An assessment of actions with indeterminate and indirect effects in some causal approaches

Andreas Herzig Ivan Varzinczak
IRIT – Université Paul Sabatier
118 route de Narbonne, F-31062
Toulouse Cedex 4, France
e-mail: {herzig,ivan}@irit.fr

Abstract

In this work we investigate the behavior of the main existing fluent-indexed approaches to reasoning about actions in dealing with domains that have actions with both indeterminate and indirect effects. As they all fail in solving this kind of scenarios, we argue for an action-indexed causal notion in order to deal with the frame and ramification problems. This is achieved by defining a weak form of causality in terms of a dependence relation involving actions, literals and formulae. This relation allows the literals to change their truth value without forcing or causing it. Once integrated in the framework of a propositional logic close to PDL, it gives us a simple and powerful formalism to reasoning about actions and a decision procedure in terms of tableau methods. We also show how our approach can deal with scenarios involving indeterminate and indirect effects without being subjected to the same problems of other formalisms. In order to help the knowledge engineer in describing domains, we give an interactive algorithm for generating the required dependence information.
Keywords: Reasoning about actions, causality, dependence relation.

Technical Report IRIT/2004–08–R

May 2004

1 Introduction

In the recent literature on reasoning about actions, the concept of causality has been studied as a means of overcoming the inadequacy of state constraints in tackling the ramification problem. In this sense, many types of *causal notions* have been proposed and causality has been considered in different ways: *strong* or *weak* causality (if we always *force* or only *permit* something to be caused); as a *predicate*, a *relation* or a *modality*; and *primitive* (built in the logic) or *derived* (with the aid of some meta-logical information).

Many approaches consider that it is a change in some property that produces (causes) change of some other property. We call them *fluent-indexed approaches*, for they always relate pairs of literals or formulae.

In this work we argue that fluent-indexed approaches are not enough for dealing with the ramification problem. In special in domains involving actions with both nondeterministic and indirect effects. We do this by showing an example of this class of action domain and trying to represent it in some of the most recent formalisms.

Due to the nature of the problems presented by fluent-indexed approaches, we conjecture that causality must be action-indexed, and we propose a weak causal notion based on a contextual dependence relation for dealing with the frame and ramification problems. This dependence relation, once integrated in the framework of the logic of actions and plans $\mathcal{LAP}$ [1], gives us a simple and powerful formalism to reason about actions and a decision procedure in terms of semantic tableaux.

The present text is organized as follows: in Section 2 we present an example of a scenario having actions with both indeterminate and indirect effects, which leads to counterintuitive results when formalized in fluent-indexed approaches. In Section 3 we compile some of the directives commonly agreed on in the reasoning about actions field with respect to representation of indirect and indeterminate effects. In Section 4 we show the representation of the scenario of Section 2 in the most important formalisms found in the literature, pointing at the problems they present. In Section 5 we propose our action-indexed logical framework for reasoning about actions and present an algorithm for automatically generating dependence relations. In the final section we give some conclusions and future work.

2 The Mailboxes Scenario

In this section we sketch the Mailboxes Scenario, which was originally defined in [2]. In essence, it combines Reiter's "dropping a coin on a chessboard" example with Sandewall's argument against causality-based solutions to the ramification problem [14].

In such a scenario, we reason about the status of a particular e-mail message. The domain description is as follows: Suppose $Mbox1$ means "the message is in mailbox 1", and $Mbox2$ "the message is in mailbox 2". We represent the fact that the e-mail is saved in $Mbox1$ or in $Mbox2$ or in both by the literal $Saved$.

Hence the static law (alias domain constraint) for this example is

$$Saved \leftrightarrow (Mbox1 \vee Mbox2)$$

in formalisms that are not situation-indexed, and

$$Holds(Saved, s) \leftrightarrow (Holds(Mbox1, s) \vee Holds(Mbox2, s))$$

in situation-indexed formalisms such as the situation calculus [11]. (As usual, we assume all free variables denoting situations are universally quantified.)

Consider the actions $save1$ and $save2$, whose direct effects are to save an e-mail message in $Mbox1$ and in $Mbox2$, respectively. Suppose we also have a nondeterministic $save$ action, whose direct effect is $Saved$, i.e. saving the e-mail in one of the two mailboxes or in both. Hence $save$ has the indirect effect $Mbox1 \vee Mbox2$. This is also an indeterminate effect. Note that, in particular, after executing $save$ it is also possible to have $Mbox1 \wedge Mbox2$. This is just as in Reiter's "dropping a coin on a chessboard" example, where $drop$ has the possible effect $Black \wedge White$.[1]

In the rest of this work, a domain description is a collection of formulae that is made up of

- static laws as the above, that do not speak about actions;
- action laws, each of which relates an action to its effects and preconditions.

[1]It is possible as well to rephrase our example in terms of Reiter's: we can regard *save* action as *drop*, which means putting a pin on a white, a black, or both squares (the pin lying on the region between two squares). *save*1 (resp. *save*2) can be seen as analogous to *drop*1 (resp. *drop*2), which means putting the pin in a black (resp. white) square.

3 Some postulates about effects of actions

In this section we recall some of the main directives commonly agreed on in the reasoning about actions field with respect to the representation of the effects of actions, stating them as postulates that are going to function as guidelines for the analysis in the rest of this work.

3.1 Representing direct and indirect effects

In every paper in the reasoning about actions literature, it is more or less tacitly supposed that action laws should obey some principles of parsimony and modularity. In particular, the following seem to be commonly agreed:

P1 **Do not state non-effects**.

This means that there should be no frame axioms. For example, in the Mailboxes Scenario, the action law for $\mathit{save}1$ should not mention $\mathit{Mbox}2$. Postulate P1 stems from McCarthy and Hayes' paper [11] and it can be said to be at the origin of the work in reasoning about actions.

P2. **Do not state indirect effects**.

For instance the description of save should mention neither $\mathit{Mbox}1$ nor $\mathit{Mbox}2$. Postulate P2 has been identified in Finger's PhD thesis [3] and is the core of the ramification problem.

3.2 Reasoning about indeterminate effects

In order to correctly reason about a nondeterministic action, we have to be able of properly treating its set of indeterminate effects. This gives us the following postulate:

P3. **Do not systematically interpret effects described with the inclusive disjunction "$\vee$" as the exclusive one "$\oplus$"**.

