Towards a Conditional Logic of Actions and Causation

Laura Giordano
Dipartimento di Informatica
Università del Piemonte Orientale
Corso Borsalino 54 - 15100 Alessandria, Italy
E-mail: laura@di.unito.it

Camilla Schwind
MAP, UMR CNRS - MCC N 694
183 Avenue de Luminy, Case 901
13288 Marseille Cedex 9, France,
E-mail: Camilla.Schwind@gamsau.map.archi.fr

Abstract

In this paper we present a new approach to reason about actions and causation which is based on a conditional logic. The conditional implication is interpreted as causal implication. This makes it possible to formalize in a uniform way causal dependencies between actions and their immediate and indirect effects. Furthermore, this new approach provides a natural formalization of concurrent actions and causal dependencies between actions. An abductive semantics is adopted for dealing with the frame problem.

1 Introduction

Causality plays a prominent role in the context of reasoning about actions, as the ramification effects of actions can be regarded as causal dependencies. Many approaches for reasoning about actions have been proposed which allow causal dependencies to be captured [5, 13, 16, 21, 1]. Schwind [18] has studied how causal inferences have been integrated and used in action theories by analyzing four formalisms, which are approaches to action and causality, and comparing them with respect to criteria she established for causality. Namely, the article analyses Lin's approach [12, 13], McCain and Turner's causal theory for action and change [16, 22], Thielscher's theory of ramification and causation [21], and Giordano, Martelli and Schwind's dynamic causal action logic [6]. More recently, Zhang and Foo [23, 2, 3] propose to extend propositional dynamic logic, where actions are modalities, by introducing modalities which are propositions. Sentence "ϕ causes ψ" is represented by the formula $[\phi]\psi$, where $[\phi]$ is a new modality. Note that this representation corresponds to a conditional logic approach, since EPDL formula $[\phi]\psi$ is interpreted as conditional formula $\phi > \psi$. Zhang and Foo's approach has the merit of providing a clean representation of causation as well as a uniform representation of direct and indirect effects of actions. But the notion of causality as defined in EPDL is too strong since it entails material implication. This property, $[\phi]\psi \rightarrow (\phi \rightarrow \psi)$, comes from a crucial axiom introduced for actions in EPDL, and is motivated by the fact that when "ϕ causes ψ" then the state constraint $\phi \rightarrow \psi$ must also hold. While we agree with this argument, we may object that the status of causal laws (and consequently of material implications) in [23] is not that of domain constraints. Instead, causal laws are regarded as domain axioms, which gives the corresponding material implications also the status of domain axioms, so that they can be used for inference in all possible states. In particular, this allows a contrapositive use of material implications $\phi \rightarrow \psi$, which may lead to unintended conclusions. Therefore, though EPDL preserves the directionality of causation (from $[\phi]\psi$ we cannot conclude that $[\neg\psi]\neg\phi$), the fact that causation entails material implication (from $[\phi]\psi$ we can conclude that $\phi \rightarrow \psi$ and thus that $\neg\psi \rightarrow \neg\phi$) anyhow leads to unwanted conclusions when reasoning about the effects of actions. Another difference with our present approach is due to the different modelization of actions. In EPDL, syntactically, actions are not formulas. This makes it impossible to combine assertions about actions with assertions about causality. Concurrent actions cannot be constructed from single actions. It is not possible to express for example that action a and fact B cause the effect C.

We propose to represent causality by a binary logical operator. Our causality operator has the properties of causality relations as discussed in [18] and -perhaps more important- does not have some of the undesired (or doubtful) properties many other approaches have. Our causal operator is not monotonic and does not entail material implication which makes it weaker than the one proposed in [23]. Traditionally, considering a conditional as a causal implication has frequently attracted the attention of logicians in conditional logic ([10, 11]. It allows to model both causal laws and action laws: the causal law "ϕ causes ψ" is represented by the conditional formula $\phi > \psi$ and the action law "action a

causes proposition ψ" is represented by the conditional formula $do(a) > \psi$, where $do(a)$ is a special atomic proposition associated with each action a. This uniform representation of the causal relationship between actions and their results as well as between facts and their effects gives us a great flexibility for handling both concepts in an simple way when representing actions. For example, in this setting, concurrent execution of actions is naturally modelled by conjunctions of the form $do(a_1) \wedge \ldots \wedge do(a_n)$ in the antecedents of conditionals. It is also very natural to express dependency (and independency) relations between actions and actions, actions and propositions, etc.

2 The Causal Action Logic AC

The language, $\mathcal{L}_>$, of our action logic is that of propositional logic augmented with a conditional operator $>$. The set of propositional variables in $\mathcal{L}_>$, Var, includes the set $\{do(a) : a \in \Delta_0\}$, where Δ_0 is a set of *elementary actions* including the "empty" action ϵ. Formulas are defined as usual and the modalities $\Box$ and $\Diamond$ are defined by $\Box A \equiv (\neg A > \bot)$ and $\Diamond A \equiv \neg\Box\neg A$. Intuitively, $\Box A$ means that A necessarily holds, while $\Diamond A$ means that A is possible. If A is an action proposition $do(a)$, $\Diamond do(a)$ means that a is executable. In order to express that a proposition *always* holds, we introduce the additional operator $\overline{\Box}$: $\overline{\Box} A \equiv (\Box A \wedge A)$. The following is an axiom system for logic AC.

We define an axiom system for logic AC as follows:

Definition 1 (AC) *The conditional logic AC is the smallest logic containing the following axioms and deduction rules:*

(CLASS) All classical propositional axioms and inference rules

(CV) $(\neg(A > \neg C) \wedge (A > B)) \rightarrow (A \wedge C) > B)$

(CA) $(A > C) \wedge (B > C) \rightarrow ((A \vee B) > C)$

(CE) $(do(a) > B) \wedge (do(a) > (B > C)) \rightarrow (do(a) > C)$ *where* $a \in \Delta_0$

(MOD) $\Box A \rightarrow (do(a) > A)$ *where* $a \in \Delta_0$

(RCEA) if $\vdash A \leftrightarrow B$, *then* $\vdash (A > C) \equiv (B > C)$

(RCK) if $\vdash A_1 \wedge \ldots \wedge A_n \rightarrow B$, *then* $\vdash (C > A_1) \wedge \ldots \wedge (C > A_n) \rightarrow (C > B)$.

