Modal Tableaux for Reasoning About Actions and Plans

Marcos A. Castilho*, Olivier Gasquet, and Andreas Herzig

Université Paul Sabatier — IRIT — Applied Logic Group
118, route de Narbonne
TOULOUSE CEDEX 4
F31062 – FRANCE

URL: *http://www.irit.fr/ACTIVITES/EQ_ALG*
e-mail: {*castilho, gasquet, herzig*}*@irit.fr*
fax: +33 5.61.55.83.25

Abstract. In this paper we investigate tableau proof procedures for reasoning about actions and plans. Our framework is a multimodal language close to that of propositional dynamic logic, wherein we solve the frame problem by introducing the notion of dependence as a weak causal connection between actions and atoms. The tableau procedure is sound and complete for an important fragment of our language, within which all standard problems of reasoning about actions can be expressed, in particular planning tasks. Moreover, our tableaux are analytic and provide thus a decision procedure.

1 Introduction

In this paper we investigate tableau proof procedures for reasoning about actions and plans in a propositional modal language. There is a family of modal operators $[\alpha]$, where α is the name of an action, and a single modal operator $\Box$ to represent integrity constraints. Its dual modal operator $\Diamond$ can be used to formulate goals in planning tasks. We start with the logic of actions and plans $\mathcal{LAP}$, which is a simple multimodal logic where the logic of each $[\alpha]$ is the basic modal logic K, while that of $\Box$ is $S4$. $\Box$ interacts with every α in the sense that $\Box A \rightarrow [\alpha]A$.

$\mathcal{LAP}$ can be thought of as a simplified propositional dynamic logic (PDL) [Har84]: We claim that $\Box$ is sufficient to simulate the PDL iteration operator $(*)$ in the standard reasoning about actions tasks. The difference between $*$ and $\Box$ is that the former semantically corresponds to reflexive and transitive closure, while the latter only *contains* reflexive and transitive closure. This makes that $\mathcal{LAP}$ is both compact and strongly complete, while PDL is not.

A tableau method for $\mathcal{LAP}$ can be defined straightforwardly by combining the tableau rules for the logics K and $S4$. Nevertheless, to effectively reason about actions and plans in $\mathcal{LAP}$ is difficult due to inertia: For every action only

* On leave from Universidade Federal do Paraná, Brazil

'very few' formulas change their truth value. For this reason, $\mathcal{LAP}$ tableau have to cope with a huge number of so-called frame axioms of the form $\Box(L \to [\alpha]L)$, L being a literal.

We could have recasted the solution of [Rei91] in our modal language, in a way similar to [DGL95]. There, the language of PDL is extended with converse (and also complement). We solve the problem in a conceptually simpler framework, by adapting the representation formalism (via the introduction of dependence relations), and in consequence the tableau reasoning mechanism. The story goes as follows:

1. For every action α and atomic formula P, whenever there are frame axioms $\Box(P \to [\alpha]P)$ and $\Box(\neg P \to [\alpha]\neg P)$ we say that *P is independent of α* (or α does not influence P). We drop such frame axioms and replace them by an *independence relation* between actions and atomic formulas. P is independent of α if and only if in any situation, the execution of α leaves the truth value of P unchanged.
2. In order to achieve representational economy, we do not write down the independence relation, but its complement, the dependence relation, which is supposed to be much smaller than the dependence relation, due to the hypothesis of inertia. In other words, we expect that for a given action α, almost all of the atoms P are independent of α. To check whether α and P are independent, we look up in the dependence relation whether $\alpha \rightsquigarrow P$ appears, and apply thus (a very simple version of) negation by failure. (This is the only nonmonotonic feature of our approach.)
3. To each dependence relation $\rightsquigarrow$ corresponds a logical system $\mathcal{LAP}_{\rightsquigarrow}$. We then adapt the tableau method for $\mathcal{LAP}$ to take into account the dependence relation: In the projection rule for α, every literal which is independent of α is preserved. This method is incomplete in the general case. We can show that it is complete for an important fragment of our language, and that after a normal forming step all standard problems of reasoning about actions and plans fall into that fragment. Completeness is proved w.r.t. a particular class of $\mathcal{LAP}$-models, viz. those models respecting the dependence relation: If P is independent of α, then the interpretation of P in worlds related by the accessibility relation associated to α is the same.

The rest of the paper is as follows: Section 2 contains language and terminology, and section 3 presents the logic of action and plans $\mathcal{LAP}$. In section 4 we show how to solve the frame problem by adding dependence relations to $\mathcal{LAP}$. Section 5 presents the tableau method, while in section 6 we show how it can be applied in planning tasks. Finally, in section 7 we discuss the characteristics of our approach.

2 Language and Terminology

In this section we present the logic of action and plans $\mathcal{LAP}$, and we state the terminology in terms of our modal language.

Let $ACT = \{\alpha, \beta, \ldots\}$ be the set of *atomic actions* (e.g. "*load*", "*wait*" and "*shoot*"), and let $ATM = \{P, Q, \ldots\}$ be the set of *atoms* (e.g. "*Loaded*"). The set $LIT = \{L, L_1, \ldots\}$ of literals is made up by all atoms and their negations. $|.|$ is a function mapping literals to atoms, such that for every $P \in ATM$, $|P| = P$ and $|\neg P| = P$. The set FOR of *formulas* is defined in the usual way, in particular $[\alpha]A \in FOR$, if $A \in FOR$ and $\alpha \in ACT$. We read $[\alpha]A$ as "A after α", and $\Box A$ as "A is always true". (Hence $[\alpha]\bot$ expresses that α is inexecutable.) A formula without modal operators is *classical*.