For example, in the Mailboxes Scenario, the effect of save should not be equivalent to $\mathit{Mbox}1 \oplus \mathit{Mbox}2$. The motivation for such a postulate has been originally suggested by Reiter.

As we will see along the following sections, the Mailboxes Scenario is problematic for all the existing approaches allowing for the representation of actions with both indirect and indeterminate effects, with respect to Postulates P1–P3. In what follows, we discuss the approaches of Lin [7, 8], McCain and Turner [9, 10], Thielscher [15, 16] and Zhang and Foo [18]. Indeed, it can be shown that in all these frameworks either Postulate P1 is violated, or Postulate P2, or, in order not to violate Postulate P3, the action *save*1 has the indirect indeterminate effect of changing *Mbox*2, which is clearly counterintuitive.

4 Causal approaches to ramification

In this section we analyze how the most known causal approaches in the literature perform in domains involving actions with indirect and indeterminate effects. In order to do this, in all of them we formalize the example of Section 2.

In the rest of this work we will use the following definitions: we define $ACT = \{\alpha, \beta, \ldots\}$ as the set of *actions*, like *shoot*, *load*, etc. $ATM = \{P, Q, \ldots\}$ is the set of *atomic formulae*, or *atoms* (fluents), for short. Examples of atoms are *Loaded* and *Alive*. $LIT = ATM \cup \{\neg P : P \in ATM\}$ is the set of *literals*. The set of classical propositional formulae will be denoted by *PFOR*.

4.1 Minimization of causality

We here examine the behavior of Lin's approach [7, 8] in solving the Mailboxes Scenario.

Roughly speaking, Lin proposes to add a predicate *Caused* to the situation calculus to describe the appropriate relationships between fluents by means of this predicate, and to circumscribe it. $Caused(P, v, s)$ reads as "fluent P is caused to have truth value v in situation s".

In addition, the following axiom is assumed:

$$Caused(P, true, s) \rightarrow Holds(P, s)$$

which states that something that is caused in a situation s must hold in such a situation.

The way domain constraints and effect axioms are stated defines a fluent-indexed strong causal notion.

In the example that follows we describe the Mailboxes Scenario using this formalism.

Following the definitions in the original work, the effect axioms for this scenario are:

$$\mathit{Poss}(\mathit{save}1, s) \rightarrow \mathit{Caused}(\mathit{Mbox}1, \mathit{true}, \mathit{do}(\mathit{save}1, s)) \tag{1}$$
$$\mathit{Poss}(\mathit{save}2, s) \rightarrow \mathit{Caused}(\mathit{Mbox}2, \mathit{true}, \mathit{do}(\mathit{save}2, s)) \tag{2}$$
$$\mathit{Poss}(\mathit{save}, s) \rightarrow \mathit{Caused}(\mathit{Saved}, \mathit{true}, \mathit{do}(\mathit{save}, s)) \tag{3}$$

Then, according to Lin, we have to supplement the static law $\mathit{Saved} \leftrightarrow (\mathit{Mbox}1 \vee \mathit{Mbox}2)$ in the following way: as $\mathit{save}1$ (resp. $\mathit{save}2$) has effect $\mathit{Mbox}1$ (resp. $\mathit{Mbox}2$) and $\mathit{Mbox}1$ (resp. $\mathit{Mbox}2$) being true causes the truth of Saved, then we must causally relate $\mathit{Mbox}1$ (resp. $\mathit{Mbox}2$) and Saved. This is done stating the formulae:

$$\mathit{Caused}(\mathit{Mbox}1, \mathit{true}, s) \rightarrow \mathit{Caused}(\mathit{Saved}, \mathit{true}, s) \tag{4}$$
$$\mathit{Caused}(\mathit{Mbox}2, \mathit{true}, s) \rightarrow \mathit{Caused}(\mathit{Saved}, \mathit{true}, s) \tag{5}$$

The other way round, as an execution of save has the direct effect Saved and a change in Saved means a change in $\mathit{Mbox}1$ and/or in $\mathit{Mbox}2$, we are obliged to causally relate Saved with both $\mathit{Mbox}1$ and $\mathit{Mbox}2$. This is done stating the formula:

$$\mathit{Caused}(\mathit{Saved}, \mathit{true}, s) \rightarrow \mathit{Caused}(\mathit{Mbox}1, \mathit{true}, s) \vee \mathit{Caused}(\mathit{Mbox}2, \mathit{true}, s) \tag{6}$$

Stating just these laws, according to the minimization process defined in [8], we would get an exclusive interpretation of the disjunction in (6), i.e. save would have the indirect effect $\mathit{Mbox}1 \oplus \mathit{Mbox}2$. So, in order to capture the possibility of save saving the e-mail in both mailboxes, in Lin's approach we have also to state the constraints:[2]

$$\mathit{Caused}(\mathit{Saved}, \mathit{true}, s) \rightarrow \mathit{Caused}(\mathit{Mbox}1, \mathit{true}, s) \vee \mathit{Caused}(\mathit{Mbox}1, \mathit{false}, s) \tag{7}$$
$$\mathit{Caused}(\mathit{Saved}, \mathit{true}, s) \rightarrow \mathit{Caused}(\mathit{Mbox}2, \mathit{true}, s) \vee \mathit{Caused}(\mathit{Mbox}2, \mathit{false}, s) \tag{8}$$

Thus, we have the following theorem:

[2]It is worth noting that both consequents of (7) and (8) are not tautologies (cf. [7]).