Note that all axioms and inference rules are standard in conditional logics and, in particular, they belong to the axiomatization of Lewis's logic VC (see [17]). However, we have excluded several of the standard axioms of conditional logics such as (ID), (MP) and (CS) since they model unwanted properties of causality. Reflexivity axiom (ID) $A > A$ is excluded since a proposition (or an action) should not cause itself. Axiom (MP) $(A > B) \rightarrow (A \rightarrow B)$ should not hold for causal implication, because we do not want to be able to infer the material implication from a causal rule, since the former is stronger and would give undesired properties to the latter. (CS) $A \wedge B \rightarrow (A > B)$ should not be a property of causal implication since A and B could both hold conjunctively without A being a cause of B.

Axiom (CE) allows action laws and causal laws to interact, it provides the chain effects between causal laws and action laws. (CE) says that the causal consequences of action effects are in turn action effects: if executing action a (in a state S) causes B in the next state S' ($do(a) > B$), and the causal law $B > C$ holds in S', then executing action a (in S) causes C to hold in S'. (CE) weakens (MP) as it is clear form the following formulation of (CE) $(do(a) > (B > C)) \rightarrow ((do(a) > B) \rightarrow (do(a) > C))$, which is obtained from (MP) by (RCK). (CE) has similarities with the property of transitivity (TRANS) $(do(a) > B) \wedge (B > C) \rightarrow (do(a) > C)$[1], which however requires that the causal law $B > C$ holds in the state S in which the action is executed. As we will see in our action theory causal laws do not necessarily hold in all possible states, as they may have preconditions which make them hold in some states only.[2]

(MOD) allows to deduce $\Box A \rightarrow (do(a_1) > \ldots > (do(a_n) > A)\ldots)$ for any finite sequence of actions $a_1, \ldots a_n$ ($n \geq 0$) including the empty sequence and meaning that a formula A which is true in every state is also true after the occurrence of any finite sequence of actions. So, the subsequent occurrence of actions structures a world and its subsequent states according to time, although time is not represented explicitly in our formalism.

Entailment $\vdash$ is defined as usual and given a set of formulas E, the deductive closure of E is denoted by $Th(E)$. AC is characterized semantically in terms of selection function models.

Definition 2 *An AC-structure* M *is a triplet* $\langle W, f, [[\,]]\rangle$, *where W is a non-empty set, whose elements are called possible worlds,* f, *called* the selection function, *is a function of type* $\mathcal{L}_> \times W \rightarrow 2^W$, $[[]]$, *called the evaluation function, is a function of type* $\mathcal{L}_> \rightarrow 2^W$ *that assigns a subset of W,*

[1]For standard conditional logic with reflexivity, adding TRANS would collapse the conditional to material implication. But this is not the case for our causal action logic AC, since reflexivity $A > A$ is not an axiom.

[2]As an example of causal law with precondition consider the following one: $\overline{\Box}(at(y, r) \rightarrow (at(z, r) > at(y, next(r)))$ used in Example 4 below, which says that if block y is at r then moving block z to position r causes y to move to a next position.

$[[A]]$ *to each formula* A*. The following conditions have to be fulfilled by* $[[\ ]]$*:*

(1) $[[A \wedge B]] = [[A]] \cap [[B]]$
(2) $[[\neg A]] = W - [[A]]$[3]
(3) $[[A > B]] = \{w : f(A, w) \subseteq [[B]]\}$

We assume that the selection function f *satisfies the following properties which correspond to the axioms of our logic AC:*

(S-RCEA) if $[[A]] = [[B]]$ *then* $f(A, w) = f(B, w)$

(S-CV) if $f(A, w) \cap [[C]] \neq \emptyset$ *then* $f(A \wedge C, w) \subseteq f(A, w)$

(S-CA) $f(A \vee B, w) \subseteq f(A, w) \cup f(B, w)$

(S-CE) if $f(do(a), w) \subseteq [[B]]$
then $f(do(a), w) \subseteq f(B, f(do(a), w))$

(S-MOD) if $f(B, w) \cap [[do(a)]] \neq \emptyset$
then $f(do(a), w) \neq \emptyset$*,*

where $a \in \Delta_0$ *and* $f(B, f(do(a), w))$ *represents the set of worlds* $\{z \in f(B, x) : x \in f(do(a), w)\}$.

We say that a formula A is true in a AC-structure $M = \langle W, f, [[]]\rangle$ if $[[A]] = W$. We say that a formula α is AC-valid ($\models A$)if it is true in every AC-structure. We also introduce the following notation $S \models_M A$ to say that, given a AC-structure M, a set of formulas S and a formula A, for all $w \in M$ if $w \in [[B]]$ for all $B \in S$, then $w \in [[A]]$.

The above axiom system is sound and complete with respect to this semantics.

Theorem 1 $\models A$ *iff* $\vdash A$

The completeness proof is shown by the canonical model construction [19] and can be found in the appendix. Moreover, the axiomatization is consistent and the logic is decidable. Since the logic AC is weaker than VC, each VC-structure is an AC-structure, which shows that the logic AC is ”non-trivial” in some sense.

3 Action Theories

3.1 Domain descriptions

We use atomic propositions $f, f_1, f_2, \ldots \in Var$ for *fluent names*. A *fluent literal*, denoted by l , is 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 small greek letters $\alpha, \beta, \ldots$ any formula not containing conditional formulas.

We define a *domain description* as a tuple (Π, $Frame_0$, Obs). Π is a set of *action laws*, *causal laws*, *precondition laws*, *domain constraints* and *causal independency constraints*.

Action laws have the form:

$$\overline{\Box}(\pi \to (do(a) > \rho)),$$

for an action a with precondition π and effect ρ: executing action a in a state where π holds causes ρ to hold in the resulting state. For action laws with no precondition, i. e. $\pi = true$, we just obtain $\overline{\Box}(do(a) > \rho)$.

Causal laws have the form:

$$\overline{\Box}(\pi \to (\alpha > \beta)),$$

meaning that “if π holds, then α causes β”.

