varphi(do(\alpha,S_0))$ is obtained from $\varphi(s)$ by replacing s by $do(\alpha,S_0)$.

Notice that this theorem depends on the particular form the constraints in $\mathcal{D}_{case}$ have: they are about *Case* only, that is, the result of forgetting it will yield a tautology: $(\exists Case)\mathcal{D}_{case} \equiv true$.

One of the conditions in Theorem 3 is that under the unique names assumptions, $\mathcal{R}(\varphi,\alpha)$ be equivalent to a formula that does not mention *Case*. This condition holds if the action α's effects on the fluents in φ are definite. Thus Theorem 3 informs us for reasoning about the determinate effects of actions, the auxiliary predicate *Case* can be rightly ignored.

When either $\varphi(s)$ or its regression mentions *Case*, we need to include constraints on *Case*:

Theorem 4 *Let $\varphi(s)$ be a formula that does not quantify over situation variables, and does not mention Poss and Caused. Let α be an action term. If $\mathcal{D}_{case}$ does not mention H, then*

$$\mathcal{D} \models \varphi(do(\alpha,S_0))$$

iff

$$\mathcal{D}_{S_0} \cup \mathcal{D}_{una} \cup \mathcal{D}_{case} \models \Psi(S_0) \wedge \mathcal{R}(\varphi,\alpha)(S_0).$$

Given the forms (16) and (17) the constraints in $\mathcal{D}_{case}$ must take, $\mathcal{D}_{case}$ does not mention H if all the indeterminate effects of actions are context free. This condition is needed because although $\mathcal{D}_{case}$ itself contains no information about H, it can when used together with some assumptions about $Case$ that can be easily incorporated into the query $\varphi(s)$.

Finally, notice that Theorem 3 and Theorem 4 can be generalized to temporal projections with sequences of actions.

The Ramification Problem

Although the framework of [5] was introduced for handling the ramification problem, we have so far ignore this problem in this paper. We now show how the indirect effects of actions can be represented.

Consider again our checkerboard example. Suppose that for whatever reasons, whenever the pin is touching both a white and a black square, our friend Fred will be happy. This constraint can be represented as:

$$white(s) \wedge black(s) \supset Caused(happy, true, s). \quad (21)$$

Notice that this causal law is different from the effect axioms in that the situation in both the antecedent and the consequent is the same. Thus it plays the role of a traditional domain constraint. With this causal rule, under the unique names assumptions about fluents, circumscribing $Caused$ will yield:

$$Caused(happy, v, s) \equiv (v = true \wedge white(s) \wedge black(s)).$$

And we have:

$$\begin{array}{l}
Poss(a, s) \supset \\
\quad \{[white(do(a, s)) \equiv \\
\qquad (a = drop \vee (white(s) \wedge a \neq pickup))] \wedge \\
\quad [black(do(a, s)) \equiv \\
\qquad (a = drop \vee (black(s) \wedge a \neq pickup))] \wedge \\
\quad [happy(do(a, s)) \equiv (a = drop \vee happy(s))]\} \vee \\
\quad \{[white(do(a, s)) \equiv \\
\qquad (a = drop \vee (white(s) \wedge a \neq pickup))] \wedge \\
\quad [black(do(a, s)) \equiv \\
\qquad (black(s) \wedge a \neq pickup \wedge a \neq drop))] \wedge \\
\quad [happy(do(a, s)) \equiv happy(s)]\} \vee \\
\quad \{[white(do(a, s)) \equiv \\
\qquad (white(s) \wedge a \neq pickup \wedge a \neq drop))] \wedge \\
\quad [black(do(a, s)) \equiv \\
\qquad (a = drop \vee (black(s) \wedge a \neq pickup))] \\
\quad [happy(do(a, s)) \equiv happy(s)]\}.
\end{array}$$

Notice that two of the three cases include the frame axiom $happy(do(a, s)) \equiv happy(s)$. This can be avoided if we use the predicate $Case$.

The constraint (21) and the original effect axioms of *drop* yields the following new effect axioms:

$$\begin{array}{l}
Poss(drop, s) \supset (\forall p, v)\{(p = holding \wedge v = false) \supset \\
\qquad Caused(p, v, do(drop, s))\},
\end{array}$$

$$\begin{array}{l}
Poss(drop, s) \supset \\
\quad (\forall p, v)\{[p = white \wedge v = true \vee \\
\qquad p = black \wedge v = false] \supset \\
\qquad\quad Caused(p, v, do(drop, s))\} \vee \\
\quad (\forall p, v)\{[p = white \wedge v = black \vee \\
\qquad p = black \wedge v = true] \supset \\
\qquad\quad Caused(p, v, do(drop, s))\} \vee \\
\quad (\forall p, v)\{[p = white \wedge v = true \vee \\
\qquad p = black \wedge v = true \vee \\
\qquad p = happy \wedge v = true] \supset \\
\qquad\quad Caused(p, v, do(drop, s))\}.
\end{array}$$

Apply the transformation of Section to this set of effect axioms will yield the same successor state axioms in Example 1 for $white$ and $black$, and the following one for $happy$:

$$\begin{array}{l}
Poss(a, s) \supset happy(do(a, s)) \equiv \\
\quad (a = drop \wedge Case(drop, 3, s)) \vee happy(s).
\end{array}$$

This example shows the flexibility of the $Case$ predicate when there are state constraints.