Theorem 4.1 From formulae (1)–(8) we can derive

$$\begin{aligned} \mathit{Poss}(\mathit{save}1, s) \rightarrow \quad & \mathit{Caused}(\mathit{Mbox}2, \mathit{true}, \mathit{do}(\mathit{save}1, s)) \vee \\ & \mathit{Caused}(\mathit{Mbox}2, \mathit{false}, \mathit{do}(\mathit{save}1, s)) \end{aligned}$$

Proof: Suppose that $\mathit{Poss}(\mathit{save}1, s)$ is the case. Then, from Formula (1) we obtain $\mathit{Caused}(\mathit{Mbox}1, \mathit{true}, s')$, where s' stands for $\mathit{do}(\mathit{save}1, s)$. From this and Formula (4), we can get $\mathit{Caused}(\mathit{Saved}, \mathit{true}, s')$. Thus, constraint (7) gives us $\mathit{Caused}(\mathit{Mbox}1, \mathit{true}, s') \vee \mathit{Caused}(\mathit{Mbox}1, \mathit{false}, s')$. Nevertheless, even with the minimization policy defined in [8], it is still possible to derive another extension: from $\mathit{Caused}(\mathit{Saved}, \mathit{true}, s')$ and constraint (8) we conclude $\mathit{Caused}(\mathit{Mbox}2, \mathit{true}, s') \vee \mathit{Caused}(\mathit{Mbox}2, \mathit{false}, s')$. ■

So, we get that an execution of *save*1 can produce the indirect effect of changing *Mbox*2. But we do not want such an indirect effect, for *save*1 would be nondeterministic. A possible solution for this could be to state

$$(\mathit{Poss}(\mathit{save}1, s) \wedge \neg \mathit{Holds}(\mathit{Mbox}2, s)) \rightarrow \mathit{Caused}(\mathit{Mbox}2, \mathit{false}, \mathit{do}(\mathit{save}1, s))$$

from which we derive

$$(\mathit{Poss}(\mathit{save}1, s) \wedge \neg \mathit{Holds}(\mathit{Mbox}2, s)) \rightarrow \neg \mathit{Holds}(\mathit{Mbox}2, \mathit{do}(\mathit{save}1, s))$$

which would violate Postulate P1.

Another way of tackling the problem is stating

$$\mathit{Poss}(\mathit{save}1, s) \rightarrow \mathit{Caused}(\mathit{Mbox}2, \mathit{false}, \mathit{do}(\mathit{save}1, s))$$

but, this is unintuitive, for in a situation where we already had *Saved*, with the e-mail in *Mbox*2, saving again with *save*1 would make a change in *Mbox*2.

4.2 Causal laws approach

In this section, we formalize the Mailboxes Scenario using the base formalism proposed by McCain and Turner [9]. Their approach considers that background knowledge about causation should be given in form of *causal laws*, which are stated as sentences in a modal, conditional logic with the aid of a causal modal operator $\Rightarrow$.

A causal law of the form $A \Rightarrow C$, where A and C are propositional formulae, is read as "A causes C", or "the truth of A determines the truth of C". This is thus a fluent-indexed causal approach.

Let *LAW* be the set of all causal laws concerning a given domain. A set of formulae Γ is *closed* under *LAW* if and only if whenever $A \Rightarrow C$ is in *LAW* and $A \in \Gamma$, then $C \in \Gamma$. $\Gamma \vdash_{LAW} A$ means that formula A belongs to the smallest set of formulae containing Γ that is closed w.r.t. propositional logic and also closed under *LAW*.[3]

In the formalization that follows, *KB* represents an initial knowledge base and $\mathcal{E}$ a set of direct effects.

With the causal laws approach, the representation of the Mailboxes Scenario is as follows:

$$LAW = \left\{ \begin{array}{c} Saved \Rightarrow (Mbox1 \vee Mbox2), \\ (Mbox1 \vee Mbox2) \Rightarrow Saved \end{array} \right\}$$

The causal law $Saved \Rightarrow (Mbox1 \vee Mbox2)$ is needed because the truth of fluent *Saved* causes the truth of formula $Mbox1 \vee Mbox2$. Analogously, $(Mbox1 \vee Mbox2) \Rightarrow Saved$ is necessary because $Mbox1 \vee Mbox2$ being true causes *Saved* also to be true.[4]

Completing the domain description, we have a set of initial observations:

$$KB = \{\neg Mbox1, \neg Mbox2, \neg Saved\}$$

and we suppose that *Mbox*1 was produced as a direct effect:

$$\mathcal{E} = \{Mbox1\}$$

From this representation and according to McCain and Turner's approach defined in [9], after *save* action we get an exclusive interpretation of the disjunction $Mbox1 \vee Mbox2$, violating Postulate P3. This is shown by the following theorem:

Theorem 4.2 After executing *save* we will get only the next two possible states:

$$\left\{ \begin{array}{c} \{Mbox1, \neg Mbox2, Saved\}, \\ \{\neg Mbox1, Mbox2, Saved\} \end{array} \right\}$$

[3]Observe that this definition also uses a kind of minimization policy to determine the possible states after execution of actions.

[4]Observe that instead of $(Mbox1 \vee Mbox2) \Rightarrow Saved$ one could have as well the causal laws $Mbox1 \Rightarrow Saved$ and $Mbox2 \Rightarrow Saved$, whose justifications are straightforward. On the other hand, we could not replace $Saved \Rightarrow (Mbox1 \vee Mbox2)$ by $Saved \Rightarrow Mbox1$ and $Saved \Rightarrow Mbox2$, for in this case *save* would always cause $Mbox1 \wedge Mbox2$.

Proof: Following the definitions in [9], for any knowledge base KB, any direct effects $\mathcal{E}$, and any set LAW of causal laws, the set of possible next states after performing an action is the set of interpretations KB' such that:

$$KB' = \{L : L \text{ is a literal and } (KB \cap KB') \cup \mathcal{E} \vdash_{LAW} L\}$$

where $\vdash_{LAW}$ is derivability w.r.t. the causal laws defined in LAW.

After performing action *save*, we have the direct effect *Saved*. For the case $KB_1 = \{Mbox1, \neg Mbox2, Saved\}$, we have $KB \cap KB_1 = \{\neg Mbox2\}$ and $\{\neg Mbox2\} \cup \{Saved\} \vdash_{LAW} Mbox1$, and this possible state is OK. In the case $KB_2 = \{\neg Mbox1, Mbox2, Saved\}$, we have $KB \cap KB_2 = \{\neg Mbox1\}$ and $\{\neg Mbox1\} \cup \{Saved\} \vdash_{LAW} Mbox2$, and this possible state is OK, too. The interpretation $KB_3 = \{\neg Mbox1, \neg Mbox2, Saved\}$ is not a possible state as clearly KB_3 is not closed under LAW. Now, considering the case $KB_4 = \{Mbox1, Mbox2, Saved\}$, we have $KB \cap KB_4 = \emptyset$ and neither $\emptyset \cup \{Saved\} \not\vdash_{LAW} Mbox1$ nor $\emptyset \cup \{Saved\} \not\vdash_{LAW} Mbox2$, so KB_4 is not closed under LAW. Thus, the only possible states after performing *save* action are KB_1 and KB_2 and from this the result follows.[5] ■

In order to avoid exclusive interpretation of disjunctions, we have to relax inertia by increasing LAW with the following causal laws

$$(Saved \wedge Mbox1) \Rightarrow Mbox1$$
$$(Saved \wedge Mbox2) \Rightarrow Mbox2$$

However with this apparent solution we get that an execution of *save*1 could make a change in $Mbox2$, for the interpretation $\{Mbox1, Mbox2, Saved\}$ would be closed under LAW.