Precondition laws have the form:

$$\overline{\Box}(\pi \equiv \neg(do(a) > \bot)),$$

meaning that “action a is executable iff π holds”. According to the definition of $\Diamond$, this is equivalent to $\overline{\Box}(\pi \equiv \Diamond do(a))$.

Domain constraints include formulas of the form:

$$\overline{\Box}\alpha,$$

(meaning that “α always holds”).

Causal independency constraints have the form:

$$\overline{\Box}(\neg(A > \neg B)),$$

meaning that A does not cause $\neg B$ (that is, B might be true in a possible situation caused by A). In particular, when the above constraints concern action execution, we have $\Box\neg(do(a) > \neg do(b))$, meaning that the execution of action a does not prevent action b from being executed (does not interfere with its execution). Note that as a consequence of this constraint we have, by (CV), that $(do(a) > C) \to (do(a) \wedge do(b) > C)$, namely, the effects of action a are also effects of the concurrent execution of a and b, as a does not interfere with b. Moreover, from $(do(a) > \bot) \to (do(a) \wedge do(b) > \bot)$, we have that if a is not executable it cannot be executed concurrently with b.

$Frame_0$ is a set of pairs $(f, do(a))$, where $f \in \mathcal{F}$ is a fluent and $a \in \Delta_0$ is an elementary action, meaning that f is a *frame fluent* for action a, that is, f is a fluent to which persistency applies when a is executed. Fluents which are

[3] Using the standard boolean equivalences, we obtain $[[A \vee B]] = [[A]] \cup [[B]]$, $[[A \to B]] = (W - [[A]]) \cup [[B]]$, $[[\top]] = W$, $[[\bot]] = \emptyset$.

non-frame with respect to a do not persist and may change in a nondeterministic way when a occurs.

The set $Frame_0$ defines a sort of *independence* relationship between elementary actions and fluents. It is closely related to dependency (and influence) relations that have been used and studied by several authors including Thielscher [21], Giunchiglia and Lifschitz [7], and Castilho, Gasquet and Herzig [14]. We use $Frame_0$ for defining persistency rules of the form $A_1 > \ldots > A_n > (l \rightarrow (do(a) > l))$ for every literal l, such that $(|l|, a) \in Frame_0$. These persistency rules behave like *defaults*: they belong to an "action extension" whenever no inconsistency arises. The $Frame_0$-relationship is extended to concurrent actions. Let us denote by $Frame$ the extension of $Frame_0$ to concurrent actions. (i) $Frame_0 \subset Frame$; (ii) If $(f, do(a_1)), \ldots, (f, do(a_n)) \in Frame$ then $(f, do(a_1) \wedge \ldots \wedge do(a_n)) \in Frame$.

Obs is a set of observations about the value of fluents in different *states* which we identify with action sequences. Though our language does not provide an explicit representation of time, as we abandon (MP), time can be embedded in the operator $>$. Given the properties of $>$ we assume a delay between happening of an action and occurrence of its effects, while we do not assume any delay between causes and their effects in causal laws. Observations are formulas of the form: $A_1 > \ldots > A_j > \alpha$ (where each A_i is a possibly concurrent action formula of the form $do(a_1) \wedge \ldots \wedge do(a_n)$), meaning that α holds after the concurrent execution of the actions in A_1, then those in A_2, ..., then those of in A_n. So, every acction occurrence leads from one state to the new state. In particular, we assume an *initial state* characterized by the occurrence of the empty action ϵ. If Obs contains observations α about fluents in the initial state this is written as $do(\epsilon) > \alpha$[4].

Sometimes, when we do not want to consider observations, we will then use the notion of *domain frame*, which is a pair $(\Pi, Frame_0)$.

Let us consider the following example from [23], which formalizes an electrical circuit with two serial switches.

Example 1 There is a circuit with two switches and a lamp. If both switches are on, the lamp is alight. One of the switches being off causes the lamp not to be alight. There are two actions of toggling each of the switches. The domain description is the following (for $i = 1, 2$):

Π: $\overline{\Box}(\neg sw_i \rightarrow (do(tg_i) > sw_i))$
$\overline{\Box}(sw_i \rightarrow (do(tg_i) > \neg sw_i))$
$\overline{\Box}(sw_1 \wedge sw_2 > light)$
$\overline{\Box}(\neg sw_1 \vee \neg sw_2 > \neg light)$
$\overline{\Box}(\neg(do(tg_1) > \neg do(tg_2)))$
$\overline{\Box}(\neg(do(tg_2) > \neg do(tg_1)))$
Obs: $do(\epsilon) > (\neg sw_1 \wedge \neg sw_2 \wedge \neg light)$
$Frame_0 = \{(f, a) : a \in \Delta_0, f \in \mathcal{F}\}$.

The first two rules in Π describe the immediate effects of the action of toggling a switch. The third and forth rule are causal laws which describe the dependencies of the light on the status of the switches. The last two laws are constraints saying that the two actions tg_1 and tg_2 do not interfere. All fluents are supposed to be persistent and the actions tg_1 and tg_2 are independent. As we will see, from the above domain description we can derive $do(tg_1) > \neg light$, $do(tg_1) > do(tg_2) > light$ and $do(tg_1) \wedge do(tg_2) > (sw_1 \wedge sw_2 \wedge light)$ (as actions $do(tg_1)$ and $do(tg_i)$ are independent).

Observe that we could have avoided introducing $\neg light$ in the initial state, as it can be derived, for instance, from $\neg sw_1$: from $do(\epsilon) > \neg sw_1$ and the forth action law we can derive $do(\epsilon) > \neg light$ by (CE).

The axiom (CA) makes it possible to deduce consequences of actions even when it is not deterministically known which action occurs.

Example 2 If the temperature is low, then going to swim causes you to get a cold. If you have no umbrella, then raining causes you to get cold. We have the following domain description:

Π: $\overline{\Box}(cold \rightarrow (do(swim) > get\_cold))$
$\overline{\Box}(no\_umbrella \rightarrow (do(rain) > get\_cold))$
Obs: $do(\epsilon) > (cold \wedge no\_umbrella)$
$Frame_0 = \{(f, a) : a \in \Delta_0, f \in \mathcal{F}\}$

From this theory, we can derive $do(swim) \vee do(rain) > get\_cold$.