Constraints can also yield implicit indeterminate effects. For instance, if there are k black squares, then we'll have:

$$\begin{array}{l}
Caused(black, true, s) \equiv \\
\quad Caused(black_1, true, s) \vee \cdots \vee \\
\quad Caused(black_k, true, s), \\
Caused(black, false, s) \equiv \\
\quad Caused(black_1, false, s) \wedge \cdots \wedge \\
\quad Caused(black_k, false, s), \\
(\exists v)Caused(black, v, s) \equiv (\exists v)Caused(black_i, v, s),
\end{array}$$

for any $1 \leq i \leq k$. Notice that the last axiom schema postulates that whenever an action affects the truth values of $black_i$, then it also affects the truth values of $black_j$, for any $1 \leq i, j \leq k$.

Related Work and Discussions

Epistemologically, we have shown how the causal minimization framework of [5] can be used to specify the indeterminate effects of actions. Computationally, we have shown how goal regression can be used to reason about them.

There have been other proposals in the literature (e.g. [1, 2, 3, 12]) for specifying the effects of indeterminate actions. To the best of our knowledge, the computational contribution of this work is novel.

Among the extant approaches, the ones in [3] and [1] seem closest to ours. As we mentioned in Section , the *releases* propositions of [3]: A **releases** F corresponds to the following axiom in our language:

$$\begin{array}{c}
Poss(A, s) \supset Caused(F, true, do(A, s)) \vee \\
Caused(F, false, do(A, s)).
\end{array}$$

Regarding the work of [1], the $In(F)$ and $Out(F)$ predicates there correspond to $Caused(F, true, do(a, s))$ and $Caused(F, false, do(a, s))$, respectively, in our language. However, the formalism of [3] is limited because no complex *releases* propositions are allowed. For instance, one cannot write expressions like

$$(\forall a).a \textbf{ releases } F \leftrightarrow a \textbf{ releases } F'.$$

The formalism of [1] is also limited because the action parameters of its *In* and *Out* predicates are not made explicit, thus cannot be quantified over.

Finally, we want to remark on the auxiliary predicate *Case*. In this paper, we have used it entirely for computational purposes. However, there are some interesting possible interpretations of this predicate.

There is a sense that *Case* can be interpreted in probabilistic terms. For instance, if

$$\begin{aligned}&Poss(drop, s) \wedge Case(1, drop, s) \supset \\ &\quad Caused(white, true, do(drop, s)) \wedge \\ &\quad Caused(black, false, do(drop, s)),\end{aligned}$$

then $Case(1, drop, s)$ may stand for the probability of the pin lying entirely within a white square after it has been dropped. Under this interpretation, the first constraint (16) on *Case*, in this example the following one:

$$Case(1, drop, s) \dot{\vee} Case(2, drop, s) \dot{\vee} Case(3, drop, s),$$

says that the explicitly enumerated possible outcomes are both exclusive and exhaustive, and the constraints (17) simply eliminate redundant outcomes. In this regard, it would be interesting to formally connect our approach to probabilistic ones. This is a future research that we're pursuing.

Another possible interpretation of *Case* is based on the view that in principle, it is always possible to reduce indeterminate actions to determinate ones, and one way of doing this is to introduce new fluents to name those low level contexts under which the effects of actions will be determinate. According to this view, *Case* can be seen as playing the role of such new fluents. For instance, $Case(1, drop, s)$ may name the context under which *drop* has the effect of causing the pin lying entirely within a white square. We are currently exploring the possible impact of this interpretation as well.

Acknowledgements

Thanks to the other members of the University of Toronto Cognitive Robotics group (Yves Lesperance, Hector Levesque, Daniel Marcu, and Ray Reiter), to Vladimir Lifschitz, and to Yan Zhang for helpful discussions and comments. This research was supported by grants from the Government of Canada Institute for Robotics and Intelligent Systems, and from the National Science and Engineering Research Council of Canada.

References

[1] C. Barel. Reasoning about actions: nondeterministic effects, constraints, and qualification. In *Proc. of IJCAI'95*, pp. 2017-2023.

[2] C. Boutilier and N. Friedman. Nondeterministic actions and the frame problem. In *Workign Notes of the AAAI Spring Symposium on Extending Theories of Action*, pages 39–44, 1995.

[3] G. N. Kartha and V. Lifschitz. Action with indirect effects (preliminary report). In *Proc. of KR'94*, pp. 341-350.

[4] V. Lifschitz. Computing circumscription. In *Proc. of IJCAI'85*, pp. 121-127.

[5] F. Lin. Embracing causality in specifying the indirect effects of actions. In *Proc. of IJCAI'95*, pp. 1985-1993.

[6] F. Lin and R. Reiter. State constraints revisited. *J. of Logic and Computation*, 4(5):655–678, 1994.

[7] F. Lin and R. Reiter. Forget it! In R. Greiner and D. Subramanian, editors, *Working Notes of AAAI Fall Symposium on Relevance*, pp. 154–159, 1994.

[8] J. McCarthy and P. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, Edinburgh, 1969.

[9] E. P. Pednault. Synthesizing plans that contain actions with context-dependent effects. *Computational Intelligence*, 4:356–372, 1988.

[10] R. Reiter. The frame problem in the situation calculus: a simple solution (sometimes) and a completeness result for goal regression. In V. Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*, pages 418–420. Academic Press, San Diego, CA, 1991.

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

[12] Y. Zhang. *Reasoning About Persistence: A Unified Principle for State Change*. PhD thesis, Department of Computer Science, Sydney University, Sydney, Australia, 1994.