4.3 Another version of causal laws

In this section, we use the improved version of causal laws as given in [10]. Basically, the difference is that in this approach actions are explicited and each action, fluent and formula has an associated time point. So, for example, $save1_2$ means that the action of saving the e-mail in mailbox 1 was executed at time point 2, and having $Mbox1_3$ means that at time point 3, the e-mail is

[5]The reader is invited to verify that with the causal laws $Mbox1 \Rightarrow Saved$ and $Mbox2 \Rightarrow Saved$ instead of $(Mbox1 \vee Mbox2) \Rightarrow Saved$ one obtains the same result.

saved in mailbox 1 (independently of the action that was executed to achieve that).

Besides considering time, the following *standard schemas* are also assumed (remembering that α stands for action names, P for atom (fluent) names, and A for a formula):

$$\alpha_t \Rightarrow \alpha_t \tag{9}$$

$$\neg\alpha_t \Rightarrow \neg\alpha_t \tag{10}$$

$$P_0 \Rightarrow P_0 \tag{11}$$

$$\neg P_0 \Rightarrow \neg P_0 \tag{12}$$

$$A_t \wedge A_{t+1} \Rightarrow A_{t+1} \tag{13}$$

Schema (9) (resp. (10)) states that the occurrence (resp. non-occurrence) of action α at time t is caused whenever α occurs (resp. does not occur) at t. The Schemas (11) and (12) represent facts about the initial values of each fluent. Schema (13) formalizes the common sense law of inertia, representing the fact that whenever an inertial fluent holds at two successive time points, its truth at the second time point is taken to be caused simply by virtue of its persistence.

Using this approach, we formalize the Mailboxes Scenario in the following way. *LAW*, *KB* and $\mathcal{E}$ are as in section 4.2, except that they are time-indexed:

$$LAW = \left\{ \begin{array}{c} save1_t \wedge \neg Mbox1_t \Rightarrow Mbox1_{t+1}, \\ save2_t \wedge \neg Mbox2_t \Rightarrow Mbox2_{t+1}, \\ save_t \Rightarrow Saved_{t+1}, \\ Saved_t \Rightarrow (Mbox1_t \vee Mbox2_t), \\ (Mbox1_t \vee Mbox2_t) \Rightarrow Saved_t \end{array} \right\}$$

$$KB = \{\neg Mbox1_0, \neg Mbox2_0, \neg Saved_0\}$$

So, we can state the theorem below:

Theorem 4.3 *LAW* and *KB* above violate Postulate P3: we get an exclusive interpretation of the nondeterminism of *save* action.

Proof: Analogous to that of Theorem 4.2, just considering time points in each fluent. ■

As before, if we relax inertia by means of some extra causal laws, we will also get that *save*1 may cause a change in *Mbox*2.

4.4 Postprocessing approach

In this section we examine the postprocessing generation of ramifications presented by Thielscher in [15].

The basic idea of this approach consists in admitting states not satisfying the domain constraints, which are seen as “intermediate states”. “Stable” states are obtained after successive applications of the so called *causal relations.* A causal relation L_1 causes L_2 if A, where L_1, L_2 are literals, and A is a formula, is the way a fluent indexed causal notion is defined in this approach.

In what follows, an *action law* is a triple $\langle C, \alpha, E\rangle$, where α is an action, C its preconditions and E its effects, such that $|C| = |E|$ (C and E have the same atoms). An *influence relation* is a relation between atoms (fluents) that is used to automatically generate the causal relations. Saying that a pair (P_1, P_2), where P_1 and P_2 are atoms, is in the influence relation means that a change in the truth value of P_1 *may* cause a change in the truth value of P_2.

A state of the world (not necessarily satisfying the domain constraints) is a pair $(KB, \mathcal{E})$, where KB is a knowledge base and $\mathcal{E}$ a set of direct effects. Performing an action α in a state of affairs KB corresponds to applying its associated action law $\langle C, \alpha, E\rangle$ to the pair $(KB, \mathcal{E})$, giving us a new pair $(KB', \mathcal{E}')$, where $KB' = (KB \setminus C) \cup E$ and $\mathcal{E}' = \mathcal{E} \cup E$.

With this approach, we define the following action laws:

$$\langle \{\neg Mbox1\}, save1, \{Mbox1\}\rangle \tag{14}$$

$$\langle \{\neg Mbox2\}, save2, \{Mbox2\}\rangle \tag{15}$$

$$\langle \{\neg Saved\}, save, \{Saved\}\rangle \tag{16}$$

Action law (14) expresses that “in a state where $Mbox1$ is false, after executing $save1$, $Mbox1$ will be true”.[6] For action laws (15) and (16) the reading is analogous.

The set of domain constraints is the singleton:

$$\{Saved \leftrightarrow (Mbox1 \vee Mbox2)\}$$

[6]Note that according to the definitions in [15], if we had a state where the e-mail is already saved in mailbox 1, it would not be possible to save it again with $save1$, which is a limitation of this approach. A possible solution for this problem should be stating an action law as $\langle \{Mbox1\}, save1, \{Mbox1\}\rangle$. Note, however, that this would violate Postulate P1 stated in Section 2.