The following example is inspired by one discussed by Halpern and Pearl in [8]. Suppose that heavy rain occurred in April causing wet forests in May and electrical storms in May and June. The lightning in May did not cause forest fires since the forests were still wet. But the lightning in June did since the forest dried in the meantime. Pearl and Halpern argue that the April rain caused that the fire did not occur in May and occurred in June instead. We think rather that the April rain prevents the fire from occurring in may which is expressed by the precondition of the action law.

Example 3 i is ranging over months, such that $i + 1$ is the month following i.

Π: $\overline{\Box}(do(rain_i) > wet\_forest_{i+1})$

[4] In the following, when identifying a state with an action sequence $A_1, \ldots, A_j$, we will implicitly assume that $A_1 = do(\epsilon)$. Also, in a conditional formula $A_1 > \ldots > A_j > \alpha$, we will assume that $A_1 = do(\epsilon)$.

$\overline{\Box}(do(rain_i) > (do(lightning_{i+1}) \wedge do(lightning_{i+2})))$
$\overline{\Box}(wet\_forest_i \rightarrow (do(lightning_i) > \neg forest\_fire_i))$
$\overline{\Box}(\neg wet\_forest_i \rightarrow (do(lightning_i) > forest\_fire_i))$
$\overline{\Box}(do(sun_i) > \neg wet\_forest_{i+1})$
$\overline{\Box}(\neg(do(sun_i) > \neg do(rain_j)))$ *(for $i \neq j$)*
$\overline{\Box}(\neg(do(rain_j) > \neg do(sun_i)))$ *(for $i \neq j$)*
Obs: $\{\}$
$Frame_0 = \{(f,a) : a \in \Delta_0, f \in \mathcal{F}\}$.

The second action law allows to say that rain in April causes electrical storms in May and June.

From the first two action laws we derive $do(rain_{April}) > (wet\_forest_{May} \wedge do(lightning_{May}) \wedge do(lightning_{June}))$. Then by (CE), from the third action law (for $i = May$) we get $do(rain_{April}) > \neg forest\_fire_{May}$.

Moreover, it hods that $\overline{\Box}(do(sun_{May}) > \neg wet\_forest_{June})$ (fifth action law) and, as actions $do(rain_{April})$ and $do(sun_{May})$ are independent, we can derive $do(rain_{April}) \wedge do(sun_{May}) > (\neg wet\_forest_{June} \wedge do(lightning_{June}))$. Then, by (CE), from the forth action law (for $i = June$), we conclude: $do(rain_{April}) \wedge do(sun_{May}) > forest\_fire_{June}$.

Therefore, we have that $do(rain_{April}) \wedge do(sun_{May}) > \neg forest\_fire_{May} \wedge forest\_fire_{June}$.

The following example, taken from [5], involves causal laws with preconditions.

Example 4 Consider the following scenario, where a number of blocks are in a sequence: 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. 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 $next(p)$.

Π: $\overline{\Box}(at(x,p) \rightarrow (do(push(x,p)) > at(x, next(p))))$
$\overline{\Box}(at(y,r) \wedge (at(z,r) > at(y, next(r))))$, for $z \neq y$
$\overline{\Box}(at(x,p) > \neg at(x,q))$, for $p \neq q$
Obs: $do(\epsilon) > (at(a,p_1) \wedge at(b,p_2) \wedge at(c,p_3))$
$Frame_0 = \{(f,a) : a \in \Delta_0, f \in \mathcal{F}\}$

The first (action) law says that, if block x is in p, pushing x from the place p moves it to the next place $next(p)$. The second (causal) law says that if block y is at r then moving block z to position r causes y to move to a next position. The third laws causes block x not to be at q if it is at p (different from q).

Given the initial state, it holds that: $do(push(a,p_1)) > (at(a,p_2) \wedge at(b,p_3) \wedge at(c,p_4))$.

3.2 Extensions for a domain description

In order to address the frame problem, we introduce a set of persistency laws, which can be assumed in each extension. Persistency laws are essentially frame axioms. 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 are *defeasible*. They are regarded as assumptions to be maximized. Changes in the world are minimized by maximizing these assumptions. Moreover, persistency laws have to be assumed if this does not lead to inconsistencies.

Let $A_1, \ldots, A_n$ be (possibly) concurrent actions of the form $do(a_1) \wedge \ldots \wedge do(a_m)$ (for $m = 1$ we have an atomic action). We introduce a set of *persistency laws* of the form $A_1 > \ldots > A_n > (l \rightarrow (A > l))$ for every sequence of (concurrent) actions $A_1, \ldots, A_n$ and for every fluent literal l which is a frame fluent with respect to the (concurrent) action A (according to the definition of $Frame$ in the last subsection), that is, for every fluent literal l which is frame for *every elementary action* in A. The persistency law says, that, "if l holds in the state obtained by executing the sequence of actions $A_1, \ldots, A_n$, then l persists after executing action A in that state" [5].

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 $Frame$ of frame fluents, the set of persistency assumptions $WP_{A_1,\ldots,A_n}$ is defined as follows:

$$WP_{A_1,\ldots,A_n} = \{A_1 > \ldots > A_n > (l \rightarrow (A > l))) : (|l|, A) \in Frame\}.$$

Note that the set of persistency assumptions has been defined relative to a sequence of (concurrent) actions, that is, a state.

In addition to persistency assumptions, 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 a concurrent action A 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, as well as assumptions for all fluents in the initial state. Given a set $Frame$ of frame fluents, we define the set of assumptions $Ass_{A_1,\ldots,A_n}$ (relative to a sequence $A_1, \ldots, A_n$) as follows:

$$Ass_{A_1,\ldots,A_n} = \{A_1 > \ldots > A_n > l : (|l|, A_n) \notin Frame\} \cup \{do(\epsilon) > l : l \in Lit\}$$

[5] Notice that introducing persistency laws of the form $\overline{\Box}(l \rightarrow (A > l))$ wouldn't be enough to deal with the persistency of literals at each different state.

We represent a generic assumption in this set by $A_1 > \ldots > A_n > l$, which includes assumptions on the initial state (for $n = 0$).

Now we can introduce our notion of extension, first for domain frames $(\Pi, Frame_0)$, and then for domain descriptions $(\Pi, Frame_0, Obs)$. An extension E of a domain frame is obtained by augmenting Π by as many as possible persistency laws, such that E is consistent. We define an extension *relative to a state*, which can be identified by the sequence of actions $A_1, \ldots, A_n$ leading to that state.

