Ramification and Causality in a Modal Action Logic

Laura Giordano
Dipartimento di Scienze e Tecnologie Avanzate
Università del Piemonte Orientale "A. Avogadro"
Corso Borsalino 54 - 15100 Alessandria, ITALY.
e-mail: laura@di.unito.it

Alberto Martelli
Dipartimento di Informatica - Università di Torino
C.so Svizzera 185 - 10149 Torino, ITALY.
e-mail: mrt@di.unito.it

Camilla Schwind
Faculté des Sciences de Luminy
Laboratoire d'Informatique de Marseille, CNRS
16 Avenue de Luminy, Case 901
13288 Marseille Cedex 9, FRANCE
e-mail: schwind@lim.univ.mrs.fr

July 26, 1999

Abstract

The paper presents a logic for action theory based on a modal language, where modalities represent actions. The frame problem is tackeled by using a nonmonotonic formalism which maximizes persistency assumptions. The problem of ramification is tackled by introducing a modal causality operator which is used to represent causal rules. Assumptions on the value of fluents in the initial state allow to reason with incomplete initial states and to do postdiction. The action theory can also deal with non-minimal change and nondeterministic actions.

1 Introduction

Reasoning about action and change is one of the main topics which must be addressed in building intelligent agents. Among the various approaches to reasoning about actions, one of the most popular is still the situation calculus. The situation calculus represents states of the world (*situations*) as sequences of *actions*, and *fluents* are relations whose truth values vary from state to state. The situation calculus is formulated in first-order logic: Situations are represented with terms such as $do(a_1, do(a_2, s))$, while fluents are extended with an extra argument denoting a situation.

In this paper we show that a similar approach can be profitably formulated using modal logic. The suitability of modal logic for reasoning about actions has already been pointed out by other authors [8, 36, 3, 41], that make use of Dynamic Logic [23]. As observed in [33], classical dynamic logic adopts essentially the same ontology as McCarthy's situation calculus, by taking "the state of the world as primary, and encoding actions as transformations on states". Indeed, actions can be represented in a natural way by modalities, and states as sequences of modalities.

In this paper we make use of a small fragment of dynamic logic. We represent a primitive action a by a modal operator $[a]$, and a sequence of actions $a_1, a_2, \ldots, a_n$ by the modal operator $[a_1; a_2; \ldots; a_n]$ ($[\varepsilon]$ represents the empty sequence of actions, i.e. the initial state). Furthermore we make use of a modality $\Box$ to represent any arbitrary sequence of actions and to denote those formulas which *always* hold. For instance, the *Yale shooting problem* could be formulated as follows.

Example 1
$(i)\Box[load]loaded$
$(ii)\Box(loaded \rightarrow [shoot]\neg alive)$
$(iii)\Box[shoot]\neg loaded$

Although the representation with modal logic and the one with first-order logic are apparently similar, it is important to point out a significant difference between the two. In fact, if we do not assume any particular property for the modal operators representing actions (and, in particular, they are not serial), the two formulas $\neg[s]\phi$ and $[s]\neg\phi$ have different meanings, whereas in the situation calculus both would be represented by $\neg\phi(s)$. Thus, differently from the situation calculus, we cannot derive $\neg loaded$ from (ii) and $[shoot]alive$, i.e. action rules cannot be used contrapositively. We will see in a moment that this difference becomes crucial when dealing with the frame problem.

Reasoning about actions requires to solve the *frame problem*, i.e. the problem of specifying in a concise way what does not change when an action is executed. Several solutions to this problems have been proposed in the framework of the situation calculus. The main idea is to minimize change from one state to the next one, by making use of circumscription and, in special cases, of completion constructions. For instance, Reiter [38] has shown how to derive from effect axioms a set of *successor state axioms*, describing the value of each fluent after the execution of any possible action.

In this paper we adopt a nonmonotonic approach, by adding *persistency assumptions* of the form:

$$[a_1; a_2; \ldots; a_n](l \rightarrow [a]l).$$

The basic idea is that persistency of a fluent from a state to the next one must be assumed, if it does not lead to an inconsistent state. Therefore the above persistency assumption must be assumed unless we are able to prove that $\neg l$ holds after execution of action a, i.e. unless we are able to prove $[a_1; a_2; \ldots; a_n; a]\neg l$.

Let us consider the shooting problem, and let us assume that *loaded* and *alive* hold in the inital state, i.e.:

$(iv)[\epsilon]loaded$

$(v)[\epsilon]alive$

and that the two actions *wait* and *shoot* are executed. It is possible to assume persistency of the two fluents through *wait*, i.e.

$(vi)[\varepsilon](alive \to [wait]alive)$

$(vii)[\varepsilon](loaded \to [wait]loaded)$

because it does not lead to an inconsistency. From the above formulas, it is possible to derive

$[wait; shoot]\neg alive$

which blocks persistency of *alive* through *shoot*.

On the other hand we can show that we cannot obtain the unwanted solution where the gun becomes unloaded during *wait*, and *alive* persists through *shoot*. To this regard, it is essential that action modalities are not serial, so that action laws cannot be used contrapositively. In particular, assuming that alive persists after the shooting, from $[wait][shoot]alive$ and from the action law (ii) we cannot, anyhow, conclude $[wait]\neg loaded$, and thus block persistence of *loaded* through *wait*. In fact, the contrapositive formula

$$[wait](\neg[shoot]\neg alive \to \neg loaded),$$

which follows from the action law above, cannot be applied, since $[wait][shoot]alive$ is not equivalent to $[wait]\neg[shoot]\neg alive$.

Notice that the above mentioned difference between our modal approach and the situation calculus is crucial here. In fact, in the situation calculus, rule (ii) can be formulated as:

$loaded(s) \to \neg alive(do(shoot, s))$.

By assuming persistence of *alive* through *wait* and *shoot* ,we get:

$alive(do(shoot, do(wait, s0)))$,

from which, by using the contrapositive of the previous formula, we get:

$\neg loaded(do(wait, s0))$,

which blocks persistence of *loaded* through the action *wait*.

Therefore, in situation calculus the simple approach of this paper based on maximizing persistency does not work, and more sophisticated techniques have to be adopted in order to enforce chronological minimization of change. Furthermore, the approach we propose is quite general, since complex formulas (including action modalities) are allowed in action preconditions, which is not true for most of the proposal, as we will see in section 4.

The previous example assumed complete knowledge of the initial state, which was used to reason forward. In our language however it is also possible to reason with incomplete initial states to do *postdiction*, i.e. to find the value of fluents in the inital state which explain observations made in later states. This can be easily achieved by extending the assumption based approach we use for dealing with persistence, by considering also truth values of fluents in the inital state as assumptions. For instance, the observation $[wait; shoot]alive$ in the shooting problem can be explained by assuming:

$$[\epsilon]\neg loaded$$

$$[\epsilon]alive.$$

One of the most challenging problems in reasoning about actions is *ramification*, i.e. the problem of dealing with indirect effects of actions. Most approaches today try to incorporate causality to express indirect effects [5, 30, 28, 29, 45]. The concept of causality has been widely studied in philosophy, and several theories have been proposed.

In our case we want to be able to express rules such as

$$\neg alive \ \textbf{causes} \ \neg walking$$

for the shooting problem. Thus, if the gun is initially loaded, and Fred is alive and walking, after *shoot*, $\neg alive$ will hold, and $\neg walking$ as well, according to the above causal rule.

In general we do not want to represent causal rules with material implication because it has undesired properties such as contraposition. In our example we do not want to use the causal rule to derive $alive$ from $walking$, because we do not expect the truth value of the fluent $alive$ to be influenced by the one of $walking$.

Thus any approach to causality should avoid using contrapositives of causal rules. This can be achieved in many ways, for instance by representing causal rules as inference rules. However we have already seen that modal logic provides a simple solution to this problem, since rules with a modal operator in the consequent can block contraposition. Thus, to deal with causality, we introduce a new modal operator ⓒ, and the above causal rule will be represented in our language as

$$\neg alive \rightarrow \text{ⓒ}\neg walking.$$

The introduction of this feature does not require any change in the previously described nonmonotonic approach to persistence.

Our approach is similar to the one adopted by Lin [28], who introduces in the situation calculus a metapredicate "caused". As before, the main difference between the two approaches is the possibility given by the situation calculus of using contrapositives, which sometimes may give unexpected solutions. Section 4 presents an example where the two approaches behave differently, and compares our approach to other approaches to causality.

The introduction of frame axioms enforces minimal change of fluent truth values. However, there are cases when we want to avoid minimal change. To achieve this, we introduce two kinds of fluents, *frame* and *nonframe fluents*, similarly to Kartha and Lifschitz [24]. However, differently from their proposal, in our case the property of being "frame" or "nonframe" depends on the action which is executed.

This approach allows to deal easily with nondeterministic actions. For instance, if we want to model the Russian turkey problem [39], we can extend the shooting domain by adding a new action of spinning the gun's bullet chamber, which nondeterministically makes the gun loaded or unloaded. With respect to the action *spin*, we will define *loaded* to be a "nonframe" fluent. Indeed, we do not expect *loaded* to persist after spinning.

If we perform the action *load* and then *spin*, the gun may result to be loaded or unloaded: there are two solutions, both of them containing $[load]loaded$; one contains $[load; spin]loaded$, while the other contains $[load; spin]\neg loaded$. The gun behaves precisely as expected. Since we do not know whether the gun will be loaded after spinning or not, both cases are possible.

A further important problem is that of *qualification*. Although we do not consider it in the formulation of the action rules, in Section 3.6 we will show how to accommodate it in our framework.

In the next section we introduce our modal action logic. In section 3 we will define action theories and their extensions, and we will compare them with default extensions. Section 4 is devoted to a detailed comparison with related work. In section 5 we will drow some conclusions.

2 Action Logic

2.1 The action language

In this section we introduce the propositional action language AD we will use in the paper. The choice for a propositional language is motivated by simplicity.

The action language contains the following modalities: a finite number of modalities $[a_1], \ldots, [a_k]$, where each a_i is a constant denoting a primitive action; a modality $\Box$, which represents any arbitrary sequence of actions, and is used to denote those formulas which always hold; and a modality ⓒ which is used to denote information which is caused as a ramification effect of an action. In particular, the meaning of $\Box\alpha$ is that α holds after any sequence of actions, while the meaning of ⓒα is that α is caused.

As in propositional dynamic logic [23], we also introduce modalities for sequential action composition. We define an action sequence as follows:

$$A ::= \varepsilon \mid A; a$$

where ε represents the empty sequence, A stands for an action sequence, and a for a primitive action. An action sequence can be regarded as representing a state. The intended meaning of a formula $[\varepsilon, a_{j_1}; \ldots; a_{j_k}]\alpha$ is that α holds after performing the actions $a_{j_1}, \ldots, a_{j_k}$ in the order. In the following, when there is no risk of confusion we will write $[a_1; \ldots; a_k]$ rather than $[\varepsilon; a_1; \ldots; a_k]$, as a notational convention.

Properties of the modalities are the following. Modalities $[a_i]$ $[\varepsilon]$ and ⓒ are ruled by the axioms of the logic K, and $\Box$ is ruled by the axioms of the logic S4. We have the following axiom schemas

$$
\begin{array}{ll}
(1) & [a_i](\alpha \to \beta) \to ([a_i]\alpha \to [a_i]\beta) \\
(2) & \copyright(\alpha \to \beta) \to (\copyright\alpha \to \copyright\beta) \\
(3) & [\varepsilon](\alpha \to \beta) \to ([\varepsilon]\alpha \to [\varepsilon]\beta) \\
(4) & \Box(\alpha \to \beta) \to (\Box\alpha \to \Box\beta) \\
(5) & \Box\alpha \to \alpha \\
(6) & \Box\alpha \to \Box\Box\alpha.
\end{array}
$$

Axioms (1), (2), (3) and (4) say that all modalities distribute over implications. They are instances of axiom K, which characterizes all normal modal logics. Moreover, we have an axiom schema for sequential action composition (where A stands for an action sequence)

$$(7) \quad [A;a]\alpha \leftrightarrow [A][a]\alpha,$$

and the following interaction axiom schemas

$$
\begin{array}{ll}
(8) & \Box\alpha \to [a_i]\alpha, \\
(9) & \Box\alpha \to [\varepsilon]\alpha \\
(10) & [a_i]\copyright\alpha \to [a_i]\alpha \\
(11) & [\varepsilon]\copyright\alpha \to [\varepsilon]\alpha.
\end{array}
$$

Let A be an action sequence. From (1), (3) and (7) we can derive $[A](\alpha \to \beta) \to ([A]\alpha \to [A]\beta)$. Moreover, from (6), (7), (8) and (9) we can derive $\Box\alpha \to [A]\alpha$. Finally, notice that, though the modality $\copyright$ is not reflexive, axioms (10) and (11), together with (7), allow $[A]\alpha$ to be inferred from $[A]\copyright\alpha$ (and, in general from $[A]\copyright^n\alpha$).

An axiomatization for the language also includes all the propositional tautologies, and necessitation rules for all the modalities.

These axioms together with the necessitation rule define an inference relation $\vdash_{AD}$ as usual. We denote $Th_{AD}(\Gamma) = \{\phi : \Gamma \vdash_{AD} \phi\}$.

As in classical logic, we say that a set of formulas Γ is *inconsistent* if there is a finite set $\{\alpha_1, \alpha_2, \ldots \alpha_n\} \subseteq \Gamma$ such that $\vdash_{AD} (\neg\alpha_1 \vee \neg\alpha_2 \vee \neg \ldots \vee \neg\alpha_n)$, which is equivalent to $\Gamma \vdash_{AD} \bot$. Otherwise, Γ is called *consistent*. We also will use the notion of "(in)consistency after action sequence", namely a set is $[A]-$inconsistent, if $\Gamma \vdash_{AD} [A]\bot$. Since the action modalities are not serial, an $[A]-$inconsistent set is not inconsistent in general. Moreover, this notion is useful for action theories, since it allows to represent actions which cannot be executed, and therefore to produce an empty set of resulting states (see chapter 3.6).

2.2 Semantics

The semantics of our causal action language is standard Kripke semantics. A Kripke interpretation is a tuple $M = \langle W, R_{a_1}, \ldots, R_{a_k}, R_{\Box}, R_{\copyright}, R_{\varepsilon}, V\rangle$, where W is a nonempty set of worlds; V is an assignement function, which assigns a truth value to each propositional letter at each world in W; $R_{a_1}, \ldots, R_{a_k}$, $R_{\Box}$, $R_{\copyright}$ and R_{ε} are the accessibility relations associated with the modalities $[a_1], \ldots, [a_k]$, $\Box$, $\copyright$ and ε. In particular, every action $[a_i]$ is associated with a binary transition relation R_{a_i}, while $\copyright$ is associated with a binary "causality relation" $R_{\copyright}$.

The accessibility relations R_a, R_ε and $R_{\copyright}$ are not required to satisfy any special property, as, for instance, reflexivity or seriality. Hence, it may be the case that, from a given world w, there is no world accessible by R_a (or $R_{\copyright}$, or R_ε).

As in dynamic logic, the accessibility relation for the sequence of actions can be defined as follows:

$$R_{A;a} = R_A \circ R_a,$$

where $\circ$ is the composition operation, that is, $R_A \circ R_a = \{(w_1, w_3) : (w_1, w_2) \in R_A \text{ and } (w_2, w_3) \in R_a\}$.

By axioms (5) and (6) $R_\Box$ is required to be a reflexive and transitive relation.

By axiom (8), any state reachable by an action transition relation R_a is also reachable by the relation $R_\Box$, i.e. $R_\Box \supseteq R_a$. Similarly, by axiom (9), $R_\Box \supseteq R_\varepsilon$. Since the relation $R_\Box$ is transitive, $R_\Box$ relates any given world to all the worlds reachable from it by any number of action transitions R_{a_i}.

Axiom (10) requires that

$$R_a \subseteq R_a \circ R_{\copyright},$$

for each primitive action a: given two worlds x and y, if $(x, y) \in R_a$, there is a world z such that $(x, z) \in R_a$ and $(z, y) \in R_{\copyright}$. Hence, the states resulting from the execution of an action a include the states reachable by executing action a and, then, by applying causal rules. This is intended to capture the fact that the effects of an action a are those which are obtained as direct consequence of the action and those which come after as "causal ramifications". A similar property holds by axiom (11), namely $R_\varepsilon \subseteq R_\varepsilon \circ R_{\copyright}$.

The *satisfiability* of a formula α at a given world w in a Kripke interpretation M (written $M, w \models \alpha$) is defined as usual in modal logics. In particular,

- $M, w \models \alpha$, for an atomic proposition α, iff $V(w, \alpha) = true$;
- $M, w \models [a_i]\alpha$, iff, for all $w' \in W$ such that $(w, w') \in R_{a_i}$, $M, w' \models \alpha$.

A similar definition is given for all the other modalities, by making use of the associated accessibility relation.

We say that a formula α is *satisfiable* if there are an interpretation $M = \langle W, R_{a_1}, \ldots, R_{a_k}, R_\Box, R_{\copyright}, R_\varepsilon, V\rangle$ and a world $w \in W$ such that $M, w \models \alpha$. We say that a formula α is *valid* (and we write $\models \alpha$) if $\neg\alpha$ is not satisfiable.

Given an action sequence A and a state $w \in W$ in a model M, we define $R_A(w) = \{w' : (w, w') \in R_A\}$. It is then easy to see that $R_A(w) = \emptyset$ whenever $M, w \models [A]\bot$.