According to Thielscher's approach, as for this example a change in *Mbox*1 (resp. *Mbox*2) may cause a change in *Saved* and vice-versa, we have to define the influence relation for this scenario as follows:

$$\left\{\begin{array}{c}(\mathit{Mbox}1, \mathit{Saved}),\\(\mathit{Mbox}2, \mathit{Saved}),\\(\mathit{Saved}, \mathit{Mbox}1),\\(\mathit{Saved}, \mathit{Mbox}2)\end{array}\right\}$$

From this influence information and Algorithm 1 given in [15], we obtain the following set of causal relations:

$$\left\{\begin{array}{c}\mathit{Mbox}1 \textsf{ causes } \mathit{Saved} \textsf{ if } \top,\\\mathit{Mbox}2 \textsf{ causes } \mathit{Saved} \textsf{ if } \top,\\\mathit{Saved} \textsf{ causes } \mathit{Mbox}1 \textsf{ if } \top,\\\mathit{Saved} \textsf{ causes } \mathit{Mbox}2 \textsf{ if } \top\end{array}\right\}$$

Thus, with this domain description, we get the following theorem:

Theorem 4.4 With Thielscher's approach, action *save*1 is nondeterministic with the indirect effect of possibly causing a change in *Mbox*2.

Proof: Suppose an initial state where we have $\{\neg \mathit{Mbox}1, \neg \mathit{Mbox}2, \neg \mathit{Saved}\}$. Then, if we apply the action law

$$\langle \{\neg \mathit{Mbox}1\}, \mathit{save}1, \{\mathit{Mbox}1\}\rangle$$

to the pair

$$(\{\neg \mathit{Mbox}1, \neg \mathit{Mbox}2, \neg \mathit{Saved}\}, \{\})$$

we will get the resulting pair

$$(\{\mathit{Mbox}1, \neg \mathit{Mbox}2, \neg \mathit{Saved}\}, \{\mathit{Mbox}1\})$$

As this state is inconsistent w.r.t. the domain constraint, we apply the causal relations to it and obtain

$$(\{\mathit{Mbox}1, \neg \mathit{Mbox}2, \mathit{Saved}\}, \{\mathit{Mbox}1, \mathit{Saved}\})$$

which is a successor state [15].

However we can apply the causal relation Saved causes $\mathit{Mbox2}$ if $\top$ to this pair as well, obtaining

$$(\{\mathit{Mbox}1, \mathit{Mbox}2, \mathit{Saved}\}, \{\mathit{Mbox}1, \mathit{Saved}, \mathit{Mbox}2\})$$

that is a successor state, too.

So, we have two possible successor states for action *save*1, one of them making a change in *Mbox*2. ■

Then, we get that with Thielscher's approach we cannot solve the Mailboxes Scenario, for it is possible to derive an unintuitive result (*save*1 causing a change in *Mbox*2). At first glance, we see no way of overcoming this problem inside the original definitions made in [15, 16]. Just in the same way as in the previously discussed approaches, if we do not state the relation between *Saved* and *Mbox*2, it would not be possible to obtain the nondeterminism of *save* action.

4.5 EPDL approach

We now formalize the Mailboxes Scenario using the base logic EPDL [18]. Such a logic is an extension of PDL [5] that allows a proposition as a modality for specifying the indirect effects of actions. In this sense, we are able to write formulae like $[\mathit{Mbox}1]\mathit{Saved}$, which states that in all possible worlds in which *Mbox*1 is true, *Saved* is caused to be true.

The action description of the Mailboxes Scenario in EPDL is given bellow:

$$\left\{\begin{array}{l} \langle \mathit{save}\rangle\top, \langle \mathit{save}1\rangle\top, \langle \mathit{save}2\rangle\top, \\ \mathit{Saved} \leftrightarrow (\mathit{Mbox}1 \lor \mathit{Mbox}2), \\ [\mathit{save}]\mathit{Saved}, \\ [\mathit{save}1]\mathit{Mbox}1, \\ [\mathit{save}2]\mathit{Mbox}2, \\ [\mathit{Saved}](\mathit{Mbox}1 \lor \mathit{Mbox}2), \\ [\mathit{Mbox}1]\mathit{Saved}, \\ [\mathit{Mbox}2]\mathit{Saved} \end{array}\right\}$$

It is easy to verify that with this representation we would get that an execution of *save*1 can produce a change in *Mbox*2. As far as we are concerned, the only way of avoiding this is deriving a frame axiom at inference time, at the price of not having a satisfactory solution neither to the mathematical frame problem nor to the inferential one.

In the next section we present our logical framework for reasoning about actions, which is not fluent-indexed.

5 The framework of $\mathcal{LAP}_{\mathcal{D}}$

In this section we develop an action-indexed logical framework for reasoning about actions based on the logic of actions and plans $\mathcal{LAP}$ [1] combined with the notion of dependence relations.

5.1 The base logic $\mathcal{LAP}$

$\mathcal{LAP}$ is a simple multimodal logic where formulae are constructed in the following way: We use an $S4$ operator $\Box$ to characterize laws (static or dynamic ones). $\Box A$ is read "henceforth A" or "always A". We use a collection of K operators $[\alpha]$, one for each action α, in order to state the behavior of actions. $[\alpha]C$ is read as "after executing α, C".

Given α an action and A, C classical propositional formulae, the formula $\Box\langle\alpha\rangle\top$ is read as "α is executable". $\Box(A \rightarrow [\alpha]C)$ as "if A, then after α C". $\Box[\alpha]C$ is an abbreviation for $\Box(\top \rightarrow [\alpha]C)$. For example, in the Yale shooting scenario (YSS) [4], the formula $\Box(\mathit{Loaded} \rightarrow [\mathit{shoot}]\neg\mathit{Alive})$ states that, in every situation (possible world), shooting will kill the victim if the gun is loaded. In the same scenario, the formula $\Box(\mathit{Walking} \rightarrow \mathit{Alive})$ is a static law saying that it is always true that someone who is walking must be alive.

In $\mathcal{LAP}$, every action α has the modal logic K, and the modal operator $\Box$ has logic $S4$. Actions and the $\Box$ operator are linked by an axiom $I(\Box, [\alpha])$ stating that $\Box A \rightarrow [\alpha]A$.

Despite its expressive power, in $\mathcal{LAP}$ the frame problem is not solved. In order to do this, we augment $\mathcal{LAP}$ with a metalogical causal information represented by a dependence relation.

5.2 Causality expressed by dependence

We present here our causal notion based on a dependence relation capable of capturing the *contexts* in which actions are executed. We define a ternary dependence relation involving *actions*, *literals* and *formulae*[7]. The role of the latter formulae is to characterize the particular circumstances in which the actions may influence the truth values of literals.

Saying that a literal L *depends* on a certain action α in a given *context* C means that if C is true, then, after the execution of α, L *may* be caused. We shall say that α may cause L in the context C and this will be represented by the expression α may cause L if C.