Definition 3 *An* extension *of a domain frame* $D = (\Pi, Frame)$ *relative to the action sequence* $A_1, \ldots, A_n$ *is a set* $E = Th(\Pi \cup WP' \cup F)$*, such that* $WP' \subseteq WP_{A_1,\ldots,A_n}$, $F \subseteq Ass_{A_1,\ldots,A_n}$ *and*

a) *if* $A_1 > \ldots > A_{n-1} > (l \rightarrow (A_n > l)) \in WP_{A_1,\ldots,A_n}$ *then:*
$A_1 > \ldots > A_{n-1} > (l \rightarrow (A_n > l)) \in WP'$
$\iff A_1 > \ldots > A_n > \neg l \notin E$

b) *if* $A_1 > \ldots > A_n > l \in Ass_{A_1,\ldots,A_n}$ *then*
$A_1 > \ldots > A_n > l \in F \iff A_1 > \ldots > A_n > \neg l \notin E$.

The $\Rightarrow$-part of condition a) is a consistency condition, which guarantees that a persistency axiom $A_1 > \ldots > A_{n-1} > (l \rightarrow (A_n > l))$ cannot be assumed in WP' if $\neg l$ can be deduced as an immediate or indirect effect of the action A_n. We say that the formula $A_1 > \ldots > A_n > \neg l$ 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 $A_1 > \ldots > A_n > \neg l$ 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. The conditions above have a clear similarity with the applicability conditions for a default rule in an extension.

Definition 4 E *is an* extension for a domain description $(\Pi, Frame, Obs)$ relative to the action sequence $A_1, \ldots, A_n$ *if it is an extension for the domain frame* $(\Pi, Frame)$ *relative to the action sequence* $A_1, \ldots, A_n$ *and* $E \vdash Obs$.

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. As a difference with [4, 5] an extension only describes a single course of actions, and assumptions are localized to that sequence of actions. In this way, we deal with concurrent actions without the need of introducing two different modalities for actions, which in [4] are called open and closed modalities and they are introduced to avoid that the (AND) law ($do(a) > C \rightarrow do(a) \wedge do(b) > C$) is applied to the non-monotonic consequences of actions, derived by means of the persistency assumptions. In our present approach, we can derive $do(a) \wedge do(b) > C$ from $do(a) > C$ using the axiom (CV) provided a and b are independent. Independency is formulated in the action language by $\neg(do(a) > \neg do(b)) \wedge \neg(do(b) > \neg do(a))$.

Let us consider again Example 1. Relative to the action sequence $\{do(\epsilon)\}$, $\{do(tg_1)\}$, $\{do(tg_2)\}$ we get one extension E containing the frame laws

$$\neg light \rightarrow (do(tg_1) > \neg light,$$
$$\neg sw_2 \rightarrow (do(tg_1) > \neg sw_2,$$
$$do(tg_1) > (sw_1 \rightarrow (do(tg_2) > sw_1)),$$

in which the following sentences hold:

(1) $do(tg_1) > \neg light$,
(2) $do(tg_1) > (do(tg_2) > light)$,
(3) $do(tg_1) \wedge do(tg_2) > light$,
(4) $do(tg_1) > ((do(tg_1) \wedge do(tg_2)) > \neg light)$.

An extension E relative to $A_1, \ldots, A_n$ determines an initial state and a transition function among the states obtained by executing actions $A_1, \ldots, A_n$. In particular, the *state* reachable through an action sequence $A_1, \ldots, A_j$ ($0 \leq j \leq n$) in E can be defined as :

$$S^E_{A_1,\ldots,A_j} = \{l : E \vdash A_1 > \ldots > A_j > l\}$$

(where S^E_ε represents the initial state). Due to condition (b) of definition 3, we can prove that each state $S^E_{A_1;\ldots;A_j}$ is *complete*: for each fluent f, it contains either f or $\neg f$. Moreover, it can be shown 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.

Referring to Example 1,the extension E above relative to the action sequence $\{do(\epsilon)\}$, $\{do(tg_1)\}$, $\{do(tg_2)\}$ determines the following states:

$$S^E_\varepsilon = \{\neg sw_1, \neg sw_2, \neg light\}$$
$$S^E_{\{do(\epsilon)\},\{do(tg_1)\}} = \{sw_1, \neg sw_2, \neg light\}$$
$$S^E_{\{do(\epsilon)\},\{do(tg_1)\},\{do(tg_2)\}} = \{sw_1, sw_2, light\}$$

Observe that for the domain description in Example 1 we do not obtain the unexpected extension in which $do(tg_1) >$

$(do(tg_2) > (\neg sw_1 \wedge sw_2 \wedge \neg light)$ holds: we do not want to accept that toggling sw_2 in the state $\{sw_1, \neg sw_2, \neg light\}$ mysteriously changes the position of sw_1 and lets $\neg light$ persist. To avoid this extension it is essential that causal rules are directional (see [1, 15, 12, 21]). Indeed, the causal rules in Π are different from the constraint $\overline{\Box}((sw_1 \leftrightarrow sw_2) \rightarrow light)$ and, in particular, they do not entail the formula $\neg sw_1 \wedge \neg light \rightarrow sw_2$. As observed in [12] and [21], though this formula must be clearly true in any state, it should not be applied for making causal inferences. In our formalism, contraposition of causal implication is ruled out by the fact that the conditional $>$ is not reflexive: from $\overline{\Box}(\alpha > \beta)$ and $\overline{\Box}\neg\beta$ we cannot conclude $\overline{\Box}\neg\alpha$. On the other hand, it is easy to see that, in any state of any extension, if $\alpha > \beta$ holds, and α holds, β also holds.

Our solution to the frame problem is an abductive solution and is very different from the solution proposed for EPDL in [3]. There persistency laws of the form $l \rightarrow [a]l$ are added explicitly at every state. In EPDL, persistency laws are not global to an extension but they have to be added state by state, according to which action is expected. In our theory, the frame problem is solved globally by minimizing changes modulo causation. As a further difference, in [3] unexpected solutions can be obtained by adding persistency laws as above to the domain description. As observed by Zhang and Foo (see [3], Example 4.1) in the circuit example above the state $S_1 = \{sw_1, \neg sw_2, \neg light\}$ has two possible next states under action $toggle_2$, namely $S_2' = \{sw_1, sw_2, light\}$ and $S_2'' = \{\neg sw_1, sw_2, \neg light\}$. The second one is unexpected.