Soundness and completeness of the axiom system with respect to this Kripke semantics can be easily proved by standard techniques (see, for instance, [22]).

3 Action Theories

In our language we will use atomic propositions for *fluent names*, and l to denote a *fluent literal*, consisting of a fluent name f or its negation $\neg f$. Given a fluent literal l, such that $l = f$ or $l = \neg f$, we define $|l| = f$. Moreover, we will denote by $\mathcal{F}$ the set of all fluent names, by *Lit* the set of all fluent literals, and by $\mathcal{A}$ the set of all primitive actions.

We define a *domain description* as a tuple $(\Pi, Frame, Obs)$. Π is a set of *action laws* and *causal rules*, that we will define below. *Frame* is a set of pairs (f, a), where $f \in \mathcal{F}$ is a fluent and $a \in \mathcal{A}$ is an action, meaning that f is a frame fluent for action a, that is, f is a fluent to which persistency applies when action a is executed. On the other hand, those fluents which are not frame with respect to a do non persist and may change in a nondeterministic way, when executing a. For instance, in example 1 in the introduction all fluents are frame with respect to all actions. However, in the Russian turkey problem *loaded* is defined to be a "nonframe" fluent with respect to the action *spin*, of spinning the gun's bullet chamber, which nondeterministically makes the gun loaded or unloaded. In fact, we do not expect *loaded* to persist after spinning.

Obs is a set of observations, about the value of fluents in different states. They have the form $[a_1; \ldots; a_j]\alpha$, meaning that α holds after executing the action sequence $a_1, \ldots, a_j$. In particular, *Obs* may contain observations about fluents in the initial state ($[\varepsilon]\alpha$). Sometimes, when we do not want to consider observations, we will use the notion of *domain frame*, which is a pair $(\Pi, Frame)$.

Let us define action laws and causal rules. *Action laws* have the form:

$$\Box(\pi \rightarrow [a]\rho)$$

for every primitive action a with precondition π and effect ρ: executing action a in a state where π holds causes ρ to hold. An action law with no precondition turns out to be $\Box[a]\rho$.

Causal rules have the form:

$$\Box(\beta \rightarrow \copyright\alpha),$$

with the meaning that "if β holds, then α is caused". They are intended to expresses "causal" dependencies among fluents, and, intuitively, their directionality makes them similar to inference rules: if we are able to derive β then we can conclude α. The usefulness of causality has been widely recognized in the literature [5, 30, 28, 45]. Indeed, directionality of causal rules provides information on *how* dependencies among fluents can be restored when they are possibly violated.

While we postpone a deeper discussion about this issue to later sections, we may observe that, in the case when the antecedent of a causal rule is empty (that is, it is the formula *true*) the causal rule above becomes $\Box\copyright\alpha$, where α is an arbitrary formula. From such a formula, by axioms (6), (7), (8) and (9) we get that $[A]\copyright\alpha$, for all action sequences A, and, by axioms (7), (10) and (11), we can derive $[A]\alpha$. Hence, if $\beta \equiv true$ in a causal rule, we can conclude that α has to hold in every state. Thus, as a special case of our causal rules we are able to represent domain constraints of the form **always** α, as those introduced in [24] [1]. Hence in our approach causal rules without antecedent can be regarded as constraints: if α is caused under any condition in any circumstance, α is always true. Therefore causal rules without antecedent as

$$\Box\copyright\alpha,$$

have the meaning that "α always holds". In the following they will be called *domain constraints*.

[1] The fact that, by making use of general causal rules, both ramification and qualification constraints can be defined, was already observed by McCain and Turner in [30].

As an example of domain description consider the following formalization of the *shooting problem*.

Example 2 *Shooting problem.* We define a domain description containing the following set of action laws and observations on the initial state (we will assume the set of actions $\mathcal{A} = \{load,\ wait,\ shoot\}$ and $\mathcal{F} = \{loaded,\ alive\}$):

Π: $\Box[load]loaded$ $\quad\Box(loaded \rightarrow [shoot]\neg alive)$
$\Box[shoot]\neg loaded$
Obs: $[\varepsilon](alive \wedge \neg loaded)$
Frame$= \{(l, a) : a \in \mathcal{A}, l \in \mathcal{F}\}$.

In this example the set Π only contains action laws, and all fluents are frame fluents.

As mentioned above, we can model the Russian turkey problem, by adding to the "Shooting domain" a nondeterministic action *spin*.

Example 3 There is a gun and two actions *load* and *spin*. After loading, the gun is *loaded* and after *spin* we do not know if it is loaded or not. To model this, we omit saying that *loaded* is persistent with respect to *spin*. Then (taking $\mathcal{A} = \{load,\ wait,\ shoot,\ spin\}$) we have the following domain description $D = (\Pi, Frame, Obs)$, where Π is as in Example 2:

Obs: $[\varepsilon]\neg loaded$
$Frame$$= \{(f, a) : a \in \mathcal{A}, f \in \mathcal{F}\} \setminus \{(loaded, spin)\}$.

Notice that there is no action law for spin.

As a further example, consider the *suitcase problem* presented in [28]. It illustrates the ramification problem, which is concerned with the additional effects produced by an action besides its immediate effects.

Example 4 There is a suitcase with two locks and a spring loaded mechanism which will open the suitcase when both of the locks are in the up position. We have the following domain description:

Π: $\Box(\neg up_1 \rightarrow [flip_1]up_1)$ $\quad\Box(up_1 \rightarrow [flip_1]\neg up_1)$
$\Box(\neg up_2 \rightarrow [flip_2]up_2)$ $\quad\Box(up_2 \rightarrow [flip_2]\neg up_2)$
$\Box(up_1 \wedge up_2 \rightarrow ⓒopen)$
Obs: $[\varepsilon](\neg up_1 \wedge \neg up_2 \wedge \neg open)$
Frame$= \{(l, a) : a \in \mathcal{A}, l \in \mathcal{F}\}$.

The first four rules in Π describe the immediate effects of the action of flipping the status of a lock, whereas the fifth rule is a causal rule which describes the effect of the spring loaded mechanism. All the fluents are regarded as being persistent. In particular, we want the fluent *open* to persist: when the suitcase becomes open, we expect it to persist in this state even if the locks are turned down.

As in [26, 24] we call *frame* fluents those fluents to which the law of inertia applies. However, we consider frame fluents as being dependent on the actions.

3.1 Extensions for a domain description

We are now going to introduce a notion of extension first for domain frames $(\Pi, Frame)$, and then for domain descriptions $(\Pi, Frame, Obs)$.

The idea underlying the definition of the notion of extension for a domain description is that persistency of frame fluents from a state to the next one must be assumed if it does not lead to an inconsistent state. For this reason we introduce a set of *persistency laws* of the form

$$[A](l \rightarrow [a]l),$$

one for every action sequence A, action a, and fluent literal l such that $(|l|, a) \in Frame$. The meaning of the persistency law above is that, "if l holds in the state obtained by executing the action sequence A, then l has to persist after executing a in that state".

This set of persistency laws is determined by the set *Frame* of frame fluents in the domain description. They are used, in addition to the formulas in Π, to determine the next state when an action is performed. As a difference with the formulas in Π, persistency laws must be regarded as being *defeasible*. They are regarded as assumptions to be maximized. It is by maximizing persistency assumptions, that we pursue minimization of changes in the world, when executing an action. More precisely, the idea is that the persistency law $[A](l \rightarrow [a]l)$ can be assumed (and has to be assumed) unless we are able to prove that $\neg l$ holds after execution of action a, i.e., unless we are able to prove $[A; a]\neg l$. This is a sort of "local" maximization of persistency assumptions, that is, assumptions are maximized state by state. This is different from maximizing them globally, by taking maximal consistent sets of persistency assumptions[2].

Our notion of extension will require to introduce two different kinds of assumptions. The first kind of assumptions, as we have seen, are persistency assumptions.

Given a set $Frame$ a set WP_{Frame} of persistency assumptions is defined as follows:

$$WP_{Frame} = \{[A](l \rightarrow [a]l) : \ (|l|, a) \in Frame\}$$

(where A stands for any action sequence including the empty one). Persistency laws have to be assumed if this does not lead to inconsistencies.

In addition to persistency assumptions, in defining our semantics, we introduce another kind of assumptions, which are needed to deal with non frame fluents. If a fluent f is not persistent with respect to an action a (i.e., $(f, a) \notin Frame$), then in the state obtained after executing a the value of f might be either true or false. Hence, we introduce assumptions which allow to assume, in any state, the value true or false for each nonframe fluent f. Assumptions are also needed on the initial state. Given a set $Frame$, we define a following set $Ass_{NonFrame}$ of assumptions on the value of non persistent fluents and on the value of fluents in the initial state as follows:

$$Ass_{NonFrame} = \{[A; a]l : \ (|l|, a) \notin Frame\} \cup \{[\varepsilon]l : \ l \in Lit\}$$

We can represent a generic assumption in this set by $[A]l$, which includes assumptions on the initial state (when A is the empty sequence).

Now we introduce our notion of extension.

[2] Notice that introducing persistency laws of the form $\Box(l \rightarrow [a]l)$ wouldn't be enough to deal with the persistency of literals at each different state.

Definition 5 Let $D = (\Pi, Frame)$ be a domain frame. An *extension* of D is a set $E = Th_{AD}(\Pi \cup WP' \cup F)$, such that $WP' \subseteq WP_{Frame}$ and $F \subseteq Ass_{NonFrame}$ and

(a) if $[A](l \rightarrow [a]l) \in WP_{Frame}$, then
$[A](l \rightarrow [a]l) \in WP' \iff [A;a]\neg l \notin E$

(b) if $[A]l \in Ass_{NonFrame}$, then $[A]l \in F \iff [A]\neg l \notin E$,

where A is any (possibly empty) action sequence.

The $\Rightarrow$-part of condition (a) is a consistency condition, which guarentees that a persistency axiom $[A](l \rightarrow [a]l)$ cannot be assumed in WP' if $\neg l$ can be deduced as an immediate or indirect effect of the action a. We say that the formula $[A;a]\neg l$ blocks the persistency axiom. The $\Leftarrow$-part of condition (a) is a maximality condition which forces the persistency axiom to be assumed in WP', if the formula $[A;a]\neg l$ is not proved.

Condition (b) forces each state of an extension to be complete: for all finite sequences of actions $a_1;\ldots;a_n$ each non persistent fluent must be assumed to be true or false in the state obtained after executing them. In particular, since the sequence of actions may be empty, the initial state has to be complete in a given extension E. This is essential for dealing with domain descriptions in which the initial state is incompletely specified and with postdiction.

Note that the conditions above have a clear similarity with the applicability conditions for a default rule in an extension: for a default rule to be applicable its justification has to be consistent with the extension; moreover, if the justification is consistent the default rule has to be applied. In section 3.3 we will make the relationship with default extensions more precise.

The following proposition gives a fixed point characterization of extensions.

Theorem 6 Given a domain frame $D = (\Pi,\ Frame)$, E is an extension of D if and only if E is a solution of the equation

$$\begin{aligned} E = Th_{AD}(\Pi \quad & \cup \{[A](l \rightarrow [a]l) : [A](l \rightarrow [a]l) \in WP_{Frame} \\ & \qquad and\ [A;a]\neg l \notin E\} \\ & \cup \{[A]l : [A]l \in Ass_{NonFrame}\ and\ [A]\neg l \notin E\}) \end{aligned}$$

The proof is straightforward from definition 5.

We can now define the notion of extension for domain descriptions.

Definition 7 E is an *extension for a domain description* $(\Pi, Frame, Obs)$ iff E is an extension for the domain frame $(\Pi, Frame)$ and $E \vdash_{AD} Obs$.

Notice that first we have defined extensions of a domain frame $(\Pi, Frame)$; then we have used the observations in Obs to filter out those extensions which do not satisfy them. This corresponds to what in [39] Sandewall has called "filtering".

Let us consider Example 2. The domain description $(\Pi, Frame, Obs)$ has an extension E, in which $[load;wait;shoot]\neg alive$ holds, containing (among others) the formulas

$$\begin{array}{ll} (a) & [\varepsilon]alive \qquad (b)[\varepsilon]\neg loaded \\ (c) & [\varepsilon](alive \rightarrow [load]alive) \\ (d) & [load](alive \rightarrow [wait]alive) \\ (e) & [load;wait]loaded. \end{array}$$

(a) and (b) are assumptions in the initial state, while (c) and (d) are persistency assuptions of $alive$ through the sequence of actions $load; wait$.

On the other hand, there is no extension in which $[load; wait; shoot]alive$ holds, since persistency of $alive$ after $shoot$ cannot be consistently assumed.

Note that the state obtained after execution of the sequence of actions $a_1; \ldots; a_n$, is only determined by the assumptions made from the initial state up to that state. Assumptions on later states have no influence in this respect. For instance, in the example above, if we execute the actions $load$, $wait$ and $shoot$ in the order, assuming that $alive$ persists after $shoot$ cannot allow to block the persistency of $loaded$ after $wait$. In this regard, it is essential that action modalities are not serial, so that action laws cannot be used contrapositively. In particular, assuming that alive persists after the shooting, from $[load; wait; shoot]alive$ and from the action law

$$\Box(loaded \rightarrow [shoot]\neg alive)$$

we cannot, anyhow, conclude $[load; wait]\neg loaded$. In fact, the contrapositive formula

$$[load; wait](\neg[shoot]\neg alive \rightarrow \neg loaded),$$

which follows from the action law above, cannot be applied, since $[load; wait]\,[shoot]alive$ is not equivalent to $[load; wait]\neg[shoot]\neg alive$. More generally, in our logic, we cannot conclude $\neg[a]q$ from $[a]\neg q$ since $[a]$ is not serial. Note that, on the other hand, in situation calculus, $holds(\neg q, do(a, s))$ is equivalent to $\neg holds(q, do(a, s))$, which would correspond to regard $[a]\neg p$ as being equivalent to $\neg[a]p$.

3.2 States and transitions among states

An extension E determines an initial state and a *transition function among states*. In particular, the state reachable through an action sequence A in E can be defined as :

$$S^E_{[A]} = \{l \,:\, [A]l \in E\}$$

($S^E_{[\varepsilon]}$ then represents the initial state). An extension E determines a transition function Ψ_E which maps each state and action to a new state as follows:

$$\Psi_E(S^E_{[A]}, a) = S^E_{[A;a]}.$$

Note that this transition structure is clearly similar to the structures introduced to define the semantics of the language $\mathcal{A}$ [18] and of the language $\mathcal{AR}_0$ [24].

We have already pointed out that each state $S^E_{[a_1;\ldots;a_j]}$ is *complete*, in the sense that, for each fluent f, it contains either f or $\neg f$. We prove the following theorem.

Theorem 8 Given a domain frame $D = (\Pi, Frame)$ and an extension E of D, $S^E_{[A]}$ is complete for every action sequence A (including the empty one).

Proof Let $E = Th_{AD}(\Pi \cup WP' \cup F)$ with $WP' \subseteq WP_{Frame}$ and $F \subseteq Ass_{NonFrame}$. We prove that, for any action sequence A and any fluent f, either $[A]f \in E$ or $[A]\neg f \in E$, by induction on the length n of A.

If $n = 0$ then $A = \varepsilon$ and $S^E_{[A]}$ is the initial state. Both $[\varepsilon]f$ and $[\varepsilon]\neg f$ are in $Ass_{NonFrame}$. Assume that $[\varepsilon]f \notin E$. Then, by definition 5 (b), $[\varepsilon]\neg f \in F$. Hence, $[\varepsilon]\neg f \in E$. Similarly, in the case when $[\varepsilon]\neg f \notin E$, we can prove that $[\varepsilon]f \in E$.

If $n > 0$, let $A = A'; a$. We know, by induction hypothesis that the state $S^E_{[A']}$ is complete. There are two cases either $(f, a) \in Frame$ or $(f, a) \notin Frame$.

(i) If $(f, a) \in Frame$, then both $[A'](f \rightarrow [a]f) \in WP_{Frame}$ and $[A'](\neg f \rightarrow [a]\neg f) \in WP_{Frame}$. Assume that both $[A]f \notin E$ and $[A]\neg f \notin E$. From the fact that $[A]f \notin E$, by definition 5 (a), we can conclude that $[A'](\neg f \rightarrow [a]\neg f) \in WP' \subseteq E$. Similarly, from the fact that $[A]\neg f \notin E$, we can conclude that $[A'](f \rightarrow [a]f) \in WP' \subseteq E$. Since, by induction hypothesis, the state $S^E_{[A']}$ is complete, either $[A']f \in E$ or $[A']\neg f \in E$. In the first case, from $[A']f$ and $[A'](f \rightarrow [a]f)$ being in E, we can conclude that $[A'; a]f$ is in E (since E is closed with respect to logical consequence). This contradicts the assumption that $[A]f \notin E$. Similarly, if $[A']\neg f \in E$, since $[A'](\neg f \rightarrow [a]\neg f)$ is in E, we can conclude that $[A'; a]\neg f$ is in E. Hence, again, we have a contradiction (with the assumption that $[A]\neg f \notin E$).

(ii) If $(f, a) \notin Frame$, then both $[A]f$ and $[A]\neg f$ are in $Ass_{NonFrame}$. Assume that $[A]f \notin E$. Then, by definition 5 (b), $[A]\neg f \in F$. Hence, $[A]\neg f \in E$. Similarly, in the case when $[A]\neg f \notin E$, we can prove that $[A]f \in E$.