Consider the action *shoot* and the literals *Alive* and *Loaded*. As the action *shoot* may cause $\neg$*Alive* in a circumstance where *Loaded* is true, then the expression

$$\mathit{shoot}\ \textsf{may cause}\ \neg \mathit{Alive}\ \textsf{if}\ \mathit{Loaded}$$

must be in our contextual dependence relation.

An important point to be highlighted here is the fact that the notion of dependence only *permits* change, and does not necessarily *cause* it. This can be better seen in nondeterministic domains, like the Russian turkey scenario [13], where, after shooting, the victim might die, and might as well keep on being alive.

Definition 5.1 A *contextual dependence relation* is a ternary relation $\mathcal{D} \subseteq ACT \times LIT \times PFOR$.

The triples (α, L, C) will be written α may cause L if C and represent the fact that "L *may* be caused by α in the context C". In other words, this means that "the execution of action α *may* change the truth value of the literal L, as long as the formula C is true".

Remark 5.1 Disjunctive contexts seem to be rare. We conjecture that, in practice, the formula denoting the context C will in general be a *conjunction* of literals. Anyway, the expression α may cause L if $C_1 \vee C_2$ could be substituted by α may cause L if C_1 and α may cause L if C_2. Therefore we can suppose w.l.o.g. that contexts are conjunctions of literals.

[7]Formulae of Classical Propositional Logic, without modal operators.

5.3 A new logic of actions and plans

Combining the logic of actions and plans $\mathcal{LAP}$ with the new dependence relation $\mathcal{D}$ defined so far, we obtain the new logic $\mathcal{LAP}_{\mathcal{D}}$. The $\mathcal{LAP}$-models must satisfy that whenever all the contexts in which an action may cause a given literal L are false, then the falsehood of L must be preserved along the execution of that action.

As an example, consider the action *shoot* and the literal $\neg$*Alive* in the YSS. The only way of *shoot* causing $\neg$*Alive* is when *Loaded* is true. Hence $\mathcal{D} = \{$*shoot* may cause $\neg$*Alive* if *Loaded*$\}$. Thus, in a circumstance in which we have $\neg$*Loaded*, the persistence of *Alive* will be guaranteed by the falsehood of the context *Loaded*.

For the same scenario, consider the action *wait* and the literal *Loaded*. *wait* may never cause $\neg$*Loaded*, whatever the circumstances are. Therefore there will not be in $\mathcal{D}$ any expression of the form *wait* may cause $\neg$*Loaded* if C, for any context C. In this case, we guarantee the persistence of *Loaded* through the execution of *wait*.

With this dependence-based condition, one eliminates the $\mathcal{LAP}$-models in which non-intuitive changes occur in the following way: suppose that we are in a particular situation w in which the literal L is false. First, imagine that the only element of $\mathcal{D}$ involving both α and L is α may cause L if C. Thus, as long as C is true, the execution of α may cause, or not, a change in the truth value of L, since our causal notion only allows change, not forcing it. But surely α will not change the value of L if C is false. Suppose now that there is no $C \in PFOR$ such that α may cause L if $C \in \mathcal{D}$. Then, the execution of α may never make L true, and hence L will still be false after α.

Definition 5.2 Let $\mathcal{D}$ be a ternary dependence relation. We define a model of $\mathcal{LAP}_{\mathcal{D}}$ as a $\mathcal{LAP}$-model $\mu = \langle W, \{R_\alpha : \alpha \in ACT\}, R_\Box, \tau\rangle$, such that whenever $wR_\alpha w'$ then for every α and for every $L \in LIT$, if, for all $C \in PFOR$ such that α may cause L if C, $w \not\models C$, then $w \in \tau(L)$ only if $w' \in \tau(L)$ and $w' \notin \tau(L)$ only if $w \notin \tau(L)$.

Given a dependence relation $\mathcal{D}$, we say that a formula A is true in a $\mathcal{LAP}_{\mathcal{D}}$-model $\mu = \langle W, \{R_\alpha : \alpha \in ACT\}, R_\Box, \tau\rangle$ if $w \models A$ (A is true in w) for every $w \in W$. A is $\mathcal{LAP}_{\mathcal{D}}$-valid (noted $\models_{\mathcal{LAP}_{\mathcal{D}}} A$) if A is true in all $\mathcal{LAP}_{\mathcal{D}}$-models.

5.4 Avoiding conditional frame axioms

An important criticism to the dependence-based approach of $\mathcal{LAP}_{\leadsto}$ defined in [1] is the need for writing down *conditional frame axioms.* As an example, the formula below is needed for correctly dealing with the YSS:

$$\Box((\neg Loaded \land Alive) \rightarrow [shoot]Alive) \tag{17}$$

Formulae of the type of (17) establish that if a given condition is true, then some literal persists along the execution of a given action. Without considering such conditional frame axioms, it is not possible to derive the intended conclusions in $\mathcal{LAP}_{\leadsto}$.

In what follows we show that with the ternary dependence relation $\mathcal{D}$ we do not need to state any conditional frame axiom.

Definition 5.3 Let $\alpha \in ACT$, $L \in LIT$ and $\mathcal{D}$ a dependence relation. We define

$$Pre_{\mathcal{D}}(\alpha, L) = \bigvee\{C : \alpha \textsf{ may cause } L \textsf{ if } C \in \mathcal{D}\}$$

In other words, $Pre_{\mathcal{D}}(\alpha, L)$ is the disjunction of all the contexts in which α may cause literal L, given a dependence relation $\mathcal{D}$.

Theorem 5.1 $\models_{\mathcal{LAP}_{\mathcal{D}}} \Box((\neg Pre_{\mathcal{D}}(\alpha, L) \land \neg L) \rightarrow [\alpha]\neg L)$.

Proof: Suppose that $\Box((\neg Pre_{\mathcal{D}}(\alpha, L) \land \neg L) \rightarrow [\alpha]\neg L)$ is false, i.e. there is a possible world w such that $w \models \neg Pre_{\mathcal{D}}(\alpha, L)$ and $w \models \neg L$, and it is not the case that $w \models [\alpha]\neg L$, that is to say $w \models \langle\alpha\rangle L$. Suppose now that α is executable, at least when $\neg Pre_{\mathcal{D}}(\alpha, L)$ and $\neg L$ are true in w. Then there is a possible world w' such that $wR_{\alpha}w'$ and $w' \models L$. As $w \models \neg Pre_{\mathcal{D}}(\alpha, L)$, we have that for all expression α may cause L if C in $\mathcal{D}$, $w \not\models C$, and as $w \models \neg L$, by the definition of $\mathcal{D}$, we shall have $w' \models \neg L$, which is an absurd. ■

With this result, we can see that in a domain description using a dependence relation $\mathcal{D}$ there is no need for a set of conditional frame axioms, since all the conclusions that are obtained with the aid of the latter can also be inferred with the former.