This behaviour is a side effect of (MP), which holds for EPDL and allows the material implication to be derived form the causal implication. To overcome this problem, Zhang and Foo propose an alternative approach to define the next-state function which makes use of a fixpoint property in the style of McCain and Turner's fixpiont property [15]. Their definition employs the causal operator for determining whether the indirect effects of the action are caused by its immediate effects together with the unchanged part of the state, according to the causal laws. It has to be observed, that this definition of the next state function does not require any integrated use of causal laws and action laws in the theory. In fact, "if the direct effects of an action have been given, $EPDL^-$ [that is, the logic obtained from EPDL when the set of action symbols is empty] is enough to determine how effects of actions are propagated by causal laws" [3]. On the contrary, our solution to the frame problem in the conditional logic CA relies on an integrated use of action laws and causal laws to derive conclusion about actions effects.

A domain description may have extensions containing a formula $A > \bot$. Consider the following example also mentioned by [15]:

Π: $\overline{\Box}(do(a) > p)$ $\overline{\Box}(q > \neg p)$
Obs: $do(\epsilon) > (q \wedge \neg p)$
$Frame_0 = \{(f, a) : a \in \Delta_0, f \in \mathcal{F}\}$.

If $q \wedge \neg p$ holds in the initial state, performing action a makes p true, but this cannot block the persistency of q since $\neg q$ cannot be derived from p since the causal rule is not contrapositive. However, assuming that q persists after the action leads to $do(a) > q$, since $q > \neg p$, by (CE), we derive $do(a) > \neg p$ from which we get together with $do(a) > p$, $do(a) > \bot$. This means that the execution of action a leads to an inconsistent state, i.e. a cannot be executed in the state in which $q \wedge \neg p$ holds.

Another situation which can lead to an inconsistent state can occur when two conflicting actions are executed simultaneously. In [4], conflicting actions could occur leading to inconsistent states. In this present theory, the concurrent application of two actions a and b is only possible when these actions are independent, i.e. when Π contains $\overline{\Box}(\neg(do(a) > \neg do(b)) \wedge \neg(do(b) > \neg do(a)))$.

Example 5 Consider a swinging door and two actions $push\_in$ and $push\_out$ the first one opening the door by pushing from out-side to open it and the second by pushing it in the opposite direction. We get the following formalization:

Π: $\overline{\Box}(do(push\_in) > open\_in)$
$\overline{\Box}(do(push\_out) > open\_out)$
$\overline{\Box}(open\_in > \neg open\_out)$
$\overline{\Box}(open\_out > \neg open\_in)$
$\overline{\Box}(\neg(do(push\_in) > \neg do(push\_out)))$
$\overline{\Box}(\neg(do(push\_out) > \neg do(push\_in)))$
$Frame_0 = \{(f, a) : a \in \Delta_0, f \in \mathcal{F}\}$.

Both actions are independent. But when trying to perform them at the same moment, nothing would happen, because there is a conflict between the effects of the two actions. The door stays in the same position. All the extensions of the theory contain the formulas:

(1) $do(push\_in) > open\_in$,
(2) $do(push\_out) > open\_out$,
(3) $do(push\_in) \wedge do(push\_out) > \bot$.

Let us reconsider example 1 modified such that the two switches form a two-way wiring.

Example 6 There is a two-way wiring circuit with two two-way switches and a lamp. There are two actions of toggling each of the switches and the lamp is alight when the two switches are in the same position. Toggling any of the switches changes the light (from on to off or vice-versa). We have, for $i = 1, 2$:

$$
\begin{array}{ll}
\Pi: & \overline{\Box}(\neg sw_i \rightarrow (do(tg_i) > sw_i)) \\
 & \Box(sw_i \rightarrow (do(tg_i) > \neg sw_i)) \\
 & \Box(sw_1 \leftrightarrow sw_2 > light) \\
 & \overline{\Box}(\neg(sw_1 \leftrightarrow sw_2) > \neg light) \\
 & \overline{\Box}(\neg(do(tg_1) > \neg do(tg_2))) \\
 & \overline{\Box}(\neg(do(tg_2) > \neg do(tg_1))) \\
\textit{Obs:} & do(\epsilon) > (sw_1 \wedge \neg sw_2 \wedge \neg light) \\
\multicolumn{2}{l}{Frame_0 = \{(f, a) : a \in \Delta_0, f \in \mathcal{F}\}.}
\end{array}
$$

As before, the first four rules in Π describe the immediate effects of the action of toggling one of the switches. All fluents are regarded as being persistent and the two toggling actions are independent. The domain description $(\Pi, Frame_0, Obs)$ has one extension E relative to the action sequence $\{do(tg_1)\}, \{do(tg_2)\}$ containing the frame axioms:

$$
\begin{array}{l}
\neg sw_2 \rightarrow (do(tg_1) > \neg sw_2) \\
do(tg_1) > (\neg sw_1 \rightarrow (do(tg_2) > \neg sw_1).
\end{array}
$$

The following formulas are derivable in E:

(1) $do(tg_1) > light$,
(2) $do(tg_1) > (do(tg_2) > \neg light)$,
(3) $(do(tg_1) \wedge do(tg_2)) > \neg light$,
(4) $do(tg_1) > ((do(tg_1) \wedge do(tg_2)) > light)$,
(5) $\overline{\Box}(light \rightarrow (do(tg_1) \wedge do(tg_2) > light))$,
(6) $\overline{\Box}(\neg light \rightarrow (do(tg_1) \wedge do(tg_2) > \neg light))$.

We can see that the concurrent execution of both toggling actions never changes the status of the lamp, who remains alight (5) or not alight (6). This is true in the domain frame independently of specific observations. Moreover, (3), (5) and (6) are monotonically derivable from the domain description and they hold in all extensions.