We also use the standard PDL abbreviations $\langle\alpha\rangle A$ for $\neg[\alpha]\neg A$, $\Diamond A$ for $\neg\Box\neg A$, $[\alpha;\beta]A$ for $[\alpha][\beta]A$, $[\alpha \cup \beta]A$ for $([\alpha]A \wedge [\beta]A)$, $[A?]B$ for $A \rightarrow B$, $[\lambda]A$ for A and $[\mathit{if}\ A\ \mathit{then}\ \alpha\ \mathit{else}\ \beta]B$ for $((A \rightarrow [\alpha]B) \wedge (\neg A \rightarrow [\beta]B))$. We read λ as "do nothing", $A?$ as "continue if A holds", $\alpha \cup \beta$ as "do either α or β nondeterministically", $\alpha;\beta$ as "do α followed by β".

Reasoning about actions and plans involves three main concepts:

1. *Knowledge bases* (KB) are finite sets of formulas without the $\Box$-operator, e.g. $\{\mathit{Walking}, [\mathit{shoot}]\mathit{Alive}\}$. They represent factual knowledge (observations).
2. *Integrity constraints* (IC) are finite sets of formulas of the form $\Box A$ representing general laws. If A is a classical formula then $\Box A$ is a *static constraint* (IC_{stat}), e.g. $\Box(\mathit{Walking} \rightarrow \mathit{Alive})$. (The ramification problem is generally viewed as the problem of taking into account such constraints.) Else $\Box A$ is a *dynamic constraint* (IC_{dyn}) which is used in particular to describe action laws. Among the latter there are *effect constraints* of the form $\Box(B \rightarrow [\alpha]C)$, where B and C are classical, e.g. $\Box(\mathit{Loaded} \rightarrow [\mathit{shoot}]\mathit{Dead})$. If $C = \bot$ they are *inexecutability constraints*, (e.g. $\Box(\neg HasGun \rightarrow [\mathit{shoot}]\bot)$), and if $B = C$ they are *frame axioms*, e.g. $\Box(\mathit{Alive} \rightarrow [\mathit{load}]\mathit{Alive})$. The frame problem is to avoid the explicit representation of frame axioms. Another form of action laws are *executability constraints* of the form $\Box(A \rightarrow \langle\alpha\rangle\top)$, where A is classical, e.g. $\Box(HasGun \rightarrow \langle \mathit{shoot}\rangle\top)$. We shall consider that IC is made up of static, effect, and executability constraints.
3. *Queries* are formulas whose derivability from the knowledge base under the integrity constraints is checked. Particular queries are *goals*, which are of the form $\Diamond A$ with A classical, e.g. $\Diamond\neg\mathit{Alive}$. These are intended to express that there is a plan leading to a state of affairs where A is true.

The different reasoning about actions tasks described in [San95] can be described as theorem proving problems in our modal framework. They typically involve a formula KB_1 describing (part of) the initial state of affairs, a sequence of actions $\pi = \alpha_1; \ldots; \alpha_n$ and a formula KB_2 describing (part of) the resulting state of affairs. *Prediction* means to prove $KB_1 \rightarrow [\pi]KB_2$, e.g. $Walking \rightarrow [\mathit{wait}; \mathit{shoot}]\mathit{Dead}$. *Postdiction* means to prove $\langle\pi\rangle KB_2 \rightarrow KB_1$, e.g. $\langle \mathit{shoot}\rangle Alive \rightarrow \neg\mathit{Loaded}$. *Planning* amounts to prove $KB_1 \rightarrow \Diamond KB_2$, e.g. $\mathit{Loaded} \rightarrow \Diamond\mathit{Dead}$. Then a plan π can be constructed from that proof.[1]

[1] Note that in the most general case, in prediction problems we do not know KB_2, and hence should not *find out whether* KB_2, but just *find* KB_2. The same is the case for KB_1 in postdiction. This requires abductive reasoning mecanisms.

3 Semantics of $\mathcal{LAP}$

Definition 1. *Models* for $\mathcal{LAP}$ ($\mathcal{LAP}$-models) are quadruples of the form $\mu = (W, (R_\alpha)_{\alpha \in ACT}, R_\Box, \tau)$, where W is a set of worlds, $R_\Box$ and each R_α are binary relations on W (accessibility relations), and $\tau : ATM \longrightarrow 2^W$ is an interpretation of atoms. We require $R_\Box$ to be reflexive and transitive, and $R_\alpha \subseteq R_\Box$ for every R_α.

The truth conditions are as usual, in particular:

- $\models^\mu_w [\alpha]A$ if for all $w' \in W : wR_\alpha w'$ implies $\models^\mu_{w'} A$;
- $\models^\mu_w \Box A$ if for all $w' \in W : wR_\Box w'$ implies $\models^\mu_{w'} A$.

Validity and satisfiability are defined as usual.

It is a standard result in multimodal logics that an axiomatics can be obtained by taking that of $S4$ for $\Box$ and that of K for every $[\alpha]$, together with an interaction axiom $\Box A \to [\alpha]A$ for every $[\alpha]$. As well, $\mathcal{LAP}$ is decidable, compact, and strongly complete [Cat89]. Semantic tableau for $\mathcal{LAP}$ can be obtained by combining the tableau rules for the logics K and $S4$, taking care of the inclusion relation between $\Box$ and $[\alpha]$.[2]

Within $\mathcal{LAP}$, the frame problem is unsolved, i.e. we must explicitly write down all frame axioms.