As an example, consider the action *shoot* and the literals *Alive* and $\neg$*Loaded*, and suppose $\mathcal{D} = \{shoot \textsf{ may cause } \neg Alive \textsf{ if } Loaded\}$. Then the conditional frame axiom (17) is $\mathcal{LAP}_{\mathcal{D}}$-valid. In other words, the persistence of *Alive* when $\neg$*Loaded* is true follows from the dependence information in $\mathcal{D}$, making completely unnecessary, thus, the statement of the conditional frame axiom (17).

5.5 Axiomatics and complexity

Given a dependence relation $\mathcal{D}$, we axiomatize the class of $\mathcal{LAP}_{\mathcal{D}}$-models in the same way as done for $\mathcal{LAP}$ in [1, Section 4.2], adding an axiom scheme founded on the dependence relation:

- $\mathit{Persist}([\alpha]) : \neg L \rightarrow [\alpha]\neg L$, if $\neg \mathit{Pre}_{\mathcal{D}}(\alpha, L)$.

In order to show the soundness and completeness of $\mathcal{LAP}_{\mathcal{D}}$ with respect to its semantics, we have the following result:

Theorem 5.2 For all dependence relation $\mathcal{D}$ over $\mathit{ACT} \times \mathit{LIT} \times \mathit{PFOR}$, the axiomatics of $\mathcal{LAP}_{\mathcal{D}}$ is sound and complete with respect to the class of $\mathcal{LAP}_{\mathcal{D}}$-models.

Proof: The proof is in [17]. ∎

The theorem below shows that the complexity of $\mathcal{LAP}_{\mathcal{D}}$ is the same as that of $\mathcal{LAP}$.

Theorem 5.3 $\mathcal{LAP}_{\mathcal{D}}$ is decidable, and the satisfiability problem in $\mathcal{LAP}_{\mathcal{D}}$ is EXPTIME-complete.

Proof: The proof is in [17]. ∎

This theorem guarantees that the inclusion of the ternary dependence relation $\mathcal{D}$ in $\mathcal{LAP}$ does not increase the computational complexity of the basic logic.

5.6 The Mailboxes Scenario in $\mathcal{LAP}_{\mathcal{D}}$

We present here the representation of the Mailboxes Scenario in the formalism of $\mathcal{LAP}_{\mathcal{D}}$. *KB* represents the set of observations and *LAW* that of static, effect and executability laws in $\mathcal{LAP}_{\mathcal{D}}$.

$$\mathit{LAW} = \left\{ \begin{array}{l} \Box\langle \mathit{save}\rangle\top, \Box\langle \mathit{save}1\rangle\top, \Box\langle \mathit{save}2\rangle\top, \\ \Box(\mathit{Saved} \leftrightarrow \mathit{Mbox}1 \vee \mathit{Mbox}2), \\ \Box[\mathit{save}]\mathit{Saved}, \\ \Box[\mathit{save}1]\mathit{Mbox}1, \\ \Box[\mathit{save}2]\mathit{Mbox}2 \end{array} \right\}$$

$$\mathcal{D} = \left\{ \begin{array}{c} \mathit{save}\ \mathsf{may\ cause}\ \mathit{Saved}\ \mathsf{if}\ \top, \\ \mathit{save}\ \mathsf{may\ cause}\ \mathit{Mbox}1\ \mathsf{if}\ \top, \\ \mathit{save}\ \mathsf{may\ cause}\ \mathit{Mbox}2\ \mathsf{if}\ \top, \\ \mathit{save}1\ \mathsf{may\ cause}\ \mathit{Mbox}1\ \mathsf{if}\ \top, \\ \mathit{save}2\ \mathsf{may\ cause}\ \mathit{Mbox}2\ \mathsf{if}\ \top, \\ \mathit{save}1\ \mathsf{may\ cause}\ \mathit{Saved}\ \mathsf{if}\ \top, \\ \mathit{save}2\ \mathsf{may\ cause}\ \mathit{Saved}\ \mathsf{if}\ \top \end{array} \right\}$$

$$KB = \{\neg Saved, \neg Mbox1, \neg Mbox2\}$$

With this representation, we can conclude

$$\models_{\mathcal{LAP}_{\mathcal{D}}} \quad (KB \land LAW) \to [save](Mbox1 \lor Mbox2)$$

as intended.

5.7 Designing dependence relations

In this section we propose an interactive algorithm for aiding the knowledge engineer to generate a dependence relation for a given domain description.

Let $\mathcal{E}$ be a set of effect laws, $\mathcal{S}$ a set of domain constraints, and let $\mathcal{S}^{\Box} = \{A : \Box A \in \mathcal{S}\}$. $NewCons_A(B)$ is a function that computes the set of strongest clauses that follow from $A \land B$, but do not follow from A alone (cf. e.g. [6]). The following algorithm generates a dependence relation:

Algorithm 5.1 (Generating dependence information)

```
input: $\mathcal{S}, \mathcal{E}$
output: a dependence relation $\mathcal{D}$
  $\mathcal{D} := \emptyset$
  for all $\Box(A \to [\alpha]C) \in \mathcal{E}$ do
    for all $C_i$ in $C$ do
      for all $B \in NewCons_{\mathcal{S}^{\Box}}(C_i)$ do
        for all $L$ in $B$ do
          $\gamma :=$ context in which $\alpha$ causes $L$
          $\mathcal{D} := \mathcal{D} \cup \{\alpha$ may cause $L$ if $\gamma\}$
```

Example 5.1 (The Mailboxes Scenario) The input of algorithm 5.1 is

$$\mathcal{E} = \left\{ \begin{array}{l} \Box[save]Saved, \\ \Box[save1]Mbox1, \\ \Box[save2]Mbox2 \end{array} \right\}$$

$$\mathcal{S} = \{\Box(Saved \leftrightarrow Mbox1 \vee Mbox2)\}$$

We only give the case of $\Box[save]Saved$. First of all, one computes the relevant prime implicates of $Saved \wedge Saved \leftrightarrow Mbox1 \vee Mbox2$, which is $\{Saved, Mbox1 \vee Mbox2\}$. After that, we ask for the contexts in which *save* may cause *Saved*. Intuitively, the user should answer $\top$, so we add the dependence *save* may cause *Saved* if $\top$ to $\mathcal{D}$. The next round, we ask about the context in which *save* may cause *Mbox*1. Again the user answers $\top$ and we add the dependence *save* may cause *Mbox*1 if $\top$. Analogously we obtain *save* may cause *Mbox*2 if $\top$.