4 Conclusion

We have presented a new logical approach to actions and causality which uses a single implication $>$ for causal consequence. Action execution and causal implication are represented uniformly. This makes it possible to integrate reasoning about mutual action dependence or independence into the language of the logic itself. This possibility distinguishes our approach from many other approaches, for example [14], who formulate dependencies outside the logic. Our action language can handle (co-operating, independent and conflicting) concurrent actions in a natural way without adding extra formal devices, and we believe that the language can be naturally extended to handle other boolean expressions concerning action performance.

References

[1] G. Brewka and J. Hertzberg, 'How to do things with worlds: on formalizing actions and plans', *Journal of Logic and Computation*, **3**(5), (1993).

[2] Nonman Foo, Dongmo Zhang, Yan Zhang, Samir Chopra, and Bao Vo, 'Encoding solutions of the frame problem in dynamic logic', in *Logic Programming and Nonmonotonic Reasoning*, eds., T. Eiter, W. Faber, and M. Truszczynski, volume 2173 of *LNAI*, pp. 240–253. Springer, (2001).

[3] Norman Foo and Dongmo Zhang, *Dealing with the ramification Problem in extended Propositional Dynamic logic*, volume 3 of *CSLI*, 2000. to appear.

[4] Laura Giordano, Alberto Martelli, and Camilla Schwind, 'Dealing with concurrent actions in modal action logic', in *Proceedings of the 13th European Conference on Artificial Intelligence*, ed., H. Prade, pp. 537–541. John Wiley & Sons, Ltd, (1998).

[5] L. Giordano, A. Martelli and C. Schwind. Ramification and Causality in a Modal Action Logic. *Journal of Logic and Computation*, vol. 10 No. 5, pp. 625-662, 2000.

[6] Laura Giordano, Alberto Martelli, and Camilla Schwind, 'Reasoning about actions in dynamic linear time temporal logic.', in *Proceedings of the 3rd International Conference on Formal and Applied Practical Reasoning, FAPR 2000*, London, UK. Journal of the IGPL, Vol. 9, No. 2, pp. 289-303, March 2001.

[7] E. Giunchiglia and V. Lifschitz, 'Dependent fluents', In IJCAI'95 [9], pp. 1964–1969.

[8] Joseph Halpern and Judea Pearl, 'Causes and Explanations: A structural model Approach, Part I', In *Proceedings of the 17th International Joint Conference on Artificial Intelligence, IJCAI'01*, B. Nebel, ed., pp. 27-34, Seattle, USA, 2001, Morgan Kaufmann.

[9] *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Montreal, Canada, 1995. Morgan Kaufmann.

[10] Frank Jackson, 'A causal theory of counterfactuals', *Australian Journal of Philosophy*, **55**(1), 3–21, (1977).

[11] David Lewis, 'Causation', *Journal of Philosophy*, **70**, 556–567, (1973).

[12] Fangzhen Lin, 'Embracing causality in specifying the indirect effects of actions', In IJCAI'95 [9], pp. 1985–1991.

[13] Fangzhen Lin, 'Embracing causality in specifying the indeterminate effects of actions', in *Proceedings of the 133th National Conference on Artificial Intelligence*, pp. 1985–1991. AAAI Press / The MIT Press, (August 1996).

[14] Andreas Herzig Marcos A. Castilho, Olivier Gasquet, 'Formalizing action and change in modal logic i: the frame problem', *Journal of Logic and Computation*, **9**(5), 701–735, (1999).

[15] N. McCain and H. Turner, 'A causal theory of ramifications and qualifications', In IJCAI'95 [9], pp. 1978–1984.

[16] Norman McCain and Hudson Turner, 'Causal theories of action and change', in *Proceedings of the 14th National Conference on Artificial Intelligence*, pp. 460–465. AAAI Press / The MIT Press, (July 1997).

[17] D. Nute, *Topics in Conditional Logic*, Reidel Publishing Company, Dordrecht, 1984.

[18] C. Schwind, 'Causality in action theories', *Electronic Articles in Computer and Information Science*, **3**(A), 27–50, (1999).

[19] Kurt Segerberg, 'Notes on conditional logic', *Studia Logica*, **48**, 157–168, (1989).

[20] J. R. Shoenfield, 'Mathematical Logic', Addison - Wesley Publishing Company, (1967).

[21] Michael Thielscher, 'Ramification and causality', *Artificial Intelligence*, **89**(1-2), 317–364, (1997).

[22] Hudson Turner, 'A logic of universal causation', *Artificial Intelligence*, **113**, 87–123, (1999).

[23] Dongmo Zhang and Norman Foo, 'Epld: A logic for causal reasoning', in *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pp. 131–136, Seattle, USA, (2001). Morgan Kaufmann.

Appendix

Proof of the completeness theorem

The completeness is shown by the construction of a canonical model. We construct a model such that for any consistent formula A (i.e. any formula A, such that $\not\vdash \neg A$), there is a world in this model satisfying A. Moreover, we show that the semantic properties of AC (S-CV, S-OR, etc.) hold in the canonical model.

Definition 5

1. *A set of formulas Γ is called* inconsistent *iff there is a finite subset of Γ, $\{F_1, \dots F_n\}$ such that $\vdash \neg F_1 \vee \neg F_2 \vee \dots \neg F_n$. Γ is called* consistent *if it is not inconsistent. If Γ contains only one formula F, we say that F is (in)consistent.*
2. *A set of formulas Γ is called* maximal inconsistent *iff it is consistent and if for any formula F not in Γ, $\Gamma \cup \{F\}$ is inconsistent.*

We presuppose a number of properties of maximal consistent formula sets the proof of which can be found in most text books of formal logic (see e.g. [20].

The canonical model CM is defined by $CM =< W, f, [[]] >$ where

1. W is the set of all maximal consistent formula sets of AC.
   We set $\|A\| = \{w : w \in W \text{ and } A \in w\}$ for any formula A.
2. $f(A, w) = \{w' : w' \in W \text{ and } w^A \subseteq w'\}$,
   where, for any $w \in W$, $w^A = \{B : A > B \in w\}$
3. for any atom $p \in ATM$, $[[p]] = \{w : p \in w\}$

$[[]]$ is extended to formulas as usual.

We first will show that for any formula A, $\|A\| = [[A]]$. This is proven by induction on the degree of the conditional formulas. First we need the following lemma concerning conditionals.