4 Dependence and the Semantics of $\mathcal{LAP}_\leadsto$

Now we show how frame axioms can be represented economically by means of dependence relations, and how the semantics of $\mathcal{LAP}$ can be adapted appropriately.

Definition 2. A *dependence relation* is a binary relation $\leadsto \subseteq ACT \times ATM$. The complement of $\leadsto$ is noted $\not\leadsto$ and is called the *independence relation*.

$\alpha \leadsto P$ is read "P depends on α", or "α influences P". This expresses that after the execution of action α the truth value of P *may* change: There exists a state of affairs such that either P is true and becomes false after execution of α, or P is false and becomes true after execution of α. E.g. in the Yale shooting scenario, we have $load \not\leadsto Alive$, and $load \leadsto Loaded$, $shoot \leadsto Loaded$, $shoot \leadsto Alive$ (see example 5).

P is independent of α whenever the set of frame axioms contains both $\Box(P \to [\alpha]P)$ and $\Box(\neg P \to [\alpha]\neg P)$. The construction of $\leadsto$ allows us to drop all these frame axioms from the set of effect constraints. (Note that the $\leadsto$ is in the metalanguage.)

Suppose given some dependence relation $\leadsto$. Our reading of $\leadsto$ requires to constrain $\mathcal{LAP}$-models:

[2] Although such a combination is straightforward, Fabio Massacci [Mas97] has pointed out to us that we can no longer prove in the usual way that such tableau are in PSPACE. To our knowledge, there are currently no results in the literature concerning the complexity of $\mathcal{LAP}$.

Definition 3. An $\mathcal{LAP}_{\leadsto}$-*model* is an $\mathcal{LAP}$-model (satisfying the conditions of definition 1) such that for $\alpha \in ACT$, $P \in ATM$, and for every $w, w' \in W$ such that $wR_\alpha w'$

– if $\alpha \not\leadsto P$ then $w \in \tau(P)$ iff $w' \in \tau(P)$.

This condition says that if α does not influence P then the execution of α leaves unchanged the truth value of P.

A formula A is $\mathcal{LAP}_{\leadsto}$-valid ($\models_{\mathcal{LAP}_{\leadsto}} A$) iff $\models^{\mu}_{w} A$ for every world w of every $\mathcal{LAP}_{\leadsto}$-model μ.

Example 4 (The blocksworld).

$$\begin{array}{rl} KB = & \{On_{1,2} \wedge On_{2,3} \wedge On_{3,Table}\} \\ \leadsto = & \{puton_{i,j} \leadsto On_{i,j}, puton_{i,j} \leadsto Clear_j, puton_{i,j} \leadsto Holding_i, \\ & take_{i,j} \leadsto On_{i,j}, take_{i,j} \leadsto Clear_j, take_{i,j} \leadsto Holding_i\} \\ IC = & \{\Box[take_{i,j}]\neg On_{i,j},\ \Box((Empty \wedge Clear_i \wedge On_{i,j}) \to \langle take_{i,j}\rangle \top), \\ & \Box[puton_{i,j}]On_{i,j},\ \Box((Holding_i \wedge Clear_j) \to \langle puton\rangle_{i,j} \top), \\ & \Box(Empty \leftrightarrow \bigwedge_i \neg Holding_i),\ \Box(Holding_i \leftrightarrow \bigwedge_j \neg On_{i,j}), \\ & \Box Clear_{Table},\ \Box(Clear_j \leftrightarrow \bigwedge_i \neg On_{i,j}) \text{ for } j \neq Table, \\ & \Box\neg(On_{i,k} \wedge On_{j,k}) \text{ for } i \neq j,\ \Box\neg(On_{i,j} \wedge On_{i,k}) \text{ for } j \neq k\} \end{array}$$

Then $(KB \wedge IC) \to [take_{1,2}][puton_{1,Table}][take_{2,3}]Clear_3$ is $\mathcal{LAP}_{\leadsto}$-valid. As well, $(KB \wedge IC) \to \Diamond Clear_3$ is valid, i.e. there is a plan to clear block 3.

$\mathcal{LAP}_{\leadsto}$ has the same theoretical complexity as $\mathcal{LAP}$. This can be seen by the following transformation to $\mathcal{LAP}$: Given a dependence relation $\leadsto$ and a formula A, we reconstruct the set IC_{frame} of frame axioms that are relevant for A:

$$IC_{frame} = \{\Box(L \to [\alpha]L) : |L| \text{ occurs in } A, \text{ and } \alpha \not\leadsto |L|\}$$

Clearly, A is $\mathcal{LAP}_{\leadsto}$-satisfiable iff $A \wedge IC_{frame}$ is $\mathcal{LAP}$-satisfiable, and the size of $A \wedge IC_{frame}$ is linear in that of A.

