Reasoning about actions in modal logic

Marcos A. Castilho[1] **Luis Fariñas del Cerro**
Olivier Gasquet **Andreas Herzig**

Université Paul Sabatier — IRIT — Applied Logic Group
118, route de Narbonne
31062 – TOULOUSE CEDEX – FRANCE
e-mail: {castilho, farinas, gasquet, herzig}@irit.fr

Abstract

We investigate reasoning about actions and plans in a modal framework. We solve the frame problem by replacing the frame axioms by a dependence relation between actions and literals. We show how a tableau method for the resulting logic can be defined from the basic modal tableaux for modal logics K and $S4$ in order to take into account the dependence relation. This leads us to a sound, complete and terminating tableau calculus with two new rules called SF (Step Forward) and SB (Step Back). Intuitively, these rules move literals not dependent of a given action from worlds to worlds. Contrarily to most of the frameworks for reasoning about actions in the literature, our approach does not necessitate semantical manipulations or second-order quantifiers as in literature of effective automated procedures.

1 Introduction

In this paper we investigate reasoning about actions and plans in a propositional modal lagic. 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. We start with the modal 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 $S4$. The dual modal operator of possibility $\Diamond$ can be used to formulate goals in plan generation tasks. $\Box$ interacts with every $[\alpha]$ in the sense that $\Box A \rightarrow [\alpha]A$.

To effectively reason about actions and plans in $\mathcal{LAP}$ is difficult due to inertia: for every action only '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 $\Box(L \rightarrow [\alpha]L)$, L being a literal.

[1] On leave from Universidade Federal do Paraná/Brazil.

We could have recasted the solution of [13] in our modal language, in a way similar to [3] (where the language of PDL is extended with converse and complement), but 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.

We first define an independence relation $\not\rightsquigarrow$ between actions and literals: $\alpha \not\rightsquigarrow L$ is read "action α may not cause L". It codes the set of frame axioms, in the sense that $\alpha \not\rightsquigarrow L$ if and only if the execution of α may never make L true. In other words, if L has truth value false, it remains false after execution of α. This means that we represent a frame axiom $\Box(\neg L \rightarrow [\alpha]\neg L)$ by $\alpha \not\rightsquigarrow L$. But we know from the principle of inertia that, for a given action, the number of literals that it does not affect is big. On the other hand, the number of literals affected by that action is small. Hence we shall not represent the independence relation itself, but its complement, the dependence relation $\rightsquigarrow$ between actions and literals (which is supposed to be small). 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: we add rules to propagate back and forward literals independent of actions.

The rest of the paper is as follows: Section 2 contains the basic logic $\mathcal{LAP}$. In section 3 we show how to solve the frame problem by adding dependence relations to $\mathcal{LAP}$. Section 4 presents the tableau method for $\mathcal{LAP}_{\rightsquigarrow}$. Finally, in section 5 we discuss the characteristics of our approach.

2 The logic of action and plans $\mathcal{LAP}$

We present the logic of action and plans $\mathcal{LAP}$, whose language is very similar to that of propositional dynamic logic PDL [10], the difference being that our $\Box$-operator is less expressive than the iteration operator * of PDL [10]. Our framework is simpler, but it is sufficient to treat most of the problems that have been investigated up to now in reasoning about actions.

2.1 Language and notation

Let $ACT = \{\alpha, \beta, \ldots\}$ be the set of *actions* (like "*load*" and "*shoot*"), and let $ATM = \{P, Q, \ldots\}$ the set of *atoms* (like "*Loaded*" and "*Alive*"). The set LIT of *literals* is made by all the atoms and their negations. 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 use $\vee, \rightarrow, \leftrightarrow$ as usual, $\langle\alpha\rangle A$ to abbreviate $\neg[\alpha]\neg A$ and $\Diamond A$ to abbreviate $\neg\Box\neg A$. We also use $[\alpha;\beta]A$ 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 [if A then α else β]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". We read $[\alpha]A$ as "after α A is true", $\Box A$ as "A is always true". $\langle\alpha\rangle\top$ is read as "α is executable" and $[\alpha]\bot$ as "α is not executable". A formula without modal operators is *classical*.

