A customized tableau method for reasoning about actions and plans in modal logic

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

Université Paul Sabatier — IRIT — Applied Logic Group
http://www.irit.fr/ACTIVITES/EQ_ALG/home.html
118, route de Narbonne
31062 – TOULOUSE CEDEX – FRANCE
e-mail: {castilho, gasquet, herzig}@irit.fr
fax: +33 5.61.55.83.25

Abstract

We investigate modal tableau proof procedures for reasoning about actions and plans. 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 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 speak about plans. 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 $S4$. $\Box$ 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. $\Box$ is used to represent integrity constraints, and the dual modal operator of possibility $\Diamond$ can be used to formulate goals in plan generation tasks. The difference between $*$ and $\Box$ is that the

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

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 $S4$. $\mathcal{LAP}$ and a tableau method for it were defined in [CH96].

Nevertheless, 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.

We could have recasted the solution of [Rei91] in our modal language, in a way similar to [DGL95] (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 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 check whether α and L are independent, we look up in the dependence relation whether $\alpha \rightsquigarrow L$ appears. We apply thus a very simple version of negation by failure: non-membership in a set. (This is the only nonmonotonic feature of our approach.) 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, not only all formulas of the form $\Box A$ and $[\alpha]A$ are preserved, but also every literal which is independent of α.

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 [Har84], the difference being that our $\Box$-operator is less expressive than the iteration operator * of PDL [Har84]. 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 $ACT = \{P, Q, \ldots\}$ 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 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., $\Box(Walking \rightarrow Alive)$. 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., $\Box(Loaded \rightarrow [shoot]Dead)$. Those of the form $\Box(A \rightarrow [\alpha]A)$ assert that α may not cause $\neg A$. These are the *frame axioms*. E.g., $\Box(Alive \rightarrow [load]Alive)$. 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., $\Diamond Dead$, $\Diamond Alive$.

2.2 Semantics

Definition 1 $\mathcal{LAP}$*-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 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 [CH96].

Example 1 (The Yale Shooting Scenario (YSS), part I) [HM86]
Consider a scenario where w, l and s represent the actions of waiting, loading and shooting, respectively. Let the atoms Ld and Al represent "the gun is loaded" and "the man is alive", respectively. The effect of the load action is that the gun is loaded and the effects of the shoot action are that the gun is unloaded and that the person is killed if the gun is loaded. This action has no effect if the gun is unloaded. The wait action has no effect. Initially, the gun is

unloaded and the man is alive. In $\mathcal{LAP}$ this scenario is written as:

$$\begin{array}{ll} IC = & \{\Box\langle w\rangle\top, \Box\langle l\rangle\top, \Box\langle s\rangle\top, \\ & \Box(Ld \rightarrow [w]Ld), \Box(\neg Ld \rightarrow [w]\neg Ld), \Box(Al \rightarrow [w]Al), \Box(\neg Al \rightarrow [w]\neg Al), \\ & \Box[l]Ld, \Box(\neg Al \rightarrow [l]\neg Al), \Box(Al \rightarrow [l]Al), \\ & \Box[s]\neg Ld, \Box(Ld \rightarrow [s]\neg Al), \Box(\neg Ld \rightarrow (Al \rightarrow [s]Al))\} \\ KB = & \{\neg Ld, Al\}. \end{array}$$

We have $\models_{\mathcal{LAP}} (KB \wedge IC) \rightarrow [l][w][s](\neg Ld \wedge \neg Al)$.

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

In this section we present the logic $\mathcal{LAP}_{\rightsquigarrow}$, 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 $\rightsquigarrow \subseteq ACT \times LIT$. The complement of $\rightsquigarrow$ is noted $\not\rightsquigarrow$ 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]. We shall sometimes write $\alpha \rightsquigarrow \neg L$. If 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 $\Box(L \rightarrow [\alpha]L) \in IC_{dyn}$. 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 $\alpha \in ACT$ and $L \in LIT$:

- $\alpha \not\rightsquigarrow P$ 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.