Example 5 (The Yale Shooting Scenario (YSS)). We consider the classical scenario of [HM86] adapted by [Bak91]. Let *wait*, *load* and *shoot* be the actions of waiting, loading and shooting, respectively. Let the atoms *Loaded*, *Alive* and *Walking* represent "the gun is loaded", "the man is alive", and "the man is walking" respectively. The effect of the load action is that the gun is loaded and the effects of the shoot action is that unload the gun and also of killing the person if the gun is loaded. This action has no effect if the gun is unloaded. The wait action has no effect. We have domain knowledge that dead persons do not walk. Initially, the gun is unloaded and the man is walking and alive. This is represented in $\mathcal{LAP}_{\leadsto}$ by the sets below. It is easy to check that we have $\models_{\mathcal{LAP}_{\leadsto}} (KB \wedge IC) \to [load][wait][shoot](\neg Loaded \wedge \neg Alive \wedge \neg Walking)$.:

$$\leadsto = \left\{ \begin{array}{l} load \leadsto Loaded, \\ shoot \leadsto Loaded, \\ shoot \leadsto Alive, \\ shoot \leadsto Walking \end{array} \right\}$$

$$IC = \left\{ \begin{array}{l} \Box(Walking \to Alive), \\ \Box\langle wait\rangle\top, \\ \Box\langle load\rangle\top, \\ \Box[load]Loaded, \\ \Box\langle shoot\rangle\top, \\ \Box[shoot]\neg Loaded, \\ \Box(Loaded \to [shoot]\neg Alive), \\ \Box(\neg Loaded \to ((Alive \to [shoot]Alive) \wedge (\neg Alive \to [shoot]\neg Alive))) \\ \Box(\neg Loaded \to ((Walking \to [shoot]Walking) \wedge (\neg Walking \to [shoot]\neg Walking))) \end{array} \right\}$$

$$KB = \{\neg Loaded, Alive, Walking\}$$

Note that after *wait* the independence $wait \not\leadsto Loaded$ is the warrant that the gun remains loaded after wait. Note also that the choice of a weak causal connection between actions and literals may force us to write down some additional (conditional) frame axioms.

Example 6 (Nondeterministic actions). We can illustrate by a modification of the blocksworld example that we can deal with nondeterministic actions: Suppose e.g. that we change the above definition of the action $puton_{i,j}$ (because the robot now is careless and may drop the block) to be $\Box[puton_{i,j}](On_{i,j} \vee On_{i,Table})$. We can still prove $\models_{\mathcal{LAP}_{\leadsto}} (KB \wedge IC) \to \Diamond Clear_3$, but no longer $\models_{\mathcal{LAP}_{\leadsto}} (KB \wedge IC) \to \Diamond On_{3,2}$.

Other nondeterministic actions can be formalized as well, such as tossing a coin, or throwing a coin on a chessboard [GKL95].

5 Automated Theorem Proving for $\mathcal{LAP}_{\leadsto}$

Suppose given some dependence relation $\leadsto$. In this section we show how to adapt $\mathcal{LAP}$ tableau to take into account $\leadsto$. We suppose the reader is familiar with tableau terminology. The presentation is in the style of [Gor92].

The tableaux calculus for $\mathcal{LAP}_{\leadsto}$ is defined by the set of tableau rules showed in figure 1, where:

- $\Box X = \{\Box A : A \in X\}$,
- $[\alpha]X = \{[\alpha]A : A \in X\}$,
- $\not\leadsto_\alpha X = \{L : L \in X \cap LIT \text{ and } \alpha \not\leadsto |L|\}$.

In fact, the rule $(K_{\not\leadsto}[\alpha])$ is the only non-standard tableau rule. It extends the standard one of modal logic K by the preservation of the set of independent

$\frac{X;A;\neg A}{\perp}(\perp)$	$\frac{X;\neg\neg A}{X;A}(\neg)$	$\frac{X;A\wedge B}{X;A;B}(\wedge)$	$\frac{X;\neg(A\wedge B)}{X;\neg A \mid X;\neg B}(\vee)$
$\frac{X;\Box A}{X;\Box A;A}(T\Box)$	$\frac{Y;\Box X;\neg\Box A}{\Box X;\neg A}(4\Box)$	$\frac{V;\Box X;[\alpha]Y;\not\leadsto_{\alpha} Z;\neg[\alpha]A}{\Box X;Y;\not\leadsto_{\alpha} Z;\neg A}(K_{\not\leadsto}[\alpha])$	

Fig. 1. Tableau rules for $\mathcal{LAP}_{\leadsto}$

literals. If this set was not preserved, then e.g. under $\alpha{\leadsto}P$ and $\alpha{\not\leadsto}Q$ no tableau for the $\mathcal{LAP}_{\leadsto}$-unsatisfiable formula $\Box(P \vee Q) \wedge \neg[\alpha]P \wedge \neg Q$ would close.[3]

Theorem 7 (Soundness). *If there is a closed $\mathcal{LAP}_{\leadsto}$-tableau for $\neg A$ then $\models_{\mathcal{LAP}_{\leadsto}} A$.*

The proof is in [CGH96].

Unfortunately this tableau calculus is incomplete: Suppose $\alpha{\not\leadsto}P$ and $\beta{\not\leadsto}P$. Then the formula

$$[\alpha]P \vee [\beta]\neg P \tag{1}$$

is $\mathcal{LAP}_{\leadsto}$-valid, but there is no closed $\mathcal{LAP}_{\leadsto}$-tableau for it. But completeness holds for a particular fragment:

Definition 8. A formula A is in *normal form* iff no compound dynamic operator appears in A, and for every subformula $[\alpha]B$ of A, B is of the form $B_1 \vee B_2 \vee \ldots \vee B_n$, and for $1 \leq i \leq n$ one of the following holds:

- B_i is of the form $[\beta]C_i$
- B_i is of the form $\neg[\beta]C_i$
- B_i is a literal and $\alpha{\leadsto}|B_i|$

E.g. suppose $\alpha_1{\leadsto}P_1$, $\alpha_1{\not\leadsto}P_2$, $\alpha_2{\not\leadsto}P_1$ and $\alpha_2{\leadsto}P_2$. Then $\Box[\alpha_1](P_1 \vee [\alpha_2]P_2)$ is in normal form, whereas $[\alpha_1]P_2$, $[\alpha_1]\Box P_1$, and $[\alpha_2][\alpha_1]P_2$ are not.