Formulas of the form $\Box A$ are integrity constraints (IC). They are general laws about the world and fall in two categories, depending on A. Static constraints (A is a classical formula) are intended to state that A is always true, independently of any action occurrence. E.g., $\Box(\mathit{Walking} \rightarrow \mathit{Alive})$. Dynamic constraints (there is at least one modal operator $[\alpha]$ in A) are intended to describe actions laws. In particular, *effect constraints* are of the form $\Box(B \rightarrow [\alpha]C)$. E.g., $\Box(\mathit{Loaded} \rightarrow [\mathit{shoot}]\mathit{Dead})$. Those of the form $\Box(A \rightarrow [\alpha]A)$ assert that α may not cause $\neg A$. These are the *frame axioms*. E.g., $\Box(\mathit{Alive} \rightarrow [\mathit{load}]\mathit{Alive})$. Goals are formulas of the form $\Diamond A$, where A is classical, they express that there is a plan leading to a state of affairs where A is true. E.g., $\Diamond \mathit{Dead}$, $\Diamond \mathit{Alive}$.

2.2 Semantics

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

The truth conditions are as usual, in particular:

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

Validity (noted $\models_{\mathcal{LAP}} A$) and satisfiability are as usual. To design tableau method for $\mathcal{LAP}$ is straightforward. It is just to combine the tableau rules for the logics K and $S4$. We omit a presentation here, because it will be a particular case of that presented in section 4. For details, see [2].

Example 1 (The Yale Shooting Scenario (YSS), part I) In $\mathcal{LAP}$ the classical scenario of [9] is written as:

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

$$IC = \left\{ \begin{array}{ll} \Box\langle wait\rangle\top, & \Box[shoot]\neg Loaded, \\ \Box\langle load\rangle\top, & \Box(Loaded \rightarrow [shoot]\neg Alive), \\ \Box\langle shoot\rangle\top, & \Box(\neg Loaded \rightarrow (Alive \rightarrow [shoot]Alive)), \\ \Box[load]Loaded, & \Box(\neg Loaded \rightarrow [wait]\neg Loaded), \\ \Box(Loaded \rightarrow [wait]Loaded), & \Box(\neg Alive \rightarrow [wait]\neg Alive), \\ \Box(Alive \rightarrow [wait]Alive), & \Box(\neg Alive \rightarrow [load]\neg Alive), \\ \Box(Alive \rightarrow [load]Alive), & \Box(\neg Loaded \rightarrow [load]\neg Loaded), \\ \Box(\neg Alive \rightarrow [shoot]\neg Alive), & \Box(\neg Loaded \rightarrow [shoot]\neg Loaded) \end{array} \right\}$$

We have $\models_{\mathcal{LAP}} (KB \wedge IC) \rightarrow [load][wait][shoot](\neg Loaded \wedge \neg Alive)$.

2.3 $\mathcal{LAP}$ and dynamic logic

Our language is similar to that of propositional dynamic logic. For the sake of comparison, suppose that the set of atomic actions is finite, and that $ACT = \{\alpha_1, \ldots, \alpha_n\}$. $\Box A$ is intended to simulate $[(\alpha_1 \cup \ldots \cup \alpha_n)^*]A$. But we do not have an equivalence: $\Box A$ only implies $[(\alpha_1 \cup \ldots \cup \alpha_n)^*]A$, but not the converse. The main goal of the PDL *-operator is to model loop sentences of computer programs. In what concerns reasoning about actions, it is certainly useful if we want to construct complex plans involving iterations (see e.g. [3]), but changing it by our $\Box$ has some advantages:

- $\mathcal{LAP}$ is compact and strongly complete, while PDL is not.
- A tableau method can be defined straightforwardly by combining the tableau rules for the logics K and $S4$. We omit a presentation here, because it is a particular case of that presented in section 4.

Unfortunately, $\mathcal{LAP}$ and PDL still have the same complexity, as stated in the following theorem:

Theorem 1 Complexity of $\mathcal{LAP}$-satisfiability is EXPTIME-complete.

3 The logic $\mathcal{LAP}_{\leadsto}$

In this section we present the logic $\mathcal{LAP}_{\leadsto}$, which is the logic $\mathcal{LAP}$ augmented with a dependence relation in order to solve the frame problem.

3.1 Semantics

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

When we write $\alpha \rightsquigarrow L$ we want to say "α may cause L", and this expresses that after the execution of action α the truth value of L *may change*: there exists a state of affairs such that L is false and becomes true after execution of α [2]. As we shall sometimes write $\alpha \rightsquigarrow \neg L$, when L is a negative literal of the form $\neg P$, we shall understand then $\alpha \rightsquigarrow P$.

Suppose given a set of dynamic constraints IC_{dyn}, then define $\alpha \not\rightsquigarrow \neg L$ iff $\Box(L \rightarrow [\alpha]L) \in IC_{dyn}$.[3] Then we drop all these frame axioms from IC_{dyn}. The models are as follows.