Let us point out that the completeness of states is due both to assumptions on nonframe literals and to persistency assumptions. Persistency assumptions only allow to preserve completeness of states as regards the value of frame fluents. On the other hand, completeness of states concerning non frame fluents is guaranteed by the assumptions in $Ass_{NonFrame}$.

We have already remarked for the *shooting problem* that the state obtained after execution of the sequence of actions $a_1; \ldots; a_n$ is only determined by the assumptions made from the initial state up to that state, and what holds at later states does not have any influence on it. More precisely, it is possible to show that each state can be determined from the previous one by making use of the persistency laws in that state together with action laws and causal rules. This means that an extension can be built step by step, one state after the other from the initial one.

We can prove that this property holds in general, if reasonable assumptions are made on domain frames. Given a domain frame $(\Pi, Frame)$, we will assume that in action laws $\Box(\pi \rightarrow [a]\rho)$ and in causal rules $\Box(\alpha \rightarrow ⓒ\beta)$, π, ρ, α and β are non-modal formulas. Moreover, we will assume that all fluents are frame fluents, that is, $Frame = \{(l, a) : a \in A, l \in \mathcal{F}\}$ [3]. The previous examples 2 and 4 satisfy these assumptions.

Proposition 9 Let $E = Th(\Pi \cup WP' \cup F)$ be an extension of $(\Pi, Frame)$ and let

- $S = S^E_{[a_1;\ldots;a_n]}$ be a consistent state;
- $\Pi|_{a_1;\ldots;a_n;a} = \{\pi \rightarrow [a]\rho \ : \ \Box(\pi \rightarrow [a]\rho) \in \Pi\} \cup$
  $\{[a](\alpha \rightarrow ⓒ\beta) \ : \ \Box(\alpha \rightarrow ⓒ\beta) \in \Pi\}$.

[3] This constraint could be easily removed.

- $WP'|_{a_1;\ldots;a_n;a} = \{l \rightarrow [a]l \ : [a_1;\ldots;a_n](l \rightarrow [a]l) \in WP'\}$.

Then

$$E \vdash_{AD} [a_1;\ldots;a_n;a]l \quad \Leftrightarrow \quad S \cup \Pi|_{a_1;\ldots;a_n;a} \cup WP'|_{a_1;\ldots;a_n;a} \vdash_{AD} [a]l.$$

The proof is given in the Appendix B.

Roughly speaking, $\Pi|_{a_1;\ldots;a_n;a}$ is the set of action laws and causal rules that can be applied in the state S when action a is executed. $WP'|_{a_1;\ldots;a_n;a}$ is the set of persistency laws applied in the state S when a is executed. The proposition claims that the state $S^E_{[a_1;\ldots;a_n;a]}$ obtained from S by executing a is determined only by these two sets of formulas.

Going back to the *shooting problem*, it is immediate to see that nothing can change after a *wait* action, because no action law is defined for it.

Let us consider now Example 4. We can show that $(\Pi, Frame, Obs)$ has an extension E, in which $[flip_1; flip_2]open$ holds, by going through the intermediate states $[\varepsilon]$ and $[flip_1]$.

To explain the observations, state $[\varepsilon]$ must contain the following assumptions

$$[\varepsilon]\neg up_1 \qquad [\varepsilon]\neg up_2 \qquad [\varepsilon]\neg open$$

that is $S^E_{[\varepsilon]} = \{\neg up_1, \neg up_2, \neg open\}$.

The next state $S^E_{[flip_1]}$ can be obtained from $S^E_{[\varepsilon]}$ by applying the action law

$$\neg up_1 \rightarrow [flip_1]up_1$$

and the two persistency assumptions

$$\begin{array}{l} \neg open \rightarrow [flip_1]\neg open \\ \neg up_2 \rightarrow [flip_1]\neg up_2 \end{array}$$

Therefore we have $S^E_{[flip_1]} = \{up_1, \neg up_2, \neg open\}$.

Finally, by applying the following action law, causal rule and persistency assumption to the state $S^E_{[flip_1]}$

$$\begin{array}{l} \neg up_2 \rightarrow [flip_2]up_2 \\ up_1 \wedge up_2 \rightarrow \copyright open \\ up_1 \rightarrow [flip_1]up_1 \end{array}$$

we obtain the next state $S^E_{[flip_1;flip_2]} = \{up_1, up_2, open\}$.

On the other hand, we do not obtain the unexpected extension in which $[flip_1; flip_2]\neg open$ holds. To avoid the last one, it is essential that causal rules are regarded as being directional (see [5, 30, 28, 45]). Indeed, the causal rule $\Box(up_1 \wedge up_2 \rightarrow \copyright open)$ in Π is different from the formula $\Box(up_1 \wedge up_2 \rightarrow open)$. In particular, $\Box(up_1 \wedge up_2 \rightarrow open)$ is equivalent to $\Box(up_2 \wedge \neg open \rightarrow \neg up_1)$, by contraposition. As observed in [28] and [45], though the contrapositive formula $up_2 \wedge \neg open \rightarrow \neg up_1$ must be clearly true in any state, we do not want to use it to reason about causality, i.e., we do not want to accept that lock 2 being on and the suitcase being closed may cause lock 1 to become down. In our formalism however, the causal rule $\Box(up_1 \wedge up_2 \rightarrow \copyright open)$ is equivalent to $\Box(up_1 \wedge \neg\copyright open \rightarrow \neg up_2)$ but *not* to $\Box(up_1 \wedge \neg open \rightarrow ⓒ\neg up_2)$. Contrapositive uses of causal rules are ruled out by the fact that we introduce a modality ⓒ to describe a causal dependency: from the causal rule $[A](\alpha \rightarrow ⓒ\beta)$ and $[A]\neg\beta$ we cannot conclude $[A]\neg\alpha$ [4].

When a literal l is not frame with respect to an action a, it can be assumed to be either true or false after executing a. This is useful for modelling nondeterministic changes, since it gives the possibility that l changes without being forced by the occurrence of an action, thus allowing for nonminimal change.

Let us consider again the Russian turkey problem, in which the fluent *loaded* is nonframe with respect to the action *spin*. More precisely, let us consider the domain description in example 3. If we perform the action *load* and then *spin* the gun may result to be loaded or unloaded: D has two extensions, both of them contain $[load]loaded$; one contains $[load; spin]loaded$ while the other contains $[load; spin]\neg loaded$. The gun behaves precisely as expected. Since we do not know whether the gun will be loaded after spinning or not, both cases are possible.

The fact that a nonframe fluent f with respect to action a may take an arbitrary value after execution of a is tightly related to our semantic choice, in which extensions are two-valued. In fact, as we have seen, in our extensions each state is complete: for each fluent f, either f holds or $\neg f$ holds and no fluent is undefined in a state. With a different semantic choice, in which extensions are three-valued, nonframe fluents would naturally give rise to incomplete states, in which they are undefined. In such a case, defining a nondeterministic action a would also require to state explicitly that after action a either f has to hold or $\neg f$ has to hold.

Notice that a state $S^E_{[A]}$ can be inconsistent (containing both a literal l and its negation $\neg l$). This happens, for instance, when Π contains alternative action laws for an action a, which are applicable in the same state and have mutually inconsistent effects. Consider the state when there are two action laws $p \rightarrow [a]f$ and $q \rightarrow [a]\neg f$ and p and q are two different fluents. In a state in which both p and q hold, both of the action laws above are applicable, which leads to an inconsistent state containing both f and $\neg f$. Hence, an extension E may be $[A]$-inconsistent for some action sequence A (i.e. $[A]\bot \in E$), even if E is a consistent extension (i.e. $\bot \notin E$).

Given domain description $(\Pi, Frame, Obs)$, it may be the case that it has no extensions. Consider the following example also mentioned by [30]:

Π: $\quad \Box[a]p \quad \Box(q \rightarrow ⓒ\neg p)$
Obs: $[\varepsilon](q \wedge \neg p)$
Frame= $\{(l, a) : a \in \mathcal{A}, l \in \mathcal{F}\}$.

This domain description has no extension. In fact, when q and $\neg p$ hold in the initial state, performing action a makes p true, but it cannot block the persistency of q since $\neg q$ cannot be derived from p using the causal rule. However, assuming that q persists after the action leads to a contradiction, since q causes $\neg p$. From a contradiction we can derive any formula, including $\neg q$, which blocks the persistency of q.

A similar example can be given to show that also a domain frame may have no extension.

[4] Note that, to make this true, it is important that both the modalities $[A]$ and the modality ⓒ are not serial.

3.3 Relation to Defaults

The persistence rules in WP' can be seen as *defaults*. They belong to an *extension* as far as this is justified, which is the case whenever the negation of the *justification* does not belong to the extension. This suggests to study the relationship of our formalism with default logic. But default logic is defined on a classical logic basis. However, when looking at the definition of defaults, extensions, etc. in the Reiter framework [37], one can see that they do not especially depend on the classical language and can be adapted straightforwardly to a modal language.

Given an action language AD, an $AD-$default is a rule $\frac{\alpha : \beta}{\gamma}$, where α, β and γ are formulas from AD.

An $AD-$default theory Δ is a pair (W, D), such that W is a set of formulas over AD and D is a set of $AD-$defaults.

Given an $AD-$default theory $\Delta = (W, D)$, for any set S of formulas of AD, $\Gamma_\Delta(S)$ is the smallest set satisfying the following three properties:

(1) $W \subseteq \Gamma_\Delta(S)$

(2) $Th_{AD}(\Gamma(S)) = \Gamma(S)$

(3) if $\frac{\alpha : \beta}{\gamma} \in D$ and $\alpha \in \Gamma_\Delta(S)$ and $\neg\beta \notin S$, then $\gamma \in \Gamma_\Delta(S)$

A set E of formulas from AD is a *default extension* for Δ iff $\Gamma_\Delta(E) = E$.

Let us mention the following theorem adapted from Reiter [37]:

If E is a default extension of an $AD-$default theory $\Delta = (W, D)$ then

$$E = Th_{AD}(W \cup \{\gamma : \frac{\alpha : \beta}{\gamma} \in D and \alpha \in E and \neg\beta \notin E\})$$

Given a domain frame, it is possible to define an $AD-$default theory whose default extensions are extensions of the domain frame.

Let $(\Pi, Frame)$ be a domain frame. We define the associated $AD-$default theory $\Delta = (W, D)$ in the following way.

The set of basic laws is $W = \Pi$, the set of defaults is

$$\begin{aligned} D = \quad & \{\frac{: \neg[A;a]\neg l}{[A](l \to [a]l)} : [A](l \to [a]l) \in WP_{Frame}\} \\ & \cup\{\frac{: \neg[A]\neg l}{[A]l} : [A]l \in Ass_{NonFrame}\} \end{aligned}$$

for every action a and action sequence A. Then the following theorem shows the correspondence between default extensions and action theory extensions.

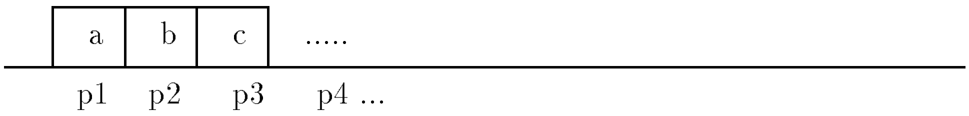
Theorem 10 Let $(\Pi, Frame)$ be a domain frame. Let $\Delta = (\Pi, D)$ be the associated $AD-$default theory. Then E is an extension for $(\Pi, Frame)$ if and only if E is a default extension of Δ.

The proof can be found in Appendix A.

3.4 Further issues concerning ramification

Many action theories today can handle correctly the issue of the ramification problem which involves choosing between multiple extensions. Our solution makes use of a causality operator which prevents to use the contrapositives of causal rules. So, we can correctly treat examples like the one introduced in [26], in which there are two switches and a light (which is on when both the switches are on, and off viceversa) by replacing the general law $(switch_1 \leftrightarrow switch_2) \leftrightarrow light\_on$ by the two causal rules $(switch_1 \leftrightarrow switch_2) \rightarrow \copyright light\_on$ and $(switch_1 \leftrightarrow \neg switch_2) \rightarrow \copyright\neg light\_on$.

A more difficult issue of the ramification problem involves indirect changes which are caused by the effects of an action, if some preconditions hold on the state preceeding action execution. As an example, consider the following scenario, where a number of blocks are in a sequence (example 11). When the first block, a, is pushed from the place p_1 to the place p_2, all other blocks move also to the next place. We can formalise this example as follows [5].



Example 11 Let $p_1, \ldots p_n$ be places and a, b, c be blocks, and let $push(x,p)$ be the action which consists in pushing the block x from the place p to the next place $next(p)$.

$$
\begin{array}{ll}
\Pi : & \Box(at(x,p) \rightarrow [push(x,p)]at(x, next(p)) \\
 & \Box(at(x,p) \wedge p \neq q \rightarrow \neg at(x,q)) \\
 & \Box(at(y,r) \wedge ([push(x,p)]at(z,r)) \wedge z \neq y \rightarrow [push(x,p)]at(y, next(r))) \\
 & \Box(p_{i+1} = next(p_i)) \\
Obs : & [\varepsilon](at(a,p_1) \wedge at(b,p_2) \wedge at(c,p_3)) \\
Frame = & \{(l,a) : \ a \in \mathcal{A}, \ l \in \mathcal{F}\}
\end{array}
$$

The formula

$$\Box(at(y,r) \wedge ([push(x,p)]at(z,r)) \wedge z \neq y \rightarrow [push(x,p)]at(y, next(r))) \qquad (1)$$

says that if action $push(x,p)$ has the effect of moving z in r, then this action has the indirect effect of moving y, which was previously in r, to the place next to r.

Rule (1) is syntactically an action law, whose precondition is a non propositional formula. Indeed it has the form $\gamma \wedge [a]\alpha \rightarrow [a]\beta$. To see the relationship between such a formula and causal rules, let us point out that a causal rule $\Box(\alpha \rightarrow \copyright\beta)$ could also be formalized in the modal language by introducing the axiom schema $[A]\alpha \rightarrow [A]\beta$, where A is any action sequence. Note that $[A]\alpha \rightarrow [A]\beta$ is entailed by $\Box(\alpha \rightarrow \copyright\beta)$. Hence, rule (1) can be regarded as a ramification rule which is specifically related to the action $push(x,p)$ and (furthermore) which can be applied if some precondition holds on the state before the action is executed.

Assume that the action $[push(a,p_1)]$ occurs. This yields:

[5] We use first-order predicates as action names in this representation. But here, this is syntactic sugar: the variables range over finite sets and the propositional formulation can straightforwardly be obtained by replacing all varables by constants.

$[push(a, p_1)]at(a, p_2)$ from the action law,
$[push(a, p_1)]at(b, p_3)$ from the ramification law
since $at(b, p_2)$ and $[push(a, p_1)]at(a, p_2)$ for $a \neq b$
$[push(a, p_1)]at(c, p_4)$ similarly.

Dependency on the action is important, as it allows for instance to describe the ramification effects of two different actions, like push-right and push-left, as follows:

$$\Box(at(y, r) \wedge ([push\_r(x, p)]at(z, r)) \wedge z \neq y \rightarrow [push\_r(x, p)]at(y, next(r))) \qquad (2)$$

and

$$\Box(at(y, r) \wedge ([push\_l(x, p)]at(z, r)) \wedge z \neq y \rightarrow [push\_l(x, p)]at(y, prev(r))) \qquad (3)$$

While the first one moves the block y (previously in r) to the place next to r, the second one moves it to the place previous to r.

This example cannot be treated in the language $\mathcal{AR}$ of Giunchiglia, Kartha and Lifschitz [21] nor in the language $\mathcal{AC}$ of McCain and Turner [31]. In fact, the following action law

$$push(x, p) \textbf{ causes } ((at(z, r) \wedge z \neq y) \rightarrow at(y, next(r))) \textbf{ if } at(y, r)$$

which can be defined in both languages, does not correspond to rule (1) above, which can be regarded as a causal rule related to the action $push(x, p)$. In fact, matherial implication in the effect is not well suited to capture the ramification, and, even in the language $\mathcal{AC}$ in which causal rules can be expressed, the effect $(at(z, r) \wedge z \neq y) \rightarrow at(y, next(r))$ of action $push(x, p)$ cannot be expressed as a causal rule.

3.5 Non minimal changes and postdiction

In this section we will describe, through some examples, how non-minimal change, non deterministic actions and postdiction can be formalized in our language.

Non-minimal change is another issue related to causality. It concerns fluents that may not persist as an action is performed, even when they are consistent with the action consequences. The most classical problem in this context concerns the *non persistence* of fluents, which have been *caused* by the execution of an action and may disappear together with the action result which caused them. Consider the following example due to Myers and Smith [34]. The world is characterized by two fluents specifying whether some object (the heavy hammer) is safe (from the baby), and whether it is on the table. There is one constraint, namely, whenever an object is on the table, then it is safe. There are two actions, "putting the hammer on the table" and "removing the hammer from the table". Initially, the hammer is not on the table. Then the events of "putting the hammer on the table", and "removing the hammer from the table" take place in succession. At the end, the hammer is no more on the table, and we do not know if it is safe or not.