Formulas having $\Box$-operators in the scope of $[\alpha]$-operators, e.g. $[\alpha]\Box P$, cannot be put into normal form. Such kind of formulas would be used to express "from now on" laws, i.e. laws which are valid after the execution of an action. e.g. $\mathit{Loaded} \to [\mathit{shoot}]\Box \mathit{Dead}$ ("shoot causes the person to be definitively dead") or executability order of actions, e.g. $\Box[\alpha]\Box[\beta]\perp$ ("action α can not be executed before action β).

Luckily enough, the fragment of the language for which this can be done is sufficiently big to do our three reasoning tasks of section 2, named, prediction,

[3] The naive implementation of this rule may lead to an inefficient prover, because a (theoretical) huge number of atoms must be preserved each time the rule is applied. This is due to our hypothesis of inertia. This may be solved in practical cases by the implementation of a labelled tableau together with some strategy that allow us to just look up for independent literals when they are needed.

postdiction and planning. We were unable to find in the literature of reasoning about actions a domain that is not expressible in this sub-language.

We call a $\mathcal{LAP}_{\rightsquigarrow}$-formula *ordered* if no $\Box$-operator appears in the scope of any $[\alpha]$-operator.

Theorem 9. *For every ordered formula there is an equivalent ordered formula in normal form.*

The proof is in [CGH96].

E.g. let $\alpha \rightsquigarrow P$, $\alpha \rightsquigarrow Q$ and $\alpha \not\rightsquigarrow R$. Then the formula $[\alpha](P \wedge [\alpha](R \vee \neg[\alpha]Q))$ takes the normal form $[\alpha]P \wedge (R \vee [\alpha][\alpha]\neg[\alpha]Q)$. As another example, if $\alpha \not\rightsquigarrow P$ and $\beta \not\rightsquigarrow P$, the formula 1 takes the normal form $P \vee \neg P$. It is easy to see that there is a tableau proof of the latter.

Theorem 10 (Completeness). *Let A be an ordered formula in normal form. If $\models_{\mathcal{LAP}_{\rightsquigarrow}} A$ then there is a closed $\mathcal{LAP}_{\rightsquigarrow}$-tableau for $\neg A$.*

The proof is in [CGH96]. It uses that our tableaux are analytic, and hence the tableau can be constructed in a way such that termination is ensured. Therefore our tableau method can be used as a decision procedure to check validity of ordered formulas A.[4]

Example 11 (YSS, cont.). A proof of $\models_{\mathcal{LAP}_{\rightsquigarrow}} KB \wedge IC \rightarrow [l][s](\neg L \wedge \neg A)$ (consider l to be *load*, s to be *shoot*, L to be *Loaded* and A to be *Alive*) is:

$$\dfrac{\neg(KB \wedge IC \rightarrow [l][s](\neg L \wedge \neg A))}{KB; IC; \neg[l][s](\neg L \wedge \neg A)}{\scriptstyle (\wedge), several times}$$

$$\dfrac{KB; IC; \neg[l][s](\neg L \wedge \neg A)}{KB; IC; \neg[l][s](\neg L \wedge \neg A); [l]L}{\scriptstyle (T\Box)}$$

$$\dfrac{KB; IC; \neg[l][s](\neg L \wedge \neg A); [l]L}{A; IC; \neg[s](\neg L \wedge \neg A); L}{\scriptstyle (K_{\not\rightsquigarrow}[l])}$$

$$\dfrac{A; IC; \neg[s](\neg L \wedge \neg A); L}{A; IC; \neg[s](\neg L \wedge \neg A); L; L \rightarrow [s]\neg A}{\scriptstyle (T\Box)}$$

$$\dfrac{A; IC; \neg[s](\neg L \wedge \neg A); L; L \rightarrow [s]\neg A}{\dfrac{A; IC; \neg[s](\neg L \wedge \neg A); L; \neg L}{\bot}{\scriptstyle (\bot)} \qquad \dfrac{\dfrac{A; IC; \neg[s](\neg L \wedge \neg A); L; [s]\neg A}{A; IC; \neg[s](\neg L \wedge \neg A); L; [s]\neg A; [s]\neg L}{\scriptstyle (T\Box)}}{IC; \neg(\neg L \wedge \neg A); \neg A; \neg L}{\scriptstyle (K_{\not\rightsquigarrow}[s])}}{\scriptstyle (\vee)}$$

$$\dfrac{IC; \neg(\neg L \wedge \neg A); \neg A; \neg L}{\dfrac{IC; \neg\neg L; \neg A; \neg L}{\bot}{\scriptstyle (\bot)} \qquad \dfrac{IC; \neg\neg A; \neg A; \neg L}{\bot}{\scriptstyle (\bot)}}{\scriptstyle (\vee)}$$

Note how A ($Alive \in KB$) was preserved when rule $(K_{\not\rightsquigarrow}[l])$ was applied between lines 3 and 4 due to $load \not\rightsquigarrow Alive$.