Definition 3 An $\mathcal{LAP}_{\rightsquigarrow}$*-model* μ is an $\mathcal{LAP}$-model (satisfying the conditions of definition 1) which moreover satisfies, for $\alpha \in ACT$, $L \in LIT$ and every w, w' such that $wR_\alpha w'$:

- $\alpha \not\rightsquigarrow P$ and $w \notin \tau(P)$ implies $w' \notin \tau(P)$.
- $\alpha \not\rightsquigarrow \neg P$ and $w \in \tau(P)$ implies $w' \in \tau(P)$.

This condition says that if α may not cause L then the execution of α cannot make L true. (Note that the dependence relation is in the metalanguage.)

Example 2 (The Yale Shooting Scenario (YSS), part II) We rewrite in $\mathcal{LAP}_{\rightsquigarrow}$example 1:

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

$$IC = \left\{ \begin{array}{ll} \Box\langle wait\rangle\top, & \Box[shoot]\neg Loaded, \\ \Box\langle load\rangle\top, & \Box(Loaded \rightarrow [shoot]\neg Alive), \\ \Box\langle shoot\rangle\top, & \Box(\neg Loaded \rightarrow (Alive \rightarrow [shoot]Alive)), \\ \Box[load]Loaded & \end{array} \right\}$$

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

We have $\models_{\mathcal{LAP}_{\rightsquigarrow}} (KB \wedge IC) \rightarrow [load][wait][shoot](\neg Loaded \wedge \neg Alive)$.

4 A tableau calculus for $\mathcal{LAP}_{\rightsquigarrow}$

In this section we adapt the a tableau tableau for $\mathcal{LAP}$ to take into account the dependence relation. We introduce two new rules, called stepback and stepforward with this aim. Our tableau is based on labelled formulas.

[2] Note this does not imply that $\neg L$ may change, too, unless $\alpha \rightsquigarrow \neg L$.

[3] Of course, as $\rightsquigarrow$ is the complement of $\not\rightsquigarrow$, $\alpha \rightsquigarrow L$ represent the law $\Diamond(\neg L \wedge \langle\alpha\rangle L)$, keeping the intuitive meaning of $\rightsquigarrow$ explained in the previous paragraph. The $\Diamond$ is the warrant that such state of affairs exists.

Definition 4 A *labelled formula* is a pair (n, A), where A is a formula and n is an integer called the *label*[4] of A.

Definition 5 A *skeleton*[5] Σ is a 3-ary relation $\Sigma \subseteq (ACT \cup \Box) \times I\!N \times I\!N$. Elements (α, n, n') of the relation are noted $n \stackrel{\alpha}{\to} n'$.

Definition 6 A *tree*[6] is a pair $\langle S, \Sigma \rangle$, where S is a set of labelled formulas and Σ is a skeleton.

Definition 7 A *tableau for A* is the limit[7] of a sequence $C^0, \ldots, C^n, \ldots$ of forests (i.e. sets of trees) where the initial $C^0 = \{\langle \{0, A\}, \emptyset \rangle\}$, and every C^{i+1} is obtained from C^i by applying one of the tableau rules of figure 1 to some tree $\langle S, \Sigma \rangle \in C^i$.

Hence, a tableau is a particular forest which is denoted C^∞. For each tree $\langle S, \Sigma \rangle$ in that forest, we have that all the sets are downward closed in the usual sense [8].

Definition 8 Trees are closed if they contain $(i, \bot)$ for some i. A forest is closed if all of its trees are closed. (Hence, a particular tableau is closed if all of its trees are closed.)

Remark 1 The last two rules SF and SB are associated with the dependence relation: literals whose negation is independent of an action α remain unchanged when α is executed. Let us recall that in a $\mathcal{LAP}_{\leadsto}$-model if $(w, w') \in R_\alpha$ and α may not cause $\neg L$ (i.e. if $\alpha \not\leadsto \neg L$) then L is true at w implies that L must be true at w', too (this is rule SF); and conversely, if $\alpha \not\leadsto L$ then L is true at w' implies that L must be true at w, too (this is rule SB). The other rules are just the well knows rules for basic modal logics K and $S4$.

Remark 2 The rule application order is irrelevant in the sense that if at some stage some rule is applicable and is not applied, it remains applicable.

Theorem 2 (completeness) If there is a fair open tableau for A then A is satisfiable.