Example 12 Let $safe$ and $onTable$ be fluents, and let $Puton$ and $Removefrom$ be action symbols.

$$
\begin{array}{ll}
\Pi: & \Box(\neg OnTable \rightarrow [Puton]OnTable) \\
 & \Box(OnTable \rightarrow [Removefrom]\neg OnTable) \\
 & \Box(OnTable \rightarrow ⓒ safe) \\
\mathit{Obs:} & [\varepsilon]\neg OnTable
\end{array}
$$
$$Frame = \{(l,a):\ a \in \mathcal{A},\ l \in \mathcal{F}\} \setminus \{(safe, Removefrom)\}.$$

Assume that the actions Puton and Removefrom take place subsequently. All possible extensions for this domain description will contain the following formulas:

$[\varepsilon]\neg OnTable$
$[Puton](OnTable \wedge safe)$
$[Puton; Removefrom]\neg OnTable$

Since we do not suppose that $safe$ persists after $Removefrom$ took place, there are at least two extensions, the one containing $[Puton; Removefrom]safe$, and the other one containing $[Puton;\ Removefrom]\neg safe$.

In the above domain description, it would be reasonable to have an action precondition law saying that the action *Removefrom* can only be executed in a state in which *OnTable* holds. Such an action precondition law would allow to get rid of those states in which the action *Removefrom* is executed several times, one after the other. Such states are unwanted since, any time the *Removefrom* action is executed, the fluent *safe* may take an arbitrary value, so that, for instance, we could have an extension containing $[Removefrom]safe$ and $[Removefrom; Removefrom]\neg safe$. We will come back to the issue concerning action qualifications in Section 3.6.

Note that, using nonframe fluents, we can easily represent nondeterministic actions. An action a with inclusive indeterminate effects p_1 or p_2 is represented by the action law $\Box[a](p_1 \vee p_2)$ with $(p_i, a) \notin Frame$. An action $flip$ with exclusive indeterminate effects $head$ or $tail$ can be represented by the action law $\Box[flip](head \vee tail)$ and the constraint $\Box ⓒ \neg(head \wedge tail)$, with $(head, flip) \notin Frame$ and $(tail, flip) \notin Frame$.

The following example due to [6] shows the need of assuming a "complete" initial state for an extension. In many cases, the initial state of a theory is not completely known, i.e. we do not know the initial value of all fluents. To deal with this case and with postdiction, in building an extension, we introduce assumptions about the value of fluents in the initial state in order to make it complete (in all possible ways).

Example 13 There is a turkey who is sleeping and an alarm clock. When the clock is ringing, the turkey will wake up, unless he is deaf. We propose the following domain description and observations:

$$
\begin{array}{ll}
\Pi: & \Box([ring]alarm) \\
 & \Box(\neg deaf \wedge alarm \rightarrow ⓒ\neg sleepy) \\
Obs1: & [\varepsilon]sleepy \\
Frame = & \{(l,a):\ a \in \mathcal{A},\ l \in \mathcal{F}\}.
\end{array}
$$

In the initial state nothing is known about the turkey being deaf or not. There are two extensions one E_1 containing $[\varepsilon]deaf$, and $[\varepsilon](sleepy \rightarrow [ring]sleepy)$, such that $sleepy$ persists; the other one E_2 containing $[\varepsilon]\neg deaf$ and $[ring]\neg sleepy$, which is *caused* by $[ring]alarm$.

Now consider the same example, but with $Obs2 : [\varepsilon]sleepy \wedge [ring]sleepy$. Here we know that the turkey is sleepy after *ring*. An extension for this domain description is

E_1 and it must contain the assumptions $[\varepsilon]deaf$ and $[\varepsilon](deaf \rightarrow [ring]deaf)$, which are needed to explain the fact that the turkey is sleepy after ringing. Note that, assuming $[\varepsilon]deaf$ makes the initial state complete.

3.6 Qualification

Explicit qualifications for actions can be represented in our language by making use of action laws with false consequent. In particular, we can say that action a cannot be executed in a state in which π holds by introducing the following action law:

$$\pi \rightarrow [a]\bot,$$

where $\bot$ is denotes the inconsistency. Consider the following example from [24].

Example 14 A door in a hotel room can be opened by inserting the entry card. In the initial situation the door is closed. There are two fluents, $Door\_Open$ and Has_Card, and an action $Insert\_Card$, which makes the door open, and can only be executed in a state in which Has_Card holds. Consider the following domain description:

$$\begin{array}{ll}
\Pi: & \Box([Insert_Card]Door_Open) \\
 & \Box(\neg Has_Card \rightarrow [Insert_Card]\bot) \\
Obs: & [\varepsilon]\neg Door_Open \\
Frame = & \{(l,a): \ a \in \mathcal{A}, \ l \in \mathcal{F}\}.
\end{array}$$

The second action law in Π says that the action $Insert\_Card$ cannot be executed if Has_Card is false. Has_Card is indeterminate in the initial state. This domain description has an extension in which Has_Card holds in the initial state, and in the state obtained after executing action $Insert\_Card$ the door in open. Moreover, there is another extension in which, instead, $\neg Has\_Card$ holds in the initial state. In such an extension $[Insert\_Card]\bot$ holds, which means that the action $Insert\_Card$ is not possible in the initial state.

In a similar way, in Example 12 we could add to the domain description the action law $\Box(\neg OnTable \rightarrow [Removefrom]\bot)$ to express the fact that the action $Removefrom$ is only executable in a state in which $OnTable$ holds.

In essence, we represent the fact that an action is not executable in a certain state by the fact that it leads to a state containing the inconsistency $\bot$. Notice that $\bot$ is inconsistency in classical logic, and from it every formula can be derived. Hence, when an inconsistent state is reached, all the subsequent states will be inconsistent too. Therefore, all action sequences containing an action which cannot be executed are not qualified (and lead to an inconsistent state).

While our language may deal with this simple kind of explicit qualifications of actions, the presence of implicit qualifications may lead to non-existence of extensions.

Consider for instance the following example from [27].

Example 15 In an ancient kingdom there are two blocks, either block may be painted yellow, but (by order of the emperor) at most one block can be painted yellow. Hence, in a state in which the second block is yellow the first one cannot be painted yellow. The following domain description.

$$\begin{array}{ll} \Pi : & \Box([Paint_1]Yellow_1) \\ & \Box([Paint_2]Yellow_2) \\ & \Box ⓒ(Yellow_1 \wedge Yellow_2 \rightarrow \bot) \\ Obs : & [\varepsilon](Yellow_1 \wedge \neg Yellow_2) \\ Frame = & \{(l, a) :\ a \in \mathcal{A},\ l \in \mathcal{F}\}. \end{array}$$

has no extension. Consider the case when we execute action $Paint_2$ in the initial state. After executing it $Yellow_2$ has to be true. There are two cases. If $Yellow_1$ persists, then $[Paint_2]\bot$ can be derived, and from it $[Paint_2]\neg Yellow_1$, which violates condition (a) in the definition of extension. On the other hand, if $Yellow_1$ does not persists, again condition (a) is violated, since $[Paint_2]\neg Yellow_1$ cannot be proved.

What is unsatisfactory in the formalization of the problem above is the fact that inconsistency is not confined to a single state, the one which should be unreachable. On the contrary it has the global effect that no solution exists for the domain description. Some different solution could be devised to deal with this problem, like, for instance, introducing an alternative notion of inconsistency, which is different from that of classical logic, so that from inconsistency one cannot derive every formula (this happens for instance with inconsistency in "minimal logic"). Another possibility is that of adopting a different definition of extension, in which only action sequences of interest and the states reachable by them are considered. A similar notion has been introduced in section 4 in theorem 19.

4 Related Work

The observation that classical implication cannot correctly represent causality relations led to many action theories who studied the integration of causality rules [28, 29, 20, 30, 45].

Framefluents

The first systematic solution to the ramification problem has been proposed by Lifschitz in [26]. He distiguishes between *frame* and *non-frame* fluents and defines the frame law to be

$$framefluent(f) \rightarrow (\neg ab(f, a, s) \rightarrow (Holds(f, s) \leftrightarrow Holds(f, result(a, s)))).$$

Framefluents have to persist, if possible, as an action occurs, while non-frame fluents may or may not change. This solution is not quite satisfactory since framefluents do not depend on actions. Moreover, it can be seen that the distinction among frame and nonframe fluents does not allow the suitcase problem (of example 4) to be solved.

Consider example 4, for which a formalization in the situation calculus can be given, containing the formulas $\neg Holds(up_i, s) \leftrightarrow Holds(up_i, result(flip_i, s)$ (for $i = 1, \ldots, 2$) and $Holds(up_1, s) \wedge Holds(up_2, s) \rightarrow Holds(open, s)$. Consider the initial state $\neg Holds(open, s_0) \wedge Holds(up_1, s_0) \wedge \neg Holds(up_2, s_0)$. Let us assume that all fluents are frame. Hence, using the frame law above is like using the simple frame law $\neg ab(f,a,s) \rightarrow (Holds(f,s) \leftrightarrow Holds(f,result(a,s)))$. After flipping lock 2 and minimizing the predicate ab, one gets two minimal models, one containing, for $s_1 = result(flip_2, s_0)$, $\neg Holds(open, s_1) \wedge \neg Holds(up_1, s_1)$ letting $\neg open$ to persist and the other containing $Holds(open, s_1) \wedge Holds(up_1, s_1)$ letting up_1 to persist.

In order to supress the first solution (which is unintended) one has to prevent $open$ from persisting. This is achieved by modifying the frame law and by declaring up_1 and up_2 to be $framefluents$ and $open$ to be non frame. However, with this choice, $open$ can never persist (since it is nonframe). Consider the following initial situation, where the suitcase is not open and both locks are not up, i.e. $\neg open \wedge \neg up_1 \wedge \neg up_2$. After flipping one of the locks, e.g. lock 1, we can derive that lock 2 is still not up but we are unable to predict what happened to $open$, since $open$ cannot persist, because it is not a framefluent and nothing can be derived for $open$ since lock 2 is still not up. This is cleary not what we expect. Notice that the above problem arises even in the case when framefluents depend on actions.

Relation to Lin's work

Lin in [28, 29] introduces a predicate *Caused(f,v,s)*, meaning that fluent f is caused to have the truth value v in the situation s. The basic axioms for $Caused$ are $Caused(f,true,s) \rightarrow Holds(f,s)$ and $Caused(f,false,s) \rightarrow \neg Holds(f,s)$. Action laws are expressed by formulas of the form

$$Poss(a,s) \rightarrow (Holds(\phi,s) \rightarrow Caused(F,v,do(a,s))),$$

where F is a fluent name. The predicate $Caused$ is equally used for formulating causal relationships within domain constraints. For example, the domain constraint of the suitcase example 4 is then $Holds(up(1),s) \wedge Holds(up(2),s) \rightarrow Caused(open,true,s)$. $Caused$ is circumscribed in the theory consisting of constraints and action laws with $Holds$ varying. This yields a first-order theory where a fluent f is $Caused$ (to admit the truth value v), whenever f is caused to be v as an immediate or ramification effect of an action. Frame laws expressing that a fluent persists as long as it is not $caused$ to change its value are added to the theory *after* the minimization of $caused$. This is important: given a theory T *without* frame axioms, $Caused$ is circumscribed in T and the frame axioms F are added after that, yielding an action theory $Circ(T, Caused, Holds) \cup F$.

Though our ⓒ modality is very similar to Lin's *Caused* predicate, they are used in a rather different way. The *Caused* predicate is used in order to control the persistency of fluents: only fluents which are caused are allowed to change value (and for this reason the *Caused* predicate has to occur also in action laws) and only fluents which are *not* caused are allowed to persist. On the other hand, in our definition of extension (Definition 5) condition (a) says that the persistency of a fluent l after an action a is only blocked by the fact that $\neg l$ is proved to hold after the action, and it does not require $\neg l$ to be caused [6].

[6] This choice was motivated by its simplicity, and also by the fact that we want to allow arbitrary formulas to occur in the right hand side of action laws and causal rules. In the case when persistency is governed by causality, when a complex formula is caused it is not obvious at all which subformulas and fluents occurring in it are caused, but this is essential to determine what persists to the next

Lin's approach treats correctly many problems for action theories. But it seems that his causal rules can sometimes be contraposed: from $Holds(p,s) \rightarrow Caused(q, true, s)$ and $Caused(q, false, do(a,s))$, he derives $\neg Holds(p, do(a,s))$ by the basic laws. We illustrate this through an example. Consider the following theory T_0 representing the Yale shooting example:

$$
\begin{array}{lll}
T_0: & (1) & \forall s(Holds(loaded, s) \rightarrow Caused(alive, false, do(shoot, s))) \\
 & (2) & \forall s Caused(loaded, true, do(load, s)) \\
 & (3) & Caused(f, true, s) \rightarrow Holds(f, s) \\
 & (4) & Caused(f, false, s) \rightarrow \neg Holds(f, s) \\
F_0: & & \forall s \forall a \forall f(\neg \exists v Caused(f, v, s) \rightarrow (Holds(f, s) \leftrightarrow Holds(f, do(a, s))))
\end{array}
$$

(1) and (2) are action laws, (3) and (4) are the basic laws about *Caused*. F_0 is the frame axiom. *Caused* is circumscribed in T_0 and F_0 is added afterwards, yielding the theory $T'_0 = Circ(T_0, Caused, Holds) \cup F_0$

Now consider the theory T_1 obtained from T_0 by adding the observation

$$O : Holds(alive, do(shoot, do(wait, do(load, s_0)))),$$

i.e. $T_1 = T_0 \cup \{O\}$ and $T'_1 = Circ(T_1, Caused, Holds) \cup F_0$. The following formula is derivable from T_1 (and from T'_1):

$$\neg Holds(loaded, do(wait, do(load, s_0))) \quad (5) \tag{4}$$

This means that, as a consequence of the observation that *alive* holds after shooting, we deduce that the gun was not loaded before, i.e. after waiting. However, it is not possible in Lin's theory, to derive from this that the gun has been *caused* to become not loaded. Hence, this is not enough to block the persistency of *loaded* after *wait*, since only caused effects may block persistency (and $Caused(loaded, false, do(wait, do(load, s_0)))$ cannot be derived form (5)). Hence, we can derive in T'_1 (from the circumscription of *causes*)

$$\neg \exists v Caused(loaded, v, do(wait, do(load, s_0)))$$

and therefore, by the frame axiom F_0,

$$Holds(loaded, do(wait, do(load, s_0))) \leftrightarrow Holds(loaded, do(load, s_0)),$$

from which we get

$$Holds(loaded, do(wait, do(load, s_0))),$$

by (2) and (3). This is the negation of (5). Therefore, the circumscription of T_1 plus F_0, T'_1, is an inconsistent set. This seems to be reasonable: there is no minimal model in which in the initial state *alive* and $\neg$*loaded* hold and *alive* holds after *load*; *wait*; *shoot*.

It is not that Lin gets a solution in which the gun becomes unloaded while waiting, he gets no solution. Similarly, in our theory there is no extension for the domain description in which the observation [*load*; *wait*; *shoot*]*alive* holds. As a difference,

state. Indeed, in Lin's approach action laws and causal rules may only cause single fluents to become true or false.

however, as we have observed in section 3.1 when discussing the shooting example, in our approach the contrapositives of causal rules cannot be used to perform inferences as above, since the action and the causality modalities are not serial. From $\Box(p \rightarrow ⓒq)$ and $[a]\neg q$, we cannot derive $[a]\neg p$, since $[a]\neg p$ and $\neg[a]p$ do not entail each other. Such a difference plays an important role in the next example, in which our theory and Lin's theory [28, 29] give different outcomes. The example models the behaviour of a flip-flop.

Example 16 A flip-flop Set/Reset has two inputs Set and Reset, and two outputs E and F (which must be complementary, and represent the state of the flip-flop). The outputs are not uniquely determined by the input, but they also depend on the previous state of the flip-flop.

"Setting" the flip-flop (input $R = 0$, $S = 1$) makes the output $E = 1$, $F = 0$. "Resetting" the flip-flop (input $R = 1$, $S = 0$) makes the output $E = 0$, $F = 1$. If the input is $R = 0$, $S = 0$, E and F keep their previous values. The case when input is $R = 1$, $S = 1$, is usually excluded (it gives an output which does not represent a consistent state for a flip-flop, since E and F are not complementary; moreover, the value of E and F changes non-deterministically, when S and R are returned to 0). For this reasons, we will avoid to consider this ambiguous case.

The internal logic of the flip-flop makes use of NOR gates and it can be described through the following state constraints:

$$\begin{array}{ll} (a) & (\neg R \land \neg F) \leftrightarrow E \\ (b) & (\neg S \land \neg E) \leftrightarrow F \end{array}$$

In our formalism we can give the following domain description (containing action laws and causal rules):

$$\begin{array}{lll} \Pi: & (1) & [set](S \land \neg R) \\ & (2) & [reset](\neg S \land R) \\ & (3) & [put\_to\_zero](\neg S \land \neg R) \\ & (4) & \neg R \land \neg F \rightarrow ⓒE \\ & (5) & R \rightarrow ⓒ\neg E \\ & (6) & F \rightarrow ⓒ\neg E \\ & (7) & \neg S \land \neg E \rightarrow ⓒF \\ & (8) & S \rightarrow ⓒ\neg F \\ & (9) & E \rightarrow ⓒ\neg F \\ Obs: & & E \land \neg F \land \neg S \land \neg R \end{array}$$

In this domain description, constraints (a) and (b) above have been naturally represented by introducing the causal rules $(4), \ldots, (9)$.