[4] The naive way of putting the formula in conjunctive normal form may cause exponential growth. In classical logic, this can be avoided by a more sophisticated algorithm where new atoms are introduced [Tse83,Sie87]. The resulting formula is satisfiability-equivalent to the original formula. Recently, this method has been extended to modal logic in [Mat93].

6 Plan Generation

Given a closed tableau for $KB \wedge IC \wedge \Box\neg G$, can we obtain a plan from that proof, i.e. a sequence of actions π such that $\models_{\mathcal{LAP}_{\leadsto}} (KB \wedge IC) \rightarrow \langle\pi\rangle G$? This will be the case if no negative occurrences of the $\Box$-operator occur in KB, IC and G.[5] Such a fragment is still large enough for our reasoning tasks of section 2: There, both KB and IC are conjunctions of formulas of the form $\Box A$, A containing no $\Box$-operator, and G is classical in the goal $\Diamond G$.

Here is the *plan generation algorithm* extracting a complex action π from the closed tableau for $IC \wedge KB \wedge \Box\neg G$:

1. Mark all occurrences of subformulas of G produced by the tableau rules (including G).
2. Mark all nodes where $(\bot)$ has been applied to a marked formula (i.e. where a subformula of $\Box\neg G$ is 'responsible' for closure).
3. Mark all ancestor nodes of marked nodes.
4. Associate a plan to the root by bottom-up iteration.[6] Let n be some marked node in the tableau.
   - If the rule $(\bot)$ has been applied to n, then the plan of n is λ.
   - If the rule $(\wedge)$ has been applied to n, then the plan of n is that of the successor node.
   - If the rule $(\vee)$ has been applied to n, and only one successor node is marked, then the plan of n is the plan of that successor node.
   - If the rule $(\vee)$ has been applied to n, decomposing the formula $\neg(A_1 \wedge A_2)$, and both successor nodes are marked and have plans π_1 and π_2, then the plan of n is: "*if* A_1 *then* π_1 *else* π_2".
   - If the rule $(T\Box)$ has been applied to n, then the plan of n is that of the successor node.
   - If the rule $(K_{\not\leadsto}[\alpha])$ has been applied to n, and the plan of the successor node is π_1, then the plan of n is: "$\alpha; \pi_1$".

Theorem 12. *Let $KB \wedge IC \wedge \Box\neg G$ be $\mathcal{LAP}_{\leadsto}$-unsatisfiable, and let π be a complex action that has been associated to the root of some closed $\mathcal{LAP}_{\leadsto}$-tableau for it by the plan generation algorithm. If neither KB nor IC nor $\neg G$ contain negative occurrences of $\Box$ then $(KB \wedge IC) \rightarrow \langle\pi\rangle G$ is $\mathcal{LAP}_{\leadsto}$-valid.*

The proof is done by replacing $\Box\neg G$ by $[\pi]\neg G$ in the tableau. Then the same sequence of tableau rules as in the original tableau can be applied, and we can prove by induction that the latter leads again to a closed tableau. E.g. take the case

[5] This can be explained by the fact that in the tableau, if there is some negative occurrence of the $\Box$-operator, then one can "jump" to another world without identifying any specific action. If the transitions are only triggered by $\neg[\alpha]$, then a plan can be found by the composition of all actions that have been applied from the initial world to the last one.

[6] We suppose here that the root is marked. Else $IC \wedge KB$ would be unsatisfiable alone, and any plan would do.

of the rule ($\vee$) decomposing the node $X; \neg(A_1 \vee A_2)$ into $X; \neg A_1$ and $X; \neg A_2$. Suppose both $X; \neg A_1$ and $X; \neg A_2$ are marked and have plans π_1 and π_2. First, note that we must have $X = X' \cup \{\Box\neg G\}$ (because each of our tableau rules preserves boxed formulas). Then by induction hypothesis both $X'; [\pi_1]\neg G; \neg A_1$ and $X'; [\pi_2]\neg G; \neg A_2$ are closed. Now $[$ *if* A_1 *then* π_1 *else* $\pi_2]\neg G \wedge \neg(A_1 \wedge A_2)$ abbreviates $(A_1 \to [\pi_1]\neg G) \wedge (\neg A_1 \to [\pi_2]\neg G) \wedge \neg(A_1 \wedge A_2)$, and the latter classically implies $([\pi_1]\neg G \wedge \neg A_1) \vee ([\pi_2]\neg G \wedge \neg A_2)$. Hence the node $X'; \Box\neg G; \neg(A_1 \vee A_2)$ must be closed. Finally, note that in unmarked nodes, $\Box\neg G$ is not 'responsible' for the contradiction, and hence $\Box$ can be instantiated without harm.

Example 13 (The blocksworld, cont.). Consider again example 4. As $\mathcal{LAP}_{\leadsto}$-tableaux are complete, a closed tableau for $KB \wedge IC \wedge \Box\neg(Clear_3 \wedge On_{3,Table})$ exists and the above algorithm will extracts a plan from it. The smallest one (coming from a smallest tableau) is $\pi = take_{1,2}; puton_{1,Table}; take_{2,3}; puton_{3,Table}$.

Generally there might be more than one closed tableau for a given formula $\neg A$, and in consequence an extracted plans might be unnecessarily complex. Clearly, the shorter the tableau proof, the simpler the extracted plan.