[4] A label denotes a possible world. (Of course, any infinite set of labels would do.)

[5] A skeleton intuitively corresponds to the accessibility relations.

[6] A tree intuitively corresponds to a model.

[7] This limit exists since as we will see, all the rules either increase the set of labelled formulas and/or increase the successor relations (of course, it may be infinite). It is obtained in the obvious way by pointwise union.

rule $\bot$**:** If S contains (n, A) and $(n, \neg A)$, then add $(n, \bot)$ to it.

rule $\neg$**:** If S contains $(n, \neg\neg A)$, then add (n, A) to it.

rule $\wedge$**:** If S contains $(n, A \wedge B)$, then add (n, A) and (n, B) to it.

rule $\vee$**:** If S contains $(n, \neg(A \wedge B))$, then add $(n, \neg A)$ to it, and add $\langle S \cup \{(n, \neg B)\}, \Sigma\rangle$ to C^{i+1}.

rule K_α**:** If S contains $(n, [\alpha]A)$, and Σ contains $n \stackrel{\alpha}{\rightarrow} n'$, then add (n', A) to it.

rule $T_\Box$**:** If S contains $(n, \Box A)$, then add (n, A) to it.

rule $4_\Box$**:** If S contains $(n, \Box A)$, and Σ contains $n \stackrel{\Box}{\rightarrow} n'$ or $n \stackrel{\alpha}{\rightarrow} n'$ for some $\alpha \in ACT$, then add $(n', \Box A)$ to it.

rule $\langle\alpha\rangle$**:** If S contains $(n, \neg[\alpha]A)$, then add $(n', \neg A)$ to it, and add $n \stackrel{\alpha}{\rightarrow} n'$ to Σ, where n' is a fresh integer.

rule $\Diamond$**:** If S contains $(n, \neg\Box A)$, then add $(n', \neg A)$ to it and add $n \stackrel{\Box}{\rightarrow} n'$ to Σ, where n' is a fresh integer.

rule SF**:** If S contains (n, L), where L is a literal, and $n \stackrel{\alpha}{\rightarrow} n' \in \Sigma$, and $\alpha \not\leadsto \neg L$, then add (n', L) to it.

rule SB**:** If S contains (n', L), $n \stackrel{\alpha}{\rightarrow} n' \in \Sigma$, and $\alpha \not\leadsto L$, then add (n, L) to it.

Figure 1: Tableau rules

Theorem 3 (soundness) If all tableaux for A are closed then A is unsatisfiable.

Remark 3 (termination and decidability) The formulas generated by the tableau rules are subformulas of the initial formula. Hence there is a finite number of distinct states. Using a loop test as for S4 [5] we can associate to our initial tableau a finite tableau from which a finite model can be extracted.

5 Conclusion

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. We have shown by a sound and complete translation [1] that our formalism has at least the same expressive power as logic $\mathcal{A}$ [6].

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: there is a general feeling in the community of reasoning about actions that it is necessary to dispose of some causality-like relation, and weak causality notions such as our dependence have been used to handle indirect effects, e.g. in [12, 7, 16]. These approaches also treat indirect effects and nondeterminism. As well, modal logics (viz. PDL and extensions of it) have been used before, e.g. in [14, 15, 3]. In [15] frame axioms are computed in a way similar to our construction of dependence relations, and in [3] there is another monotonic solution to the frame problem in the style of [13] and others, using converse and complement of actions. But none of these approaches unites *all* of our above features.

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 $\Box(Dead \leftrightarrow \neg Alive)$ can be formulated easily, 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 [4, 11].

References

[1] M.A. Castilho, O. Gasquet, and A. Herzig. Modal tableaux for reasoning about actions and plans. In S. Stell and R. Alami, editors, *European Conference on Planning (ECP'97)*, number 1348 in Lecture Notes in Artificial Intelligence, pages 104–116, Toulouse, France, sep 1997. Springer-Verlag.

[2] M.A. Castilho and A. Herzig. An alternative to the iteration operator of propositional dynamic logic. Technical Report 96-05-R, IRIT/UPS, jan 1996. http://www.irit.fr/ACTIVITES/EQ_ALG/-Papers/CaHe96a.ps.Z.

[3] 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.

[4] 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.

[5] M. Fitting. *Proof methods for modal and intuitionistic logics.* D. Reidel, Dordrecht, 1983.

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

[7] 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.

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

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] M. Thielscher. Computing ramifications by postprocessing. In *Proc. of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*, pages 1994–2000, Montreal, Canada, 1995.