After action *reset* is performed, we get R and $\neg S$ (as immediate effects) and $\neg E$ and F (by applying state constraints (5) and (7)).

If action *put_to_zero* is performed in the initial state, a next state is obtained which is equal to the initial one: $\neg R$ and $\neg S$ are obtained as immediate effects, E and $\neg F$ persist from the previous state (since neither $\neg E$ nor F can be derived to block persistency).

In Lin's formalism, the above problem can be represented by the following theory T:

Direct effects:

$$\begin{array}{ll}
(1') & \forall s Caused(S, true, do(set, s)) \\
 & \forall s Caused(R, false, do(set, s)) \\
(2') & \forall s Caused(S, false, do(reset, s)) \\
 & \forall s Caused(R, true, do(reset, s)) \\
(3') & \forall s Caused(S, false, do(put_to_zero, s)) \\
 & \forall s Caused(R, false, do(put_to_zero, s))
\end{array}$$

Causal rules:

$$\begin{array}{ll}
(4') & \forall s[\neg Holds(F, s) \wedge \neg Holds(R, s) \rightarrow Caused(E, true, s)] \\
(5') & \forall s[Holds(R, s) \rightarrow Caused(E, false, s)] \\
(6') & \forall s[Holds(F, s) \rightarrow Caused(E, false, s)] \\
(7') & \forall s[\neg Holds(E, s) \wedge \neg Holds(S, s) \rightarrow Caused(F, true, s)] \\
(8') & \forall s[Holds(S, s) \rightarrow Caused(F, false, s)] \\
(9') & \forall s[Holds(E, s) \rightarrow Caused(F, false, s)]
\end{array}$$

Unfortunally, this theory contains a dependency of $Caused(E, false, s)$ on $\neg Holds(E, s)$, due to the causal rules (6') and (7'). As observed by Lin (see [28], pp.1989) the persence of cyclic dependencies like

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

has the consequence that a *successor state axiom* cannot be computed for fluent p.

In this specific example, the cyclic dependency is indirect. However it causes problems. In particular, let us consider the outcome of executing the action *put_to_zero* in the initial state s_0. While after action *put_to_zero* the value of E and F should persist, there is a model of the circumscribed theory $Circ(T)$ plus the frame axiom

$$\neg \exists v Caused(f, v, do(a, s)) \rightarrow (Holds(f, s) \leftrightarrow Holds(f, do(a, s)))$$

in which they unexpectedly change value.

More precisely, let $s_1 = do(put\_to\_zero, s_0)$. There is a model M, minimal in the *Caused* predicate, containing (among others) the following formulas:

$Holds(E, s_0)$	$\neg Holds(F, s_0)$	$\neg Holds(R, s_0)$	$\neg Holds(S, s_0)$
$\neg Holds(E, s_1)$	$Holds(F, s_1)$	$\neg Holds(R, s_1)$	$\neg Holds(S, s_1)$
$Caused(E, false, s_1)$	$Caused(F, true, s_1)$	$Caused(R, false, s_1)$	$Caused(S, false, s_1)$

Note that, the change of the value of fluents E and F from s_0 to s_1 is justified by the fact that E is caused *false* in s_1 and F is caused *true* in s_1.

Another aspect which makes Lin's approach different from ours is that nonframe fluents are not admitted in his framework. However, in [29] he shows how actions with indeterminate effects can be specified in his causal minimization framework. This allows fluents to be released from persistency when executing an action, so that non-minimal change (see example 12) can be treated correctly.

Relation to McCain and Turner's work

In [30] McCain and Turner propose a causal approach to ramification, in which causal rules are represented by inference rules. More precisely, they focus on a single action application and they tackle the problem of determining the ramifications of an action by defining a next states function, $Res^4_C(Eff, S)$. The function $Res^4_C(Eff, S)$ gives the set of all the possible next states obtained by performing an action with explicit effect *Eff* in the state S, when C is the set of constraints (causal rules). This set is given a characterization through a fixpoint equation.

Let S be an interpretation (i.e., a maximal consistent set of literals), C a set of causal rules (of the form $\alpha \Rightarrow \beta$), $Res^4_C(Eff, S)$ is the set of interpretations S' such that

$$S' = \{L \ : \ (S \cap S') \cup Eff \vdash_C L\},$$

where $\vdash_C$ is standard derivability relation in propositional logic extended to deal with causal rules.

We can prove that our approach finds the solutions which are computed by the Res^4_C function. However, the converse does not hold. There are cases when our formalization gives solutions which cannot be found in McCain and Turner's formalization.

Example 17 Let us consider the following domain description:

$$\begin{array}{ll} \Pi : & \Box[a](p \vee q) \\ & \Box(p \rightarrow \copyright(p \wedge q)) \\ & \Box(q \rightarrow \copyright(p \wedge q)) \\ Obs : & [\varepsilon](\neg p \wedge \neg q) \\ Frame = & \{(l, a) : \ a \in \mathcal{A}, \ l \in \mathcal{F}\}. \end{array}$$

It has a single extension in which $[a](p \wedge q)$ holds. In fact, $[a](p \wedge q)$ can be derived monotonically from $[a](p \vee q)$ using the causal rules: reasoning by cases $p \wedge q$ can be concluded both when p holds (using the first causal rule) and when q holds (using the second causal rule).

In McCain and Turner's formalization, the above domain description can be modelled by taking

$$Eff = p \vee q$$

and

$$C = \{p \Rightarrow (p \wedge q), \ q \Rightarrow (p \wedge q)\}.$$

Let $S = \{\neg p, \neg q\}$. Then $Res^4_C(Eff, S) = \emptyset$, i.e., no solution can be obtained by executing an action with effect *Eff*. McCain and Turner's formalization does not capture reasoning by cases: no conclusion can be derived from the disjunctive information $p \vee q$.

We can show that given a consistent state S, all the resulting states according to the Res^4_C function, can be obtained in our formalization.

In order to prove this, it will be useful to introduce a notion of partial or "restricted" extension, in which only some sequence of actions is taken into account. Indeed, while McCain and Turner's fixed point definition only takes two states in consideration, in our notion of extension, all possible sequences of actions have to be considered along with the resulting states. More generally, given a domain description, we may be interested to know which state results from the execution of a certain action sequence while we are not interested to consider any action sequence.

A *restricted extension* E *of a domain frame on the sequence* $a_1;\ldots;a_n$ is defined precisely as in definition 5, but for the fact that A ranges only on sequences of actions which are (proper or improper) prefixes of $a_1;\ldots;a_n$. Let $WP_{a_1;\ldots;a_n}$ be the subset of WP_{Frame} containing only the assumptions $[A](l \to [a]l)$ such that the action sequence $A;a$ is a (proper or improper) prefix of the sequence $a_1;\ldots;a_n$. Similarly, let $Ass_{a_1;\ldots;a_n}$ be the subset of $Ass_{NonFrame}$ containing only the assumptions $[A;a]l$ such that the action sequence $A;a$ is a (proper or improper) prefix of the sequence $a_1;\ldots;a_n$.

Definition 18 Let $D = (\Pi, Frame)$ be a domain frame. A *restricted extension* of D *on the sequence* $a_1;\ldots;a_n$ is a set $E = Th_{AD}(\Pi \cup WP' \cup F)$, such that $WP' \subseteq WP_{a_1;\ldots;a_n}$ and $F \subseteq Ass_{a_1;\ldots;a_n}$ and

(a) if $[A](l \to [a]l) \in WP_{a_1;\ldots;a_n}$, then
$[A](l \to [a]l) \in WP' \iff [A;a]\neg l \notin E$

(b) if $[A]l \in Ass_{a_1;\ldots;a_n}$, then $[A]l \in F \iff [A]\neg l \notin E$.

Since in [30] only actions with deterministic immediate effects are considered and all fluents are frame fluents, in the following we will take $Frame = \{(l,a) : a \in \mathcal{A}, l \in \mathcal{F}\}$.

Theorem 19 Let S be a consistent state, let C be a set of causal rules, *Eff* a set of immediate effects and $S' \in Res^4_C(\mathit{Eff}, S)$. The domain frame $(\Pi, Frame)$, where

$$\Pi = \{[a]\mathit{Eff}\} \cup \{\Box(\alpha \to ©\beta) \ : \ \alpha \Rightarrow \beta \in C\}$$

and

$$Frame = \{(l,a) : a \in A, l \in \mathcal{F}\},$$

has a restricted extension E on the sequence $\varepsilon; a$, such that $S^E_{[\varepsilon]} = S$ and $S^E_{[\varepsilon;a]} = S'$.

The proof of the theorem can be found in the Appendix B.

In [30] Mc Cain and Turner also propose a conditional logic for two reasons: to make "causal rules" as object language sentences; and to enforce the property that a causal rule implies the corresponding material conditional (which does not hold for an inference rule). Our modal characterization is an alternative fixpoint characterization (see Theorem 6). Moreover, though a causal rule $\Box(\alpha \to ©\beta)$ does not logically imply the material implication $\alpha \to \beta$ (since © is not reflexive), the material conditional holds in all the states $S^E_{[A]}$ of any extension E (due to axioms (2) and (3)). A remarkable difference with [30] is that our notion of extension allows to deal with sequences of action applications, rather than with single action steps.

In [31] Mc Cain and Turner define an action language $\mathcal{AC}$ which extends the language $\mathcal{A}$ by adopting the method introduced in [30] for representing causal background knowledge and which is closely related to the language $\mathcal{AR}$ of Giunchiglia, Kartha and Lifschitz [21]. Mc Cain and Turner specify a sound and complete translation of the language $\mathcal{AC}$ into default logic, using the situation calculus. In this translation, frame axioms and assumptions on the initial state are translated to normal defaults, while actions laws and sufficiency propositions (corresponding to causal laws) are translated to defaults without justification. As a difference, in our default translation (see Section 3.3) action laws and causal rules are represented by formulas in a default theory, rather than by default rules. Since causal rules are inference rules, $\mathcal{AC}$, similarly to the formalism in [30], does not capture reasoning by cases. On the other hand, $\mathcal{AC}$ handles action preconditions (qualifications) in a flexible and robust way, although, for a domain description to be translated into default logic, a requirement is needed that action preconditions be expressed in a particular "explicit" form.

In [32] Mc Cain and Turner introduce a very simple and expressive nonmonotonic formalism for modelling causal theories. Causal laws, which behave as nonmonotonic inference rules, are used extensively in the formalization of the action domain. In particular, they are used to formalize ramification constraints, qualification constraints and action laws, to model persistency by inertia, and to make assumptions about the initial state and about action occurrences. The approach in [32] differs from ours (and from all other approaches mentioned above) primarily from the ontological point of view. In fact, in [32] actions are first class objects and action atoms are allowed to occur in causal laws besides fluent atoms. Moreover, in [32] there is an explicit representation of time which, for instance, allows a simple formalization of the scenario given in example 11 and, also, it makes the language well suited for representing delayed effects of actions.

Relation to Thielscher's work

Thielscher [45] presents a STRIPS-like approach augmented by causal rules which are automatically derived from domain constraints by means of "influence information". Based on a propositional language, he defines an action as a triplet (C, a, E), where C is the set of precondition literals, E is the set of result literals and a is the action name, and $\mid C \mid = \mid E \mid$ [7], i.e., the preconditions and the effects refer to the same set of fluent names. This restriction seems to be heavy. For an example, the action of moving an object a from place p to place q cannot be formulated by $(\{atap)\}, movea, \{ataq)\})$, but has to become $(\{atap, \neg ataq\}, movea, \{ataq, \neg atap\})$, even in the presence of the constraint $\neg(atap \wedge ataq)$.

Since constraints formulated in classical logic are too powerful to compute correctly all ramifications, Thielscher proposes a method for automatically generating a set of *causal* rules from a given set of domain constraints, which exploits the information of potential influence of some fluents upon others. What he calls "influence information" is a binary relation I between fluents, where $(f_1, f_2) \in I$ means that f_1 might possibly influence f_2. Given a constraint in clausal form $l_1 \vee \cdots \vee l_n$ and the influence information $(\mid l_i \mid, \mid l_j \mid) \in I$, he generates automatically the causal law

$$\neg l_i \ causes \ l_j \ if \bigwedge_{1 \leq k \leq n, k \neq i, k \neq j} \neg l_k$$

[7] Given a literal l, $\mid l \mid$ denotes its atomic subformula, i.e. $\mid l \mid = l$ if l is a positive literal, and $\mid l \mid = \neg l$ if l is a negative literal. For a set of literals M, $\mid M \mid = \{\mid l \mid : l \in M\}$

For instance, in Example 16, given the the influence relation $I = \{(S, F), (R, E), (E, F), (F, E)\}$, from state constraints (a) and (b) we can get the following set of causal rules:

$$
\begin{array}{ll}
(4''a) & \neg R \textbf{ causes } E \textbf{ if } \neg F \\
(4''b) & \neg F \textbf{ causes } E \textbf{ if } \neg R \\
(5'') & R \textbf{ causes } \neg E \textbf{ if } true \\
(6'') & F \textbf{ causes } \neg E \textbf{ if } true \\
(7''a) & \neg S \textbf{ causes } F \textbf{ if } \neg E \\
(7''b) & \neg E \textbf{ causes } F \textbf{ if } \neg S \\
(8'') & S \textbf{ causes } \neg F \textbf{ if } true \\
(9'') & E \textbf{ causes } \neg F \textbf{ if } true
\end{array}
$$

Notice that, a rule l_i **causes** l_j **if** α could be restated in our modal language as $\alpha \wedge ©l_i \to ©l_j$. As a difference with the causal rules $(4), \ldots, (9)$ we have given in section 3.4, the modality © also occurs in the body of causal rules. This allows to make a distinction between what has been caused by the action and what holds since it persists from the previous state [8]. As a consequence, there are two rules (4"a) and (4"b) instead of the single rule (4).

One of the original contributions of Thielscher's work is this capacity to derive causal rules automatically by means of the influence informations. But it is probably highly problematic to establish such an influence relationship. Thielscher doesn't say anything about the way how influence information is found. He gives the relation I by explicit enumeration supported by the intuitive idea "if $(f_1, f_2) \in I$ then this is intended to denote that a change of f_1's truth value might possibly influence the truth value of f_2". A notion rather similar to that of "influence" was studied by Fariñas and Herzig in [14], where they define an "interference" relation in a very similar way. The work is also related to some work by Dubois et al. [11] who go further and study axiomatics and semantics for dependency relations. There have also been important philosophical discussions about how to rule relevance [16], discussing various approaches within a probabilistic framework. Furthermore, the relationships between dependency and update (and theory revision), which are very important for the use of causality rules within action theories, have extensively been studied, e.g. by Gärdenfors [17] and by Fariñas and Herzig [15].

The effects of the execution of an action (C, a, E) in a state S (states are always complete in Thielscher's framework) are computed by replacing C within S by E (provided that the action is possible, i.e. $C \subset S$) yielding an intermediate result state S' and then by applying causal rules, i.e. by successively adding literals l' to S' for every $l \in S'$ whenever l *causes* l' *if* γ is a causal law and $\gamma \subset S'$. Thielscher's system solves the frame problem in a simple, nice and efficient way, since all what is not changed by an action or by a causal law remains in the state and thus persists.

But two important problems are not solved in a completely satisfactory way. The one is related to causation and the other one to ramification.

As we have pointed out in this paper and illustrated by several examples, causality relations between fluents may also have as an effect to produce nonminimal changes after actions. Thus, in some cases, a fluent which became true *because* another flu-

[8] Of course, in our language the use of the modality © in the body of causal rules would only make sense when immediate effects of actions are also caused, that is when actions laws have the form $2(\pi \to [a]©\rho)$.

ent became true, might not persist when the other fluent subsequenly becomes false. In the example 12, Thielscher would use the causal law "*onTable causes safe*", yielding the state $\{onTable, safe\}$ from $\{\neg onTable, \neg safe\}$ after performing the action $(\{\neg onTable\}, putOn, \{onTable\})$. But when performing subsequently the action $(\{onTable\}, removeFrom, \{\neg onTable\})$, he obtains $\{\neg onTable, safe\}$, unless he could generate a causal law "$\neg$*onTable causes*$\neg$*safe*". But this causal law is simply wrong. Clearly, this insufficiency is related to the fact that Thielscher's action theory in [45] makes no distinction between the fluents which may persist and those which do not persist, when an action is executed.

In a more recent paper [44], which deals with qualification and causality, the action theory of [45] is extended by allowing actions with non-deterministic immediate effects. Using non-deterministic actions the above problem could be modelled by explicitely introducing an action law for $removeFrom$, which says that the execution of this action can make $safe$ either true or false.

Concerning the ramification problem, the problem of example 11 can be represented in Thielscher's framework. In particular, rule (1) of example 11 can be represented by the following causal rule:

$$at(z, r) \textbf{ causes } at(y, next(r)) \textbf{ if } at(y, r).$$

However, it is not obvious at all how a similar problem could be modelled, when two actions "push-right" and "push-left" occur. Indeed, neither action laws with complex preconditions (as those we have used in laws (2) and (3) of example 11) are allowed in Thielscher's framework, nor causal rules can be specifically related to an action.