Note that the ($\vee$)-rule introduces conditional plans, and it might be impossible to check such conditions. Consider e.g. $IC = \{\langle\alpha_1\rangle\top, \langle\alpha_2\rangle\top\}$, and $KB = \{[\alpha_1]P \vee [\alpha_2]P\}$. There is a tableau proof that $(IC \wedge KB) \to \Diamond P$, which gives us the plan *if* $[\alpha_1]P$ *then* α_1 *else* α_2. But such a plan might not allow us to effectively act in order to obtain P. Take e.g. a game where there are two doors behind one of which there is a treasure, and read α_1 as "open the first door", α_2 as "open the second door", and P as "you get the treasure".

Note also that when a conditional plan is associated to a ($\vee$)-tableau rule, we can choose between A_1 and A_2 as a condition. This is of interest because it might be the case that A_2 can be tested, while A_1 cannot (or only in a more expensive way). In particular, the above example illustrates that it should be generally preferable to test classical formulas (instead of modal ones).

7 Discussion

We have presented a modal framework for reasoning about actions and plans, for which we have given a decision procedure. Although this framework is less expressive than propositional dynamic logic, it is expressive enough to deal with the standard reasoning about actions tasks of prediction, postdiction, and planning. The differences concerning the theoretical complexity remain to be established.

The main features of our approach can be summarized as follows: It is in terms of a family of *modal logics*, and allows to express *nondeterminism* and *indirect effects* of actions. It is based on the use of *dependence relations* and provides a *monotonic* solution to the frame problem. The decision procedure is based on a simple *tableau theorem prover*. Neither of these characteristics is novel alone, but, to the best of our knowledge, it is the first time that one approach unites *all* of such features:

(1) Modal logics and monotonic solutions to the frame problem: Modal logics (viz. PDL and extensions of it) have been used before to model reasoning about actions domains, e.g. in [Ros81,SB93,DGL95]. In [SB93] frame axioms are computed in a way similar to our construction of dependence relations, and in [DGL95] there is another monotonic solution to the frame problem in the style of [Rei91] and others, using converse and complement of actions.[7]

(2) The concept of dependence: The idea of adding information about dependence between actions and atoms is similar to the use of causality-like notions found in the literature on reasoning about actions. In this sense, dependence is a sort of weak causality. We just mention Sandewall's notion *occluded* [San95] and Kartha and Lifschitz' *releases* [KL94]. Nevertheless, contrarily to the approaches of Lifschitz *et col.*, it makes no sense in $\mathcal{LAP}_{\leadsto}$ to state a dynamic constraint $[\alpha]P$ without stating that P depends on α, while this might be the case e.g. in $\mathcal{AR}_0$. Dependence is also closely related with the next topic.
(3) Nondeterminism and indirect effects of actions: While nondeterministic actions can be handled in $\mathcal{LAP}_{\leadsto}$ in a straightforward way, actions with indirect effects (which are not covered by Reiter's solution) must be analysed in more depth. Although static integrity constraints such as $\Box(\mathit{Walking} \rightarrow \mathit{Alive})$ can be easily formulated, we must take care of the 'indirect dependences' of actions: The dependences of α must mention *all* the atoms that may be changed by the execution of α. E.g. if $\Box(P \leftrightarrow Q)$ and $\alpha \leadsto P$ then we must also have $\alpha \leadsto Q$. For the time being this must be done by hand. The automatic generation of such derived dependences is subject of ongoing research. This motivates the formal study of the notion of dependence, and its relation to that of an integrity constraint. First results can be found in [FdCH96,Her97]. Anyway, there is a general feeling that that it is necessary to dispose of some causlity-like relation to solve the ramification problem [Lin95,MT95,Thi95,Thi97] and weak causality notion such as our goes into the same direction.
(4) Proof methods: $\mathcal{LAP}_{\leadsto}$ -tableaux need a preliminary step of normal form, as we have seen in section 5. But even if the naive algorithm is employed, in practical cases the size of the resulting formula should be close to that of the original one. One of the reasons is that action laws are already in normal form. (It would be strange to write down action laws $A \rightarrow [\alpha]C$ with C containing atoms that do not depend on α.) Furthermore, knowledge bases describing only the actual situation are in normal form. Hence it is only the goal which must be put in conjunctive normal form. Note that in planning problems, even this step is superfluous because the goal is already in normal form. Hence the complexity of

[7] This is opposed to minimization-based approaches, which are used in many formal systems dealing with the frame problem. Generally speaking, minimization is a particular reasoning mechanism which is added on top of a logic. Often things are formulated in the situation calculus, which means that we are in the framework of second-order logic. In our case there is no such minimization – or at least, it is a very simple one, which corresponds to the fact that we represent the dependence relation, while what is used in the procedure is the independence relation. Nevertheless, we have shown by a sound and complete translation [CGH96] that our formalism has at least the same expressive power as the minimization-based action logic $\mathcal{A}$ [GL93].

reasoning in our system is that of our tableau method. Unfortunately, as we have already pointed out, it seems that the theorectical complexity of our system is over PSPACE, which is the complexity of propositional STRIPS [Byl94].[8] Future work must be done in order to implement and evaluate the tableau method in practice. We have particular interest in the work of [GS96], which takes profit of existing SAT-algorithms for classical logic, and in that of [HSZ96], which presents a very efficient loop test for modal logic $S4$.

Acknowledgements We are grateful to Fabio Massacci, who has pointed out to us that our proof of PSPACE completeness in a previous version of this paper was erroneous. Thanks to Luis Fariñas del Cerro and two anonymous referees, whose comments helped to improve the presentation of the paper. This work was partially supported by CAPES (Brazil) under grants to the first author.