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

$$\begin{array}{ll} \rightsquigarrow = & \{l \rightsquigarrow Ld, s \rightsquigarrow \neg Ld, s \rightsquigarrow \neg Al\}; \\ IC = & \{\Box\langle w\rangle\top, \Box\langle l\rangle\top, \Box\langle s\rangle\top, \\ & \Box[l]Ld, \\ & \Box[s]\neg Ld, \Box(Ld \rightarrow [s]\neg Al), \Box(\neg Ld \rightarrow (Al \rightarrow [s]Al))\} \\ KB = & \{\neg Ld, Al\}. \end{array}$$

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

We have $\models_{\mathcal{LAP}_{\rightsquigarrow}}(KB \wedge IC) \rightarrow [l][w][s](\neg Ld \wedge \neg Al)$.

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

In this section we present a tableau proof procedure for $\mathcal{LAP}_{\rightsquigarrow}$. The first tentative to adapt the tableau for $\mathcal{LAP}$ to take into account the dependence relation is to add to the $\mathcal{LAP}$-tableau rules a new rule that takes into account the semantics of the dependence relation, *i.e.* whenever L is in some node of the tableau, if $\alpha \not\rightsquigarrow L$, this rule allows to put the literal L in all worlds accessible from that node via α. We call such a rule "stepforward"-rule. But this is not enough: suppose, for example, that $\alpha \not\rightsquigarrow P$ and $\beta \not\rightsquigarrow \neg P$, for some atom P and actions α, β. Then the formula $\langle\alpha\rangle P \wedge \langle\beta\rangle \neg P$ is unsatisfiable, but the above rule is not enough to make the tableau close. Our solution is to add a special "stepback"-rule, which permits to propagate independent literals backwards. Our tableau is based on labelled formulas, the label denoting a possible world. Of course, any infinite set of labels would do.

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

Definition 5 A *branch* is a set of arbitrary labelled-formulas. Let Br be a branch, $Br(n)$ denotes the set of formulas A such that $(n, A) \in Br$: $Br(n)$ will be called a *state*.

Definition 6 A *skeleton*[3] 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 $n \stackrel{\alpha}{\rightarrow} n'$.

Definition 7 A *form*[4] is a pair $\langle Br, \Sigma\rangle$ where Br is a branch, and Σ a skeleton.

Definition 8 A *bundle*[5] is a set of forms.

Definition 9 A *tableau for* A is the limit[6] of a sequence $C_0, \ldots, C_n, \ldots$ of bundles where $C_0 = \{\langle\{0, A\}, \Sigma^0\rangle\}$, where $\Sigma^0_{\Box}$ and each Σ^0_{α} is empty, and every C_{i+1} is obtained from C_i by applying one of the tableau rules below to some form of C_i. Hence, a tableau is a bundle.

Definition 10 States, branches and forms are closed if they contain $(i, \bot)$ for some i. A bundle is closed if all its forms are closed. A tableau is closed if all its forms are closed. A formula A is closed if one its tableaux is closed.

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

[4] A form intuitively corresponds to a model.

[5] A bundle intuitively corresponds to distinct forms from disjunctive choices.

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

Tableau rules

rule $\bot$: If Br^i contains (n, A) and $(n, \neg A)$ then add $(n, \bot)$ to it.

rule $\neg$: If Br^i contains $(n, \neg\neg A)$ then add (n, A) to it.

rule $\wedge$: If Br^i contains $(n, A \wedge B)$ then add (n, A) and (n, B) to it.

rule $\vee$: If Br^i contains $(n, \neg(A \wedge B))$ then add $(n, \neg A)$ to it, and add $(Br^i \cup \{(n, \neg B)\}, \Sigma^i)$ do C_i.

rule K_α: If Br^i contains $(n, [\alpha]A)$ then add (n', A) to it if Σ^i_α contains (n, n').

rule $T_\Box$: If Br^i contains $(n, \Box A)$ then add (n, A) to it.

rule $4_\Box$: If Br^i contains $(n, \Box A)$ then add $(n', \Box A)$ to it if $(n, n') \in \Sigma^i_\Box$ if for some $\alpha \in ACT$ $(n, n') \in \Sigma^i_\alpha$.