Given the above correspondence with Mc Cain and Turner's approach, we can also establish a relationship with the causal approach proposed by Thielscher. It has been established in [45] that all the resulting states computed in the fixpoint characterization (by the function Res_C^4) can be obtained through Thielscher's notion of *successor* state, while the converse does not hold. Indeed, there are successor states which do not correspond to any fixpoint. A stable state can be reached through a sequence of unstable states, and the value taken by fluents in these unstable states (though different from their final value) may affect the value of some other fluents. So for instance the fact that a light was on in an intermediate unstable state (which is different from the final stable one), may affect the value of a sensor in the final stable state.

The transition function determined by our notion of extension relates to Thielscher's notion of successor state in the very same way as Mc Cain and Turner's next state function relates to it. As in Mc Cain and Turner's fixpoint characterization, we are not allowed to reason about the unstable states which are intermediate between one state and its successors. Hence there are successor states in Thielscher's approach which cannot be obtained in any of our extensions.

5 Conclusions and relation to other approaches

In this paper we have presented a language for reasoning about actions in which modalities are used for representing actions. In our action theory, which is nonmonotonic, persistency is achieved by maximizing persistency assumptions. The problems of ramification, of reasoning with incomplete initial states and of postdiction are addressed in a uniform way (in particular, a modal causality operator is introduced to represent causal relationships). Moreover, the action theory can also deal with non-minimal change and nondeterministic actions.

Since we have actions as modalities, our action language is strongly related to dynamic logic [23]. As a difference, on the one hand we only allow sequential composition of actions and, on the other hand, we introduce in our action logic other kinds of modalities (as, for instance, the ⓒ modality) which are not present in dynamic logic. In [19], the multimodal language has been extended by introducing composition of modal action operators $[a]$ and $[b]$ to $[a\&b]$, in order to deal with concurrency of actions. This induces a new logical axiom $[a]p \rightarrow [a\&b]p$.

Other papers have proposed to use dynamic logics for reasoning about action and change. In particular, in [36] Prendinger and Schurz propose an approach based on dynamic logic which provides a monotonic solution to the problems related to the frame problem. In particular, they address problems that arise in a multi-agent setting, rather then a single-agent setting, like the *overcommitment* problem. Since it is often impossible to predict what actions other agents will perform, one should not commit to the assumption that such actions have not been performed. Nonmonotonic solutions to the frame problem are not well suited to deal with the overcommitment problem, since they minimize changes of facts, when an action is executed. This is not appropriate in a complex environment, when other agents may contribute to changes in the domain. The solution proposed in [36] requires to state explicitely which facts have to persist when some (complex) action is executed, so that only those facts that are prerequisites or consequences of agent actions are maintained, while other facts are allowed to vary freely. In [36] different kinds of preservation axioms are introduced. In particular, the terminal preservation axioms have the same form $\Box(\alpha \rightarrow [A]\alpha)$ of our persistency axioms, but they also apply to complex actions like, for instance, action sequences (that is A may be any complex action). Though Prendinger and Schurz allow domain constraints in their action language, they do not make use of causal rules in addressing the ramification problem.

In [8] DeGiacomo and Lenzerini define an extension of propositional dynamic logic which constitutes a sort of Principled Monotonic Propositional Situation Calculus extended to deal with complex actions. The logic is well suited to exploit monotonic solutions to the frame problem. It allows to deal with concurrency, nondeterminism, to organize actions in hierarchies, to represent mutual exclusion between actions and to answer to historical queries.

In our language we have adopted a causal approach to ramification. This makes our proposal different from the language $\mathcal{AR}_0$ of Kartha and Lifschitz [24], in which domain constraints of the form **always** α are used to express ramifications rather than causal rules. If we omit causal rules while allowing these domain constraints (which also are allowed in our laguage), we get a language for reasoning about actions which has strong similarities with the languages $\mathcal{AR}_0$ as well as $\mathcal{AR}$ presented in [21]. In particular, our way of dealing with non frame fluents has some relation with the "possibly changes" propositions in [21], which allow to cope with nondeterministic actions (we may say that $(f, a) \notin Frame$ corresponds to saying that a **possibly changes** f **if** $True$).

As a difference with $\mathcal{AR}_0$ and $\mathcal{AR}$, which deal with the ramification problem by using a categorization among fluents, here we use the distinction among frame and non frame fluents to deal with non minimal change and causal rules to deal with ramifications. As shown by Thielscher in [45], in categorization-based approaches, it is sometimes difficult to find the right category for each fluent. For instance, he shows an example which cannot be formalized using two categories of fluents, while it can be formulated by using causal rules.

When our language is restricted to a clausal form and all fluents are taken as being frame with respect to all actions, it becomes equivalent to the modal logic programming language presented in [3], an extension of the language $\mathcal{A}$ [18] in which causal rules are represented by means of directional logic programming clauses (by making use of explicit negation). Both the semantics and the proof procedure of the logic programming language in [3] are defined within an abductive framework, namely the argumentation framework [13, 4].

Starting from Gelfond and Lifschitz' seminal work [18], several proposals have been put forward for representing actions in logic programming, which provides a simple and well studied nonmonotonic mechanisms. In particular, some of these proposals, like for instance [9, 10, 12, 35], also adopt an abductive approach to reasoning about actions. More precisely, [10] and [12] have proposed translation of (extensions of) the language $\mathcal{A}$ into abductive logic programming, while [35] has defined an extension of the language $\mathcal{A}$ to deal with concurrent actions, together with a sound and complete translation to abductive normal logic program. Still in the spirit of the language $\mathcal{A}$, in [1] a language for reasoning about concurrent actions has been defined. Furthermore, in [2] Baral has introduced a language of "state specifications", which is closely related to the language of "revision programs" by Marek and Truszczyński. In the language of state specifications nondeterministic actions, static and dynamic constraints, and action precondition laws can be uniformily specified. Its semantic is given through a translation to disjunctive logic programming.

Acknowledgement We are grateful to the referees for their careful checking of the work and for their helpful comments.

References

[1] C.Baral and M. Gelfond. Representing concurrent actions in extended logic programs. In *Proc. IJCAI'93* , pages 866–871, 1993.

[2] C. Baral. Reasoning about actions: Non-deterministic effects, Constraints, and Qualification. In *Proc. IJCAI'95* , pages 2017–2023, 1995.

[3] M. Baldoni, L.Giordano, A.Martelli and V.Patti. An abductive proof procedure for reasoning about actions in modal logic programming. In JICSLP'96 Workshop on Non-monotonic extensions of Logic Programming, 1996.

[4] A. Bondarenko, F. Toni, R. A. Kowalski. An assumption based framework for non-monotonic reasoning. in *Proc. 2nd Int. Workshop on Logic Programming and Non-monotonic Reasoning*, 1993.

[5] G. Brewka and J. Hertzberg. How to do things with worlds: on formalizing action and plans. In *J. Logic and Computation*, vol.3, no.5, pages 517–532, 1993.

[6] M.O. Cordier and P. Siegel. Prioritized Transitions for Updates. In *Symbolic and Quantitative Approaches to Reasoning and Uncertainty ECSQARU'95*, LNAI 946, Springer Verlag, pages 142–150, 1995.

[7] J. Crawford. Three issues in action. In *Fifth International Workshop on Nonmonotonic Reasoning*, 1994. unpublished Manuscript.

[8] G. De Giacomo, M. Lenzerini. PDL-based framework for reasoning about actions. In *LNAI 992*, pages 103–114, 1995.

[9] M. Denecker, L. Missiaen, M. Bruynooghe. Temporal Reasoning with Abductive Event Calculus. In *Proc. ECAI-92* , pages 384–388, Vienna, 1992.

[10] M. Denecker, D. De Schreye. Representing Incomplete Knowledge in Abductive Logic Programming. In *Proc. ILPS'93* , pages 147–163, Vancouver, 1993.

[11] D. Dubois, L. Fariñas, A. Herzig, and H. Prade. An ordinal view of independence with application to plausible reasoning. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, 1994.

[12] P. M. Dung. Representing Actions in Logic Programming and its Applications to Database Updates. In *Proc. ICLP'93* , pages 222–238, Budapest, 1993.

[13] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning and logic programming. In *Proc. IJCAI93*, pages 852–857, 1993.

[14] L. Fariñas and A. Herzig. Interference logic =conditional logic +frame axiom. *International Journal of Intelligent System*, 9(1):119–130, 1994.

[15] L. Fariñas and A. Herzig. Belief change and dependence. In Y. Shoham, editor, *Proceedings of the 6th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 147–162. Morgan Kaufmann Publishers, 1996.

[16] P. Gärdenfors. On the logic of relevance. *Synthese*, (37):351–367, 1978.

[17] P. Gärdenfors. Belief revision and irrelevance. *PSA*, 2:349–356, 1990.

[18] M. Gelfond and V. Lifschitz. Representing Action and Change by Logic Programs. In *J. Logic Programming*, no. 17, pages 301–321, 1993.

[19] L. Giordano, A. Martelli and C. Schwind. Dealing with Concurrent Actions in Modal Action Logic. In *Proceedings of the 13th European Conference on Artificial Intelligence, ECAI'98*, pages 537–541, 1998.

[20] E. Giunchiglia and V. Lifschitz. Dependent fluents. In *Proc. IJCAI'95*, pages 1964–1969, Montreal, Canada, 1995. Morgan Kaufmann.

[21] E. Giunchiglia, G.C. Kartha and V. Lifschitz. Representing Actions: Indeterminacy and Ramifications. In *Artificial Intelligence*, to appear.

[22] J. Y. Halpern and Y. Moses. A Guide to Completeness and Complexity for Modal Logics of Knowledge and Belief. *Artificial Intelligence*, 54:319–379, 1992.

[23] D. Harel. First order dynamic logic in *Extensions of Classical Logic, Handbook of Philosophical Logic II*, pp. 497–604, 1984.

[24] G.N. Kartha and V. Lifschitz. Actions with Indirect Effects (Preliminary Report). In *Proc. KR'94* , pages 341–350, 1994.

[25] G.N. Kartha and V. Lifschitz. A simple formalisation of Actions Using Circumscription. In *Proc. IJCAI95*, pages 1970–1975, Montreal, Canada, 1995. Morgan Kaufmann.

[26] V. Lifschitz. Frames in the Space of Situations. *Artificial Intelligence Journal*, Vol. 46, pages 365–376, 1990.

[27] F. Lin and R.Reiter. State Constraints Revised. In *J. of Logic and Computation*, vol. 4, no. 5, pages 655–678, 1994.

[28] F. Lin. Embracing Causality in Specifying the Indirect Effects of Actions. In *Proc. IJCAI'95*, pages 1985–1991, 1995.

[29] F. Lin. Embracing Causality in Specifying the Inditerminate Effects of Actions. In *Proc. AAAI'96*, pages 670–676, 1996.

[30] N. McCain and H. Turner. A Causal Theory of Ramifications and Qualifications. In *Proc. IJCAI'95*, pages 1978–1984, 1995.

[31] N. McCain and H. Turner. Representing actions in logic programs and default theories. A situation calculus approach. In *J. of Logic Programming*, vol. 31, no. 1-3, pages 245–298, 1997.

[32] N. McCain and H. Turner. Causal theories of action and change. In *Proc. AAAI'97*, pages 460–465, 1997.

[33] L.T. McCarty. Modalities over actions, I. Model Theory. In *Proc. KR'94* , pages 437–448, 1994.

[34] K. Myers and D. Smith. The persistence of derived information. In *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 496–500, 1988.

[35] R. Li and L.M. Pereira. Representing and Reasoning about Concurrent Actions with Abductive Logic Programs. To appear in *Annals of Mathematics and AI*, 1996.

[36] H. Prendinger and G. Schurz, Reasoning about Action and Change. A Dynamic Logic Approach. In *Journal of Logic, Language, and Information.* Vol.5, N.2, pages 209–245, 1996.

[37] R. Reiter. A Logic for Default Reasoning. *Artificial Intelligence*, Vol. 13, no. 1, pages 81–132, 1980.

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

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

[40] E. Sandewall. Systematic comparison of approaches to ramification using restricted minimization of change. IDA Technical Report 1995, Linköping University.

[41] C. B. Schwind. A Logic Based Framework for Action Theories. In *Language, Logic and Computation*, (J. Ginzburg, Z. Khasidashvili, C. Vogel, J.-J. Lévy and E. Vallduví eds.), pp. 275-291. CSLI publication, Stanford, USA.

[42] L.A. Stein and L. Morgenstern. Motivated Action Theory: a formal theory of causal reasoning. Artificial Intelligence 71, pp. 1–42, 1996.

[43] M. Thielscher. An Analysis of Systematic Approaches to Reasoning about Actions and Change. In *Proc. AIMSA94*, pages 195–204, 1994.

[44] M. Thielscher. Qualification and Causality. Technical report ICSI TR-96-26. A shorter version in *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR'96)*, Cambridge, MA, 1996, pp. 51–62.

[45] M. Thielscher. Ramification and Causality. Artificial Intelligence Journal, vol. 89, No. 1-2, pp. 317-364, 1997.

Appendix

A Proof of Theorem 10

Theorem 10 Let $(\Pi, Frame)$ be a domain frame. Let $\Delta = (\Pi, D)$ be the associated $AD-$default theory. Then E is an extension for $(\Pi, Frame)$ if and only if E is a default extension of Δ.

Proof To prove this result, recall that an extension E of a default theory (W, D) is a solution of the fixed-point equation $\Gamma(E) = E$, where $\Gamma(S)$ is the smallest set satisfying

(1) $W \subseteq \Gamma(S)$

(2) $Th(\Gamma(S)) = \Gamma(S)$

(3) if $\dfrac{\alpha : \beta}{\gamma} \in D$ and $\alpha \in \Gamma(S)$ and $\neg\beta \notin S$, then $\gamma \in \Gamma(S)$

Let us also recall the following theorem adapted from Reiter [37]:

If E is a default extension of an $AD-$default theory $\Delta = (W, D)$ then $E = Th_{AD}(W \cup \{\gamma : \dfrac{\alpha : \beta}{\gamma} \in D \text{ and } \alpha \in E \text{ and } \neg\beta \notin E\})$.

In order to omit confusion within this proof, we will speak of *extension* in the case of a default theory and of *action extension* otherwise (in the case of an extension as we defined it for action theories within this paper). In the following, given a set of $AD-$defaults D, we define $Cons(D) = \{\gamma : \dfrac{\alpha : \beta}{\gamma} \in D\}$ is the set of consequents of $AD-$defaults of D.

"$\Rightarrow$":

Let E be an extension for $(\Pi, Frame)$. By definition 5, $E = Th_{AD}(\Pi \cup WP' \cup F)$ and $WP' \subseteq WP_{Frame}$ and $F \subseteq Ass_{NonFrame}$, such that

(a) $[A](l \rightarrow [a]l) \in WP'$ iff $[A; a]\neg l \notin E$

(b) $[A]l \in F$ iff $[A]\neg l \notin E$

Observe that the elements of WP' and F are all consequents of defaults of the associated default theory $\Delta = (\Pi, D)$, hence $E = Th(\Pi \cup Cons(D'))$, for some $D' \subseteq D$. In order to show that E is an extension of Δ, we have to show that $E = \Gamma(E)$. In order to prove that $\Gamma(E) \subseteq E$, we will show that E verifies (1) to (3) above.

(1) and (2) are clearly true for E by the definition of action extension. For (3), let be $d \in D$. Then either $d = \dfrac{: \neg[A; a]\neg l}{[A](l \rightarrow [a]l)}$ or $d = \dfrac{: \neg[A]\neg l}{[A]l}$. In the first case, let be $[A; a]\neg l \notin E$, then by (a), $[A](l \rightarrow [a]l) \in WP'$, hence $[A](l \rightarrow [a]l) \in E$. In the second case, let be $[A]\neg l \notin E$, then by (b), $[A]l \in F$, hence $[A]l \in E$. E verifies conditions (1), (2) and (3), therefore by the minimality of Γ, $\Gamma(E) \subseteq E$.

To show that $E \subseteq \Gamma(E)$, observe first, that, clearly, $\Pi \subseteq \Gamma(E)$ by (1). For default consequences, consider $d \in D'$. Then either $d = \dfrac{: \neg[A; a]\neg l}{[A](l \rightarrow [a]l)}$ or $d = \dfrac{: \neg[A]\neg l}{[A]l}$. In the first case, from $[A](l \to [a]l) \in E$ we conclude $[A;a]\neg l \notin E$, by (a) and hence by (3) and since $d \in D$, $[A](l \to [a]l) \in \Gamma(E)$. In the second case, from $[A]l \in E$, we conclude that $[A]\neg l \notin F$ by (b) and from (3), we obtain that $[A]l \in \Gamma(E)$.

"$\Leftarrow$":
Immediate consequence of Reiter's Theorem mentioned above [37].

Notice that in the above proof the properties of the logic AD are not used. This should not be surprising, since the proof establishes a relationship between two notions of extensions (default extensions and action extensions) which are both based on the same logic AD.

B Proof of Theorem 19

In order to prove Theorem 19, we need some Propositions. In the following we will assume that in action laws $\Box(\pi \to [a]\rho)$ and in causal rules $\Box(\alpha \to \copyright\beta)$, π, ρ, α and β are non-modal formulas. Moreover, we will assume that all fluents are frame fluents, that is, $Frame = \{(l,a) : a \in A, l \in \mathcal{F}\}$.

The following Proposition shows that, in an extension E, each state can be determined from the previous one by making use of the persistency laws in that state together with action laws and causal rules. This means that an extension can be built step by step, from the initial one.