Lemma 1 *For any conditional formula we have $A > B \in w$ iff for all $w' \in f(A, w)$, $B \in w'$*

Proof: The first half follows immediately from the definition of the selection function f. For the second half, we first observe that $w^A \cup \{\neg B\}$ is an inconsistent formula set. Suppose for the contrary, that $w^A \cup \{\neg B\}$ is consistent. Then it is included in a maximal consistent formula set $w' \in W$, $w^A \cup \{\neg B\} \subseteq w'$. But then $w^A \subseteq w'$, which means that $w' \in f(A, w)$. From this follows by our precondition that $B \in w'$. This is a contradiction to $\neg B \in w'$, since w' is consistent. Since $w^A \cup \{\neg B\}$ is inconsistent, there are formulas $\{F_1, \dots F_n\} \subseteq w^A$ such that $\vdash \neg F_1 \vee \neg F_2 \vee \dots \vee B$. By the rules of propositional calculus and rule RCK, we conclude $\vdash (A > F_1) \wedge (A > F_2) \wedge \dots (A > F_n) \rightarrow (A > B)$. But $A > F_i \in w$ for $1 \leq i \leq n$, hence $A > B \in w$ by the maximality of w. Q.E.D.

We now proceed proving $\|A\| = [[A]]$ for arbitrary formula A. This is shown by induction on the degree of A.

- If A is a classical formula (degree(A) = 0), then by construction of the canonical model and the definition of the valuation function $[[]]$, we get straightforwardly that $w \in \|A\|$ iff $w \in [[A]]$, which means that $\|A\| = [[A]]$.

- Suppose, that we have $\|F\| = [[F]]$ for all formulas F which have a degree less than n. Let be $A > B$ a formula of degree n. Let be $w \in \|A > B\|$. By the definition of $\|A > B\|$, this is equivalent to $A > B \in w$. By lemma 1, this is the case iff for all $w' \in f(A, w)$, $B \in w$. By the definition of $\|B\|$, we get equivalently $\forall w' \in f(A, w)$, $w \in \|B\|$. By induction hypothesis, since $degree(B) < n$, we can replace $\|B\|$ by $[[B]]$ and we get $\forall w' \in f(A, w)$, $w \in [[B]]$. And this is the case iff $f(A, w) \subseteq [[B]]$ which means that $w \in [[A > B]]$.

It remains to show that the canonical model CM has the properties required by our logic AC, provided the corresponding axioms belong to the logic (S-CV, S-CV, ...)

- S-RCEA if $[[A]] = [[B]]$ then $f(A, w) = f(B, w)$
  If $[[A]] = [[B]]$ by the maximality of w, we get that $A \leftrightarrow B \in w$. By RCEA, it follows that $A > C \leftrightarrow B > C \in w$, from which we get $f(A, w) = f(B, w)$.

- (S-CV) if $f(A, w) \cap [[C]] \neq \emptyset$ then $f(A \wedge C, w) \subseteq f(A, w)$
  Let be $w' \in f(A \wedge C, w)$ iff $w^{A \wedge C} \subseteq w'$. By precondition, we have $f(A, w) \cap [[C]] \neq \emptyset$ which means that $\neg(A > \neg C) \in w$. This yields using axiom CV, $(A > B) \rightarrow (A \wedge C > B) \in w$. From this we conclude $\{B : A > B \in w\} \subseteq \{B : A \wedge C > B \in w\}$ which means that $w^A \subseteq w^{A \wedge C}$. Hence we get $w^A \subseteq w'$, i.e. $w' \in f(A, w)$.

- (S-CA) $f(A \vee B, w) \subseteq f(A, w) \cup f(B, w)$
  Suppose for the contrary that there is $w_1 \in W$ such that $w_1 \notin f(A, w)$ and $w_1 \notin f(B, w)$. Then there are formulas F_1 and F_2 such that $A > F_1 \in w$ and $F_1 \notin w_1$ and $B > F_2 \in w$ and $F_2 \notin w_1$ by the definition of the selection function of the canonical model. Since w_1 is maximal consistent, we have that $\neg F_1 \in w_1$ and $\neg F_2 \in w_1$. By RCK and the maximality of w_1 , we get $A > F_1 \vee F_2 \in w_1$ and $B > F_1 \vee F_2 \in w_1$. By axiom CA this yields $A \vee B > F_1 \vee F_2 \in w_1$, from which follows that $F_1 \vee F_2 \in w^{A \vee B}$. Hence we cannot have $w^{A \vee B} \subseteq w_1$ because this would contradict $\neg F_1 \in w_1$ and $\neg F_2 \in w_1$ (maximality of w_1). Therefore $w_1 \notin f(A \vee B, w)$.

- (S-CHAINING) if $f(A, w) \subseteq [[B]]$ then $f(A, w) \subseteq f(B, f(A, w))$
  where $f(B, f(A, w))$ represent the set of worlds $\{z \in f(B, x) \ : \ x \in f(A, w)\}$.
  By the precondition, we have $A > B \in w$; by axiom CE, we then get $(A > (B > C)) \rightarrow (A > C) \in w$. But this means that $\{F : B > F \in w^A\} \subseteq \{F : A > F \in w\}$, where $\{F : A > F \in w\} = w^A$. Let be $w' \in f(A, w)$, i.e. $w^A \subseteq w'$. Then we have $\{F : B > F \in w^A\} \subseteq w'$. This means that $w' \in f(B, f(A, w))$.

- (S-MOD) if $f(B, w) \cap [[A]] \neq \emptyset$ then $f(A, w) \neq \emptyset$
  If $f(A, w) = \emptyset$ then $A > \perp \in w$, from which follows that $B > \neg A \in w$ by axiom MOD. This is equivalent to $f(B, w) \subseteq [[\neg A]]$, which gives equivalently $f(B, w) \not\subseteq [[A]]$, i.e. $f(B, w) \cap [[A]] = \emptyset$.

Proof of the completeness theorem: Proof: Let be A a formula not derivable in AC. Then $\not\vdash A$, i.e. $\{\neg A\}$ is consistent. Then there is a maximal set of formulas w such that $\neg A \in w$. And this means that the canonical model CM satisfies $\neg A$, i.e. $CM, w \not\models A$. Q.E.D.