References

[Bak91] A.B. Baker. Nonmonotonic reasoning in the framework of situation calculus. *Artificial Intelligence (AI)*, 49(1–3):5–23, may 1991.

[Byl94] T. Bylander. The computational complexity of propositional STRIPS planning. *Artificial Intelligence (AI)*, 69(1–2):165–204, 1994.

[Cat89] L. Catach. *Les logiques multi-modales.* PhD thesis, Université Paris VI, France, 1989.

[CGH96] M.A. Castilho, O. Gasquet, and A. Herzig. Modal tableaux for reasoning about actions and plans. IRIT internal report, nov 1996.

[DGL95] G. De Giacomo and M. Lenzerini. PDL-based framework for reasoning about actions. In *Proc. 4th Congresss of the Italian Association for Artificial Intelligence (IA*AI'95)*, number 992 in LNAI, pages 103–114. Springer-Verlag, 1995.

[FdCH96] L. Fariñas del Cerro and A. Herzig. Belief change and dependence. In Yoav Shoham, editor, *Proc. 6th Conf. on Theoretical Aspects of Rationality and Knowledge (TARK'96)*, pages 147–162. Morgan Kaufmann Publishers, 1996.

[GKL95] E. Giunchiglia, G. N. Kartha, and V. Lifschitz. Actions with indirect effects (extended abstract). In *Working notes of the AAAI-Spring Sysposium on Extending Theories of Actions*, 1995.

[GL93] M. Gelfond and V. Lifschitz. Representing action and change by logic programs. *Journal of Logic Programming*, pages 301–321, 1993.

[Gor92] R.P. Goré. *Cut-free sequent and tableau systems for propositional normal modal logics.* PhD thesis, University of Cambridge, England, 1992.

[GS96] F. Giunchiglia and R Sebastiani. A SAT-based decision procedure for $\mathcal{ALC}$. In *Proc. Int. Conf. on Knowledge Representation and Reasoning (KR'96)*, pages 302–314, Cambridge, Massachussetts, 1996.

[Har84] D. Harel. Dynamic logic. In D. Gabbay and F. Günthner, editors, *Handbook of Philosophical Logic*, volume II, pages 497–604. D. Reidel, Dordrecht, 1984.

[Her97] A. Herzig. How to change factual beliefs using laws and independence information. In Dov M. Gabbay and Rudolf Kruse, editors, *Proc. Int. Joint Conf. on Qualitative and Quantitative Practical Reasoning (ECSQARU/FAPR-97)*, LNCS. Springer-Verlag, jun 1997.

[8] Unlike STRIPS, in our approach we can deal with nondeterminism and ramifications.

[HM86] S. Hanks and D. McDermott. Default reasoning, nonmonotonic logics, and the frame problem. In *Proc. Nat. (US) Conf. on Artificial Intelligence (AAAI'86)*, pages 328–333, Philadelphia, PA, 1986.

[HSZ96] A. Heuerding, M. Seyfried, and H. Zimmermann. Efficient loop-check for backward proof search in some non-classical propositional logics. In P. Miglioli, U. Moscato, D. Mundici, and M. Ornaghi, editors, *Proceedings of the 5th International Workshop TABLEAUX'96: Theorem Proving with Analytic Tableaux and Related Methods*, number 1071 in LNAI, pages 210–225. Springer-Verlag, 1996.

[KL94] G. N. Kartha and V. Lifschitz. Actions with indirect effects (preliminary report). In *Proc. Int. Conf. on Knowledge Representation and Reasoning (KR'94)*, pages 341–350, 1994.

[Lin95] F. Lin. Embracing causality in specifying the indirect effects of actions. In *Proc. of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*, pages 1985–1991, Montreal, Canada, 1995.

[Mas97] F. Massacci. Personal communication, may 1997.

[Mat93] C. Mathieu. A resolution method for a non-monotonic multimodal logic. In S. Moral, editor, *Proc. ECSQUARU'93*, LNCS, pages 257–264. Springer-Verlag, 1993.

[MT95] N. McCain and H. Turner. A causal theory of ramifications and qualifications. In *Proc. of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*, pages 1978–1984, 1995.

[Rei91] R. Reiter. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. *Artificial Intelligence and Mathematical Theory of Computation*, Papers in Honor of John McCarthy:359–380, 1991.

[Ros81] S. Rosenschein. Plan synthesis: a logical approach. In *Proc. of the 8th International Joint Conference on Artificial Intelligence (IJCAI'81)*, pages 359–380, Academic Press, 1981.

[San95] E. Sandewall. *Features and Fluents.* Oxford University Press, 1995.

[SB93] W. Stephan and S. Biundo. A new logical framework for deductive planning. In *Proc. of the 13th International Joint Conference on Artificial Intelligence (IJCAI'93)*, pages 32–38, 1993.

[Sie87] P. Siegel. *Représentation et utilisation de la connaissance en calcul propositionnel.* PhD thesis, Université d'Aix-Marseille II, Aix-Marseille, France, jul 1987.

[Thi95] M. Thielscher. The logic of dynamic systems. In *Proc. of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*, pages 1956–1962, Montreal, Canada, 1995.

[Thi97] M. Thielscher. Ramification and causality. *Artificial Intelligence (AI)*, 89:317–364, 1997.

[Tse83] G.S. Tseitin. On the complexity of derivations in propositional calculus. In Siekmann Wrightson, editor, *Automated Reasoning 2: Classical papers on computational logic*, pages 466–483, 1983.