Proposition 9 Let $E = Th(\Pi \cup WP' \cup F)$ be an extension of $(\Pi, Frame)$ and let

- $S = S^E_{[a_1;\ldots;a_n]}$ be a consistent state;
- $\Pi|_{a_1;\ldots;a_n;a} = \{\pi \to [a]\rho \ : \ \Box(\pi \to [a]\rho) \in \Pi\} \cup$
  $\{[a](\alpha \to \copyright\beta) \ : \ \Box(\alpha \to \copyright\beta) \in \Pi\}$.
- $WP'|_{a_1;\ldots;a_n;a} = \{L \to [a]L \ : [a_1;\ldots;a_n](L \to [a]L) \in WP'\}$.

Then

$$E \vdash_{AD} [a_1;\ldots;a_n;a]L \quad \Leftrightarrow \quad S \cup \Pi|_{a_1;\ldots;a_n;a} \cup WP'|_{a_1;\ldots;a_n;a} \vdash_{AD} [a]L.$$

Proof Roughly speaking, $\Pi|_{a_1;\ldots;a_n;a}$ is the set of action laws and causal rules that can be applied in the state S when action a is executed. $WP'|_{a_1;\ldots;a_n;a}$ is the set of persistency laws applied in the state S when a is executed. The proposition claims the state $S^E_{[a_1;\ldots;a_n;a]}$ obtained from S by executing a is determined by these two sets of formulas.

Note that, the sequence $a_1;\ldots;a_n$ might be the empty sequence ε in the case $n = 0$. By soundness and completeness of our logic, we can prove the thesis by proving that

$$E \models [a_1;\ldots;a_n;a]L \quad \Leftrightarrow \quad S \cup \Pi|_{a_1;\ldots;a_n;a} \cup WP'|_{a_1;\ldots;a_n;a} \models [a]L.$$

"$\Leftarrow$": obvious

"$\Rightarrow$": We prove the contrapositive implication. Let us assume that $S \cup \Pi|_{a_1;\ldots;a_n;a} \cup WP'|_{a_1;\ldots;a_n;a} \not\models [a]L$. Then, there is model $M = \langle W, R, V\rangle$, and a world $w \in W$ such that, for $\Gamma = S \cup \Pi|_{a_1;\ldots;a_n;a} \cup WP'|_{a_1;\ldots;a_n;a}$,

$$M, w \models \Gamma \text{ and } M, w \not\models [a]L.$$

In particular, there is a $\tilde{w}$ such that $wR_a\tilde{w}$ and $M, \tilde{w} \not\models L$. We want to show that $E \not\models [a_1; \ldots; a_n; a]L$.

Since S is a consistent state (i.e. $E \not\models [a_1; \ldots; a_n]\bot$) there is a model $M' = \langle W', R', V' \rangle$ and a world w_0, such that $M', w_0 \models E$ and $M', w_0 \not\models [a_1; \ldots; a_n]\bot$. Hence, $M', w_0 \models \langle a_1; \ldots; a_n \rangle true$, and, therefore, there is a sequence of worlds $w_1, \ldots, w_n$ such that $w_0 R_{a_1} w_1 \ldots w_{n-1} R_{a_n} w_n$.

Note that $W', w_n \models S$. In fact, since, for all $l \in S$, $[a_1; \ldots; a_n]l \in E$ and $M', w_0 \models E$, we have that $M', w_0 \models [a_1; \ldots; a_n]l$, and thus $M', w_n \models l$.

We want to define a new model M'' by modifying M' in such a way that the effects of an action a at the world w_n are the same as at the world w of M. We can assume, without loss of generality, that the sets of worlds W and W' are disjoint. Let $M'' = \langle W'', R'', V'' \rangle$ be a new model obtained from M' by adding to W' all those worlds of M that are reachable from w through R_a. More precisely, let $N = \{w' : w' \in W,\ (w, w') \in R_a\}$ be the set of worlds from M, we want to add to those of M'. We define M'' as follows:

- $W'' = W' \cup N$;
- The accessibility relations in M'' are defined as follows:

  $R''_b = R'_b$, for all actions $b \neq a$;

  $R''_a = R'_a \cup \{(w_n, w') \ : \ w' \in N\}$;

  $R''_{\copyright} = R'_{\copyright} \cup \{(w', w'') \ : \ w', w'' \in N \text{ and } (w', w'') \in R_{\copyright}\}$;

  $R''_{2} = R'_{2} \cup \{(w', w'') : \ w' \in W'' \text{ and } w'' \in N \text{ and } (w', w'') \in (\bigcup R''_{a_i})^*\}$;

  $R''_{\varepsilon} = R'_{\varepsilon}$.
- The valuation function V'' is defined as V' on the worlds in W' and as V on the worlds in N.

Essentially, we have added to M'' the accessibility relations $R''_{\copyright}$ among the worlds in N as they are defined in M.

The fact that the accessibility relations R_{a_i}, R_ε and $R_{\copyright}$ are not required to be serial is essential to construct the model M'' suitably. In particular, notice that by construction there are no worlds accessible from those in N through the accessibility relations R_{a_i} and R_ε.

It is easy to show that M'' is a Kripke model for our logic, that is, the accessibility relations in M'' satisfy the right properties. As an example we show that the property $R''_a \subseteq R''_a \circ R''_{\copyright}$ is satisfied.

Assume that $(w', w'') \in R''_a$. If $w', w'' \in W'$, then $(w', w'') \in R'_a$ and, since $R'_a \subseteq R'_a \circ R'_{\copyright}$ holds for the model M', we get $(w', w'') \in R'_a \circ R'_{\copyright}$, and hence $(w', w'') \in R''_a \circ R''_{\copyright}$. Otherwise, $w'' \in N$ and $w' = w_n$. In such a case, $(w, w'') \in R_a$ and, since $R_a \subseteq R_a \circ R_{\copyright}$ holds for M, we have $(w, w'') \in R_a \circ R_{\copyright}$. Therefore, there is a world w''' such that $(w, w''') \in R_a$ and $(w''', w'') \in R_{\copyright}$. By construction, $w''' \in N$ and thus (since $w' = w_n$) $(w', w''') \in R''_a$ and $(w''', w'') \in R''_{\copyright}$. Therefore, $(w', w'') \in R''_a \circ R''_{\copyright}$.

Note that, since S holds in M'' at the world w_n, and S is a complete state (see Theorem 8), the valuation of literals in M'' at world w_n has to be the same as the valuation of literals in M at world w (in which S also holds).

We have now to prove that $M'', w_0 \models E$, i.e. $M'', w_0 \models \Pi \cup WP' \cup F$ (note that, since all fluents are frame, F contains only assumptions on the initial state). From this, since by construction of the model M'', $M'', w_0 \not\models [a_1; \ldots; a_n; a]L$, we have that that $E \not\models [a_1; \ldots; a_n; a]L$, which proves our thesis.

Let us now prove that $M'', w_0 \models E$.

(a) First we show that $M'', w_0 \models \Pi$. For each action law $\Box(\pi \rightarrow [b]\rho)$ and causal rule $\Box(\alpha \rightarrow ⓒ\beta)$ contained in Π, we have to show that,

> for all $w' \in W''$, such that $(w_0, w') \in R''_{\Box}$, $M'', w' \models \pi \rightarrow [b]\rho$ and $M'', w' \models \alpha \rightarrow ⓒ\beta$.

There are three cases.

1. Assume that $w' \notin N$ and $w' \neq w_n$. Since $w' \in W'$, $(w_0, w') \in R'_{\Box}$ and $M', w_0 \models E$, we get $M', w' \models \pi \rightarrow [b]\rho$ and $M', w' \models \alpha \rightarrow ⓒ\beta$. Therefore, by construction of M'', since $w' \neq w_n$ and $w' \notin N$, $M'', w' \models \pi \rightarrow [b]\rho$ and $M'', w' \models \alpha \rightarrow ⓒ\beta$.
2. If $w' \in N$, then there are only worlds reachable from w' with the relation $R_{ⓒ}$. Hence, vacuously, $M'', w' \models \pi \rightarrow [b]\rho$, for all actions b. Moreover, if $(w', w'') \in R''_{ⓒ}$ then, by construction, in M $(w', w'') \in R_{ⓒ}$. Since $M, w \models \Pi|_{a_1;\ldots;a_n;a}$, then $M, w \models [a](\alpha \rightarrow ⓒ\beta)$, for all causal laws $\Box(\alpha \rightarrow ⓒ\beta)$. Therefore, given that $w' \in N$ and $(w, w') \in R_a$, $M, w' \models \alpha \rightarrow ⓒ\beta$. Hence, if $M, w' \models \alpha$, for all w'' such that $(w', w'') \in R_{ⓒ}$ $M, w'' \models \beta$.

   What we want to show is that if $M'', w' \models \alpha \rightarrow ⓒ\beta$. Assume that $M'', w' \models \alpha$. Then, since the valuation function V'' is defined as V on the worlds in N and α is non modal, $M, w' \models \alpha$ and, therefore, for all $w'' \in W$ such that $(w', w'') \in R_{ⓒ}$ $M, w'' \models \beta$. This holds, in particular, for those $w'' \in N \subseteq W$, which are the worlds accessible from w' via $R''_{ⓒ}$. Since these worlds in M'' have the same propositional valuation as in M, we have that for all $w'' \in W''$ such that $(w', w'') \in R''_{ⓒ}$ $M'', w'' \models \beta$.
3. Let us consider the case $w' = w_n$. Again since $M', w_0 \models E$, for all action laws and causal rules $M', w' \models \pi \rightarrow [b]\rho$ and $M', w' \models \alpha \rightarrow ⓒ\beta$.

   Note that the only worlds accessible from w_n in M'' and not in M' are those accessible through R_a, Hence, we have that, for all actions $b \neq a$, $M'', w' \models \pi \rightarrow [b]\rho$, and, for all causal rules, $M'', w' \models \alpha \rightarrow ⓒ\beta$. We have to prove that, for all action laws for a, $M'', w' \models \pi \rightarrow [a]\rho$. Let us assume that $M'', w' \models \pi$. We have to prove that for all $w'' \in W''$ such that $(w', w'') \in R''_a$ $M'', w'' \models \rho$. There are two cases, either $w'' \in W'$ or $w'' \in N$.

   If $w'' \in W'$ then , $(w', w'') \in R'_a$. From the fact that $M', w' \models \pi \rightarrow [a]\rho$, it follows that $M', w'' \models \rho$ and, hence, $M'', w'' \models \rho$. If $w'' \in N$, then $(w', w'') \in R_a$ in M. Since $M, w \models \Pi|_{a_1;\ldots;a_n;a}$, then $M, w \models \pi \rightarrow [a]\rho$. But, given that $M, w \models \pi$ (since w in M has the same propositional evaluation as w_n in M''), $M, w'' \models \rho$, and, therefore, $M'', w'' \models \rho$.

(b) We show that $M'', w_0 \models WP'$, that is, for all $[b_1; \ldots; b_k](L \rightarrow [b]L) \in WP'$, $M'', w_0 \models [b_1; \ldots; b_k](L \rightarrow [b]L)$. We have to prove that, for any sequence of worlds $w_1, w_2, \ldots, w_k$, such that $(w_{i-1}, w_i) \in R_{b_i}$, for $i = 1, \ldots, k$, $M'', w_k \models L \rightarrow [b]L$. If $w_k \neq w_n$ or $b \neq a$ this can be concluded from the fact that $M', w_0 \models WP'$. If $w_k = w_n$ and $b = a$, the proof is similar to the case (a)3 above and it makes use of the fact that $M, w \models WP'|_{a_1;\ldots;a_n;a}$.

(c) To show that $M'', w_0 \models F$ we can observe that F is a set of observations on the initial state, having the form $[\varepsilon]\alpha$, $M'', w_0 \models [\varepsilon]\alpha$ iff for all w' such that $(w_0, w') \in R''_\varepsilon$ $M'', w' \models \alpha$. We can conclude this from the hypothesis that $M', w_0 \models F$.

Proposition 20 Let S be a consistent state, R be a set of formulas of the form $\gamma \rightarrow [a]\delta$ (where γ is a non modal formula which may be empty, and δ is any formula), and let

$$Cons(S, R) = \{[a]\delta \ : \ \gamma \rightarrow [a]\delta \in R \text{ and } \gamma \in Th_{PC}(S)\}$$

(where $Th_{PC}(S)$ is the set of propositional consequences of S in classical logic). Then, for all literals L,

$$S \cup R \vdash_{AD} [a]L \quad \text{iff} \quad Cons(S, R) \vdash_{AD} [a]L.$$

Proof "$\Leftarrow$": obvious

"$\Rightarrow$": Let us assume that $Cons(S, R) \not\vdash_{AD} [a]L$. Then there is model $M = \langle W, R, V\rangle$, and a world $w \in W$ such that, $M, w \models Cons(S, R)$ and $M, w \not\models [a]L$. Hence, there is a world $w^* \in W$ such that $wR_a w^*$ and $M, w^* \not\models L$.

Let $M' = \langle W', R', V'\rangle$ be a new model obtained from M by adding to W a new world w_0. More precisely, let:

- $W' = W \cup \{w_0\}$;
- For all $b \neq a$, $R'_b = R_b$;

  $R'_a = R_a \cup \{(w_0, w') \ : \ w' \in W \ and \ (w, w') \in R_a\}$;

  $R'_{©} = R_{©}$;

  $R'_\varepsilon = R_\varepsilon$

  $R'_{2} = R_{2} \cup \{(w_0, w') : \ w' \in W' \text{ and } (w_0, w') \in (\bigcup R'_{a_i})^*\}$;
- The valuation function V' is defined as in M on all worlds in W and, on the new world w_0, as follows: $V(w_0, p) = true$ if $p \in S$ and $V(w_0, p) = false$ if $\neg p \in S$ (remember that S is a consistent and complete state).

It is easy to show that M' is a Kripke model for our logic. First let us prove that, for all atomic actions b, $R'_b \subseteq R'_b \circ R'_{©}$. We know that $R_b \subseteq R_b \circ R_{©}$ holds for M. Since in building M' we have only added the world w_0 and we have modified the accessibility relations for connecting w_0 to other worlds, for all pairs $(w', w'') \in R'_b$, if $w' \neq w_0$ then $(w', w'') \in R_b$. Hence, $(w', w'') \in R_b \circ R_{©} \subseteq R'_b \circ R'_{©}$. If $w' = w_0$, then, $b = a$ and, by construction, $(w, w'') \in R_a$. Therefore, there is a $w_1 \in W$ such that $(w, w_1) \in R_a$ and $(w_1, w'') \in R_{©}$. By construction, $(w_0, w_1) \in R_a$, and, since $R'_{©} = R_{©}$, $(w_1, w'') \in R'_{©}$. Hence, $(w_0, w'') \in R'_A \circ R'_{©}$.

We also show that the that R_2' is transitive, that is $R_2' \circ R_2' \subseteq R_2'$. Let $(x, y) \in R_2'$ and $(y, z) \in R_2'$. We want to show that $(x, z) \in R_2'$. If $x, y, z \in W$, this easily comes from the fact that $(x, y) \in R_2$ and $(y, z) \in R_2$ and R_2 is transitive. If $x = w_0$ and $y = w_0$, then we have $(w_0, w_0) \in R_2'$ and $(w_0, z) \in R_2'$. Hence, $(x, z) \in R_2'$. If $x = w_0$ and $y \in W$, then from $(x, y) \in R_2'$ we get that there is a world $v \in W$ such that $(x, v) \in R_a'$ and and $(v, y) \in R_2'$. Hence, $(v, y) \in R'_{a_{i_1}} \circ \ldots \circ R'_{a_{i_k}}$ for some $k \geq 0$. Since $(y, z) \in R'_{a_{i_{k+1}}} \circ \ldots \circ R'_{a_{i_m}}$ for some $m - k \geq 0$, then $(v, z) \in R'_{a_{i_1}} \circ \ldots \circ R'_{a_{i_m}}$ for some $m \geq 0$. Therefore, $(x, z) \in R_a' \circ R'_{a_{i_1}} \circ \ldots \circ R'_{a_{i_m}} \in (\bigcup R'_{a_i})^*$ and thus $(x, z) \in R_2'$.

All the other properties of accessibility relations in M' can be proved in a similar way.

We have now to prove that $M', w_0 \models S \cup R$. From this, since by construction $M', w_0 \not\models [a]L$, we have that $S \cup R \not\models [a]L$, and hence (by soundness) $S \cup R \not\vdash_{AD} [a]L$. Note that $M', w_0 \models S$ by construction of M' (for the way we have defined the valuation function in w_0). We have to show that, for each rule $\gamma \rightarrow [a]\delta \in R$,

$$M', w_0 \models \gamma \rightarrow [a]\delta. \tag{5}$$

There are two cases. Either $M', w_0 \not\models \gamma$, and hence (5) holds trivially; or $M', w_0 \models \gamma$. In this case, $\gamma \in Th_{PC}(S)$, and then $[a]\delta \in Cons(S, R)$. As hypothesis, we have that $M, w \models Cons(S, R)$ and, in particular, $M, w \models [a]\delta$, i.e., for all $w' \in W$, such that $wR_a w'$, $M, w' \models \delta$. Let us now note that all worlds reachable from w_0 with R_a' in M' are exactly the ones reachable from w with R_a in M, and that from these worlds the same worlds are accessible in M and in M' through the same accessibility paths. Hence, we have $M', w_0 \models [a]\delta$. So we have proved that $M', w_0 \models S \cup R$.