6 Conclusions

We have presented a typical scenario involving actions with both indeterminate and indirect effects and seen the difficulties that arise when we try to formalize it with fluent-indexed approaches.

The problem with all these formalisms is that in this scenario there is a fluent (*Saved*) that can be caused in two different ways (directly with *save* or indirectly with *save*1 or *save*2) and that can or cannot cause nondeterministic ramifications depending on the way it was generated. With fluent-indexed approaches we cannot record this subtlety and this is the main reason they all fail in formalizing this example.

The problems shown in Sections 4.2 and 4.3 arise because the inertial/non-inertial classification, for this scenario, depends on the action that is executed. With the approach presented in [9, 10] we cannot capture this property, for causality is fluent-indexed.

So, with all this discussion, we have seen that with the approaches presented in [7, 8, 9, 10, 15] we cannot deal with indirect and implicit indeterminate effects without referring to actions. This supports the thesis that causality must be action indexed and motivated us to define our action-based framework.

In the $\mathcal{LAP}_{\mathcal{D}}$ representation of the Mailboxes Scenario, Postulate P3 is satisfied. Moreover, as we neither need to explicitly state non-effects of actions nor relate actions with their indirect effects, Postulates P1 and P2 are satisfied, too. Nevertheless, we still need to write down indirect dependences, e.g. *save*1 may cause *Saved* if $\top$, in order to be able to infer ramifications. At one hand, doing things this way will require us to write more information than commonly used, but we argue this is the only way of avoiding unintended conclusions, at least for the known classes of problems.

The interactive algorithm proposed for aiding the conception of domain descriptions can be useful for realistic scenarios. A disadvantage of its implementation, however, resides in the fact that the degree of interaction with the user is too high, in the sense that too many possibilities must be evaluated in order to generate the good dependences.

With the tableau method for $\mathcal{LAP}_{\leadsto}$ given in [1] and the translation of $\mathcal{LAP}_{\mathcal{D}}$ into $\mathcal{LAP}_{\leadsto}$ presented in [17], we automatically have a proof procedure for domain descriptions in $\mathcal{LAP}_{\mathcal{D}}$.

Acknowledgements

Ivan Varzinczak has been supported by a fellowship from the government of the Federative Republic of Brazil. Grant: CAPES BEX 1389/01-7.

References

[1] M. A. Castilho, O. Gasquet, and A. Herzig. Formalizing action and change in modal logic I: the frame problem. *J. of Logic and Computation*, 9(5):701–735, 1999.

[2] M. A. Castilho, A. Herzig, and I. J. Varzinczak. It depends on the context! a decidable logic of actions and plans based on a ternary dependence relation. In S. Benferhat and E. Giunchiglia, editors, *Proc. Int. Conf. on Non-Monotonic Reasoning (NMR'02)*, pages 343–348, Toulouse, 2002.

[3] J. J. Finger. *Exploiting constraints in design synthesis.* PhD thesis, Stanford University, Stanford, 1987.

[4] S. Hanks and D. McDermott. Default reasoning, nonmonotonic logics, and the frame problem. In *Proc. 5th Nat. Conf. on Artificial Intelligence (AAAI'86)*, pages 328–333, Philadelphia, 1986. Morgan Kaufmann Publishers.

[5] D. Harel. Dynamic logic. In D. M. Gabbay and F. Günthner, editors, *Handbook of Philosophical Logic*, volume II, pages 497–604. D. Reidel, Dordrecht, 1984.

[6] K. Inoue. Linear resolution for consequence finding. *Artificial Intelligence*, 56(2–3):301–353, 1992.

[7] F. Lin. Embracing causality in specifying the indirect effects of actions. In Mellish [12], pages 1985–1991.

[8] F. Lin. Embracing causality in specifying the indeterminate effects of actions. In *Proc. 13th Nat. Conf. on Artificial Intelligence (AAAI'96)*, pages 670–676, Portland, 1996. AAAI Press/MIT Press.

[9] N. McCain and H. Turner. A causal theory of ramifications and qualifications. In Mellish [12], pages 1978–1984.

[10] N. McCain and H. Turner. Causal theories of action and change. In *Proc. 14th Nat. Conf. on Artificial Intelligence (AAAI'97)*, pages 460–465, Providence, 1997. AAAI Press/MIT Press.

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

[12] C. Mellish, editor. *Proc. 14th Int. Joint Conf. on Artificial Intelligence (IJCAI'95)*, Montreal, 1995. Morgan Kaufmann Publishers.

[13] E. Sandewall. *Features and Fluents.* Oxford University Press, 1994.

[14] E. Sandewall. Assessments of ramifications methods that use static domain constraints. In L. C. Aiello, J. Doyle, and S. Shapiro, editors, *Proc. 6th Int. Conf. on Knowledge Representation and Reasoning (KR'96)*, pages 99–110, Cambridge, MA, 1996. Morgan Kaufmann Publishers.

[15] M. Thielscher. Computing ramifications by postprocessing. In Mellish [12], pages 1994–2000.

[16] M. Thielscher. Ramification and causality. *Artificial Intelligence*, 89(1–2):317–364, 1997.

[17] I. J. Varzinczak. Causalidade e dependência em raciocínio sobre ações. Master's thesis, Depto. Informática — Universidade Federal do Paraná, Curitiba, 2002.

[18] D. Zhang and N. Y. Foo. EPDL: A logic for causal reasoning. In B. Nebel, editor, *Proc. 17th Int. Joint Conf. on Artificial Intelligence (IJCAI'01)*, pages 131–138, Seattle, 2001. Morgan Kaufmann Publishers.