Notice that in the previous proposition, since for $\gamma \rightarrow [a]\delta \in R$, δ is an arbitrary formula and γ may be the formula *true*, so that a formula $\gamma \rightarrow [a]\delta$ in R may have the form $[a](\alpha \rightarrow \copyright\beta)$.

The next proposition shows that causal rules in our language have a similar role as inference rules in [30].

Proposition 21 Let Γ be a finite set of non modal formulas and C a set of causal laws of the form $\alpha \Rightarrow \beta$ (α and β are non modal formulas). Let $\Phi = \{[a]\delta : \delta \in \Gamma\} \cup \{[a](\alpha \rightarrow \copyright\beta) : \alpha \Rightarrow \beta \in C\}$. Then, for all literals L,

$$\Gamma \vdash_C L \ \Rightarrow \ \Phi \vdash_{AD} [a]L$$

where $\vdash_C$ is standard derivability relation in propositional logic extended to deal with causal rules, as defined in [30].

Proof Let us assume that $\Gamma \vdash_C L$. Following McCain and Turner definition, L belongs to the smallest set of formulas containing Γ that is closed with respect to propositional logic and closed under C. A set H is said to be closed under C if, for every rule $\alpha \Rightarrow \beta \in H$, if $\alpha \in H$, then $\beta \in H$.

Based on the sequence of applications of the causal rules in C which are needed to derive L from Γ, a modal (Hilbert-style) proof can be constructed for $[a]L$ from Φ.

Assume that $\Gamma \vdash_C L$ and that L can be derived from Γ by applying the causal laws

$$\alpha_1 \Rightarrow \beta_1, \ \ldots, \ \alpha_m \Rightarrow \beta_m$$

in the order.

Then, denoting by $\vdash_{PC}$ the standard derivability relation in the propositional calculus, we have

$$\begin{array}{l}
\Gamma \vdash_{PC} \alpha_1 \\
\Gamma \cup \{\beta_1\} \vdash_{PC} \alpha_2 \\
\Gamma \cup \{\beta_1, \ldots, \beta_{m-1}\} \vdash_{PC} \alpha_m \\
\Gamma \cup \{\beta_1, \ldots, \beta_m\} \vdash_{PC} L.
\end{array}$$

Hence,

$$\begin{array}{l}
\vdash_{PC} \Gamma \rightarrow \alpha_1 \\
\vdash_{PC} \Gamma \wedge \beta_1 \rightarrow \alpha_2 \\
\vdash_{PC} \Gamma \wedge \beta_1 \wedge \ldots \wedge \beta_{m-1} \rightarrow \alpha_m \\
\vdash_{PC} \Gamma \wedge \beta_1 \wedge \ldots \wedge \beta_m \rightarrow L.
\end{array}$$

By the necessitation rule and by axiom (K) for the modality $[a]$:

$$\begin{array}{l}
\vdash_{AD} [a]\Gamma \rightarrow [a]\alpha_1 \\
\vdash_{AD} [a]\Gamma \wedge [a]\beta_1 \rightarrow [a]\alpha_2 \\
\vdash_{AD} [a]\Gamma \wedge [a]\beta_1 \wedge \ldots \wedge [a]\beta_{m-1} \rightarrow [a]\alpha_m \\
\vdash_{AD} [a]\Gamma \wedge [a]\beta_1 \wedge \ldots \wedge [a]\beta_m \rightarrow [a]L,
\end{array}$$

where $[a]\Gamma$ is the conjunction of formulas $[a]\delta$ for δ in Γ. Since such formulas are formulas in Φ:

$$\begin{array}{ll}
\text{(i)} & \Phi \vdash_{AD} [a]\alpha_1 \\
\text{(ii)} & \Phi \vdash_{AD} [a]\beta_1 \rightarrow [a]\alpha_2 \\
\text{(iii)} & \Phi \vdash_{AD} [a]\beta_1 \wedge \ldots \wedge [a]\beta_{m-1} \rightarrow [a]\alpha_m \\
\text{(iv)} & \Phi \vdash_{AD} [a]\beta_1 \wedge \ldots \wedge [a]\beta_m \rightarrow [a]L.
\end{array}$$

Given that Φ contains the formulas $[a](\alpha_j \rightarrow \copyright\beta_j)$, for all $j = 1, \ldots, m$, we have

$$\Phi \vdash_{AD} [a](\alpha_j \rightarrow \copyright\beta_j),$$

for all $j = 1, \ldots, m$, and hence, by axioms (1) and (10)

$$\Phi \vdash_{AD} [a]\alpha_j \rightarrow [a]\beta_j$$

for all $j = 1, \ldots, m$. From (i) and $\Phi \vdash_{AD} [a]\alpha_1 \rightarrow [a]\beta_1$ we get, by propositional reasoning, $\Phi \vdash_{AD} [a]\beta_1$, and similarly for the cases (ii) and (iii). Hence we obtain:

$$\begin{array}{ll}
\text{(i)} & \Phi \vdash_{AD} [a]\beta_1 \\
\text{(ii)} & \Phi \vdash_{AD} [a]\beta_1 \rightarrow [a]\beta_2 \\
\text{(iii)} & \Phi \vdash_{AD} [a]\beta_1 \wedge \ldots \wedge [a]\beta_{m-1} \rightarrow [a]\beta_m \\
\text{(iv)} & \Phi \vdash_{AD} [a]\beta_1 \wedge \ldots \wedge [a]\beta_m \rightarrow [a]L.
\end{array}$$

Thus, by propositional reasoning, we get:

$$\Phi \vdash_{AD} [a]L.$$

Note that the vice-versa of the result in proposition 21 does not hold. As counterexample take $\Phi = \{[a](p \vee q),\ [a](q \rightarrow ©s),\ [a](p \rightarrow ©s)\}$. We have $\Phi \vdash_{AD} [a]s$, but, for $\Gamma = \{p \vee q\}$ and $C = \{q \Rightarrow s,\ p \Rightarrow s\}$, we have $\Gamma \not\vdash_C s$. In fact, Mc Cain and Turner's causal laws, are represented as inference rules and they do not permit reasoning by cases. However, we can prove the following weaker proposition.

Proposition 22 Let S be a consistent state, let C be a set of causal rules, Eff a set of immediate effects and $S' \in Res^4_C(\mathrm{Eff}, S)$. Let $[a]CausalRules = \{[a](\beta \rightarrow ©\alpha) : \beta \Rightarrow \alpha \in C\}$. For all literals L, if

$$[a](S \cap S') \cup [a]Eff \cup [a]CausalRules \vdash_{AD} [a]L$$

then

$$(S \cap S') \cup \mathit{Eff} \vdash_C L.$$

Proof The proposition can be proved by contraposition. Let us assume that $(S \cap S') \cup \mathit{Eff} \not\vdash_C L$. We show that there is a Kripke interpretation M and a world w such that

$M, w \models [a](S \cap S') \cup [a]Eff \cup [a]CausalRules$ and
$M, w \not\models [a]L$,

and hence that $[a](S \cap S') \cup [a]Eff \cup [a]CausalRules \not\models [a]L$ (from which by the soundness of the axiomatization we get $[a](S \cap S') \cup [a]Eff \cup [a]CausalRules \not\vdash_{AD} [a]L$).

We define a model M with two worlds w and w' such that:

$V(w', p) = true$ iff $p \in S'$, for each atomic proposition p;
$R_a = \{(w, w')\}$ and $R_b = \emptyset$, for all $b \neq a$;
$R_© = \{(w', w')\}$;
$R2 = \{(w, w), (w, w'), (w', w')\}$;
$R_\varepsilon = \emptyset$.

Remember that S' is a consistent and complete set of literals, and hence it determins a propositional interpretation (namely the valuation at the world w'). The propositional valuation for the world w can be chosen arbitrarily.

We can easily see that M is a Kripke model for our logic and, in particular, that all the properties on the accessibility relations are satisfied.

Notice that, from the hypothesis, $L \notin S'$ and, hence, $M, w' \not\models L$ and $M, w \not\models [a]L$. To prove our thesis, we have prove that $M, w \models [a](S \cap S') \cup [a]Eff \cup [a]CausalRules$. In fact, $M, w' \models S \cap S'$, since, by construction, $M, w \models S'$. $M, w' \models Eff$, otherwise $Eff \cup S'$ would be inconsistent and the state $S' = \{L : (S \cap S') \cup \mathit{Eff} \vdash_C L\}$ would be inconsistent too, which contradicts our hypothesis. Finally, for each causal rule $[a](\beta \rightarrow ©\alpha) \in [a]CausalRules$, $M, w' \models \beta \rightarrow ©\alpha$. If $M, w' \models \beta$, since β is propositional, it must be in the deductive closure of S', and hence $(S \cap S') \cup \mathit{Eff} \vdash_C \beta$, i.e., it belongs to the smallest set of formulas containing $(S \cap S') \cup \mathit{Eff}$ and closed with respect to propositional logic and closed under C. Given that, $\beta \Rightarrow \alpha \in C$, then it must be that $(S \cap S') \cup \mathit{Eff} \vdash_C \alpha$ and, hence, $\alpha \in S'$. Therefore, $M, w' \models \alpha$.

Theorem 19 Let S be a consistent state, let C be a set of causal rules, Eff a set of immediate effects and $S' \in Res^4_C(\mathit{Eff}, S)$. The domain frame $(\Pi, Frame)$, where

$$\Pi = \{[a]Eff\} \cup \{\Box(\alpha \to ©\beta) \ : \ \alpha \Rightarrow \beta \in C\}$$

and

$$Frame = \{(l, a) : a \in A, l \in \mathcal{F}\},$$

has a restricted extension E on the sequence $\varepsilon; a$, such that $S^E_{[\varepsilon]} = S$ and $S^E_{[\varepsilon;a]} = S'$.

Proof Let S be a consistent state, C a set of causal rules, *Eff* a set of immediate effects and $S' \in Res^4_C(\mathit{Eff}, S)$. We have to show that the domain frame $(\Pi, Frame)$, where

$$\Pi = \{[a]Eff\} \cup \{\Box(\alpha \to ©\beta) \ : \ \alpha \Rightarrow \beta \in C\}$$

and

$$Frame = \{(l, a) : a \in \mathcal{A}, l \in \mathcal{F}\},$$

has a restricted extension E on the sequence $\varepsilon; a$, such that $S^E_{[\varepsilon]} = S$ and $S^E_{[\varepsilon;a]} = S'$.

We define $E = Th(\Pi \cup WP' \cup F)$ by taking:

$F = \{[\varepsilon]L \ : \ L \in S\}$ and
$WP' = \{[\varepsilon](L \to [a]L) \ : \ L \in S'\}$.

We have to prove that E is an extension and that $S^E_{[\varepsilon]} = S$ and $S^E_{[\varepsilon;a]} = S'$, that is, for all literals L:

(i) $L \in S$ iff $[\varepsilon]L \in E$;
(ii) $L \in S'$ iff $[\varepsilon; a]L \in E$.

We will first prove (i) and (ii). Then, we will prove that E is a restricted extension.

To prove (i), let us assume that $L \in S$, then, by construction, $[\varepsilon]L \in F$, and hence $[\varepsilon]L \in E$. Conversely, if $L \notin S$, then we want to show that $[\varepsilon]L \notin E$. We have to find a model M and a world w such that $M, w \models E$ and $M, w \not\models [\varepsilon]L$. We define M so that it contains a single world w and

$V(w, p) = \mathit{true}$ iff $p \in S$, for each atomic proposition p;
$R_b = \emptyset$, for all atomic actions b;
$R_© = \emptyset$;
$R2 = \{(w, w)\}$;
$R_\varepsilon = \{(w, w)\}$.

It is easy to see that M is a Kripke interpretation for our logic. Moreover, by construction, $M, w \not\models [\varepsilon]L$. We have to show that $M, w \models E$, i.e., $M, w \models Th(\Pi \cup WP' \cup F)$. We have that, for all $[\varepsilon]L' \in F$, $M, w \models [\varepsilon]L'$, since by construction $M, w \models L'$, for all $L' \in S$. Moreover, for all $[\varepsilon](L' \to [a]L') \in WP'$, $M, w \models [\varepsilon](L' \to [a]L')$ holds trivially, since from the only world accessible trough R_ε, that is w, there are no worlds accessible through R_a. Similarly, $M, w \models \Box[a]Eff$ holds trivially, since there are no R_a accessible worlds from w, and, for each causal rule $\Box(\alpha \to ©\beta) \in \Pi$, $M, w \models \Box(\alpha \to ©\beta)$ holds trivially, since there are no $R_©$ accessible worlds from w. Thus $M, w \models \Pi$ and $M, w \models E$. On the other hand, by construction, since $L \notin S$, $M, w \not\models [\varepsilon]L$. Therefore, $E \not\models [\varepsilon]L$.

Let us prove (ii). Remember that $E = Th(\Pi \cup WP' \cup F)$. We have that $[\varepsilon; a]L \in E$ if and only if

$$E \vdash_{AD} [\varepsilon; a]L \tag{6}$$

if and only if, by Proposition 9,

$$S \cup \Pi|_{\varepsilon;a} \cup WP'|_{\varepsilon;a} \vdash_{AD} [a]L. \tag{7}$$

Note that $\Pi|_{\varepsilon;a}$ contains the formula $[a]Eff$ and, the formulas $[a](\alpha \rightarrow \copyright\beta)$, for all causal rule $\Box(\alpha \rightarrow \copyright\beta)$. Moreover, $WP'|_{\varepsilon;a}$ contains all the formulas $L' \rightarrow [a]L'$ such that $[\varepsilon](L' \rightarrow [a]L') \in WP'$. Hence, Proposition 20 can be applied and from it we get that (7) holds if and only if

$$Cons(S, \Pi|_{\varepsilon;a} \cup WP'|_{\varepsilon;a}) \vdash_{AD} [a]L. \tag{8}$$

Let us denote the causal formulas $[a](\alpha \rightarrow \copyright\beta)$ in $\Pi|_{\varepsilon;a}$ by $[a]CausalRules$. Then we have:

$$Cons(S, \Pi|_{\varepsilon;a}) = [a]Eff \cup [a]CausalRules. \tag{9}$$

Moreover, we can prove that:

$$Cons(S, WP'|_{\varepsilon;a}) = [a](S \cap S'). \tag{10}$$

First, we prove that if $L' \in (S \cap S')$, then $[a]L' \in Cons(S, WP'|_{\varepsilon;a})$. If $L' \in (S \cap S')$, then $L' \in S$ and $L' \in S'$. By construction, $[\varepsilon](L' \rightarrow [a]L') \in WP'$. Thus $L' \rightarrow [a]L' \in WP'|_{\varepsilon;a}$, and hence $[a]L' \in Cons(S, WP'|_{\varepsilon;a})$.

Conversely, let us assume that $[a]L' \in Cons(S, WP'|_{\varepsilon;a})$. Then it must be that $L' \rightarrow [a]L' \in WP'|_{\varepsilon;a}$ and that $L' \in S$. Moreover, by definition of $WP'|_{\varepsilon;a}$, it must be that $L' \in S'$. Therefore, $L' \in (S \cap S')$.

Given (9) and (10), (8) holds if and only if

$$[a]Eff \cup [a]CausalRules \cup [a](S \cap S') \vdash_{AD} [a]L, \tag{11}$$

Moreover, by Propositions 21 and 22, (11) holds if and only if

$$(S \cap S') \cup \mathit{Eff} \vdash_C L, \tag{12}$$

that is, if and only if $L \in S'$ (since $S' \in Res^4_C(\mathit{Eff}, S)$). Hence we have proved that $[\varepsilon; a]L \in E$ iff $L \in S'$. This concludes the proof of (ii).

Now we have to prove that E is a restricted extension of the domain frame $(\Pi, Frame)$ on the sequence $\varepsilon; a$. This can be proved easily by making use of (i) and (ii). More precisely, we have to prove that, the following conditions hold:

(a) $[\varepsilon](L \rightarrow [a]L) \in WP'$ iff $[\varepsilon; a]\neg L \notin E$

(b) $[\varepsilon]L \in F$ iff $[\varepsilon]\neg L \notin E$.

Let us consider first the case (a). *(If)* Assume that $[\varepsilon; a]\neg L \notin E$. Then, by (ii), $\neg L \notin S'$. Since S' is a complete state, $L \in S'$ and, by definition of WP', $[\varepsilon](L \rightarrow [a]L) \in WP'$. *(Only if)* We prove the contrapositive: if $[\varepsilon; a]\neg L \in E$ then $[\varepsilon](L \rightarrow [a]L) \notin WP'$. From $[\varepsilon; a]\neg L \in E$, by (ii), we get $\neg L \in S'$. By the consistency of S', $L \notin S'$. Hence, for the definition of WP', $[\varepsilon](L \rightarrow [a]L) \notin WP'$.

Let us now prove (b). *(If)* If $[\varepsilon]\neg L \notin E$, then, by the definition of E, $[\varepsilon]\neg L \notin F$. By (i), $\neg L \notin S$ and, since S is a complete set of literals $L \in S$. Then, by construction, $[\varepsilon]L \in F$. *(Only if)* We prove the contrapositive: if $[\varepsilon]\neg L \in E$ then $[\varepsilon]L \notin F$. From $[\varepsilon]\neg L \in E$ by (i) we get $\neg L \in S$. By the consistency of S, $L \notin S$ and, by (i), $[\varepsilon]L \notin E$. Thus, $[\varepsilon]L \notin F$.