rule $\langle\alpha\rangle$: If Br^i contains $(n, \langle\alpha\rangle A)$ then add (n', A) to it, and add (n, n') to Σ^i_α where n' is a fresh integer.

rule $\Diamond$: If Br^i contains $(n, \Diamond A)$ then add (n', A) to it and add (n, n') to $\Sigma^i_\Box$ where n' is a fresh integer.

rule SF: If Br^i contains (n, L) where L is a literal, and $(n, n') \in \Sigma^i_\alpha$, and $\alpha \not\rightsquigarrow \neg L$ then add (n', L) to it.

rule SB: If Br^i contains (n', L) where L is a literal, and $(n, n') \in \Sigma^i_\alpha$, and $\alpha \not\rightsquigarrow L$ then add (n, L) to it.

Figure 1: Tableau rules

Let $C_i = \{\ldots, \langle Br^i, \Sigma^i\rangle, \ldots\}$ with $\Sigma^i = \{\Sigma^i_\alpha : \alpha \in ACT\} \cup \{\Sigma^i_\Box\}$: the rules below show how to construct C_{i+1} from C_i by applying one rule of figure 1 to one labelled formula of Br^i.

Remark 1 The last two rules SF and SB are associated with the dependence relation: literals independent of an action α remain unchanged when α is executed. Let us recall that if $(n, n') \in R_\alpha$ and α may not cause $\neg L$ (i.e. if $\alpha \not\rightsquigarrow \neg L$) then L is true at n implies that L must be true at n', too (this is rule SF); and conversely, if $\alpha \not\rightsquigarrow L$ then if L is true at n' then L must be true at n, too (this is rule SB).

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 at a further

stage. This would lead to a distinct (but equivalent) tableau, as long as the fairness assumption is satisfied.

Theorem 1 (completeness) Under the fairness assumption, if there is a fair open tableau for A then A is satisfiable.

Theorem 2 (correctness) 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 [Fit83] we can associate to our initial tableau a finite tableau from which a finite model can be extracted.

Example 3 (The YSS, part III) Continuing example 2, we give in figure 2 a tableau proof of $\models_{\mathcal{LAP}_{\rightsquigarrow}}(KB \wedge IC) \rightarrow [l][w][s](\neg L \wedge \neg A)$. We remark that the information common to all the forms of the tableau is factorized for the sake of clarity (in particular, the common part of the skeletons) We denote "add (n, n') to Σ_α" by $n \stackrel{\alpha}{\rightarrow} n'$. Note how Ld is transported when rule SF is applied in line 8 due to $w \not\rightsquigarrow \neg Ld$.

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. 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: 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 [KL94, GKL95, Thi95]. These approaches also treat indirect effects and nondeterminism.[7] As well, modal

[7] Note that contrarily to the minimization-based approaches of Lifschitz and colleagues, it makes no sense in $\mathcal{LAP}_{\rightsquigarrow}$ to state a dynamic constraint $[\alpha]P$ without stating that P depends on α (whereas this might be the case e.g. in $\mathcal{AR}_0$).

label:		States			:	Skeletons	: Explanation
1	:	$(0, \neg((KB \wedge IC) \rightarrow [l][w][s](\neg Ld \wedge \neg Al)))$			:	$\Sigma_\alpha = \Sigma_\Box = \emptyset$:	
2a	:	$(0, \neg Ld)$			:		: $(\wedge)$ on 1
2b	:	$(0, Al)$			:		: $(\wedge)$ on 1
2c	:	$(0, \Box[l]Ld)$			:		: $(\wedge)$ on 1
2d	:	$(0, \Box[s]\neg Ld)$			:		: $(\wedge)$ on 1
2e	:	$(0, \Box(Ld \rightarrow [s]\neg Al))$			:		: $(\wedge)$ on 1
2f	:	$(0, \Box(\neg Ld \rightarrow (Al \rightarrow [s]Al)))$			:		: $(\wedge)$ on 1
2g	:	$(0, \neg[l][w][s](\neg Ld \wedge \neg Al))$			:		: $(\wedge)$ on 1
3	:	$(0, [l]Ld)$			:		: $(T_\Box)$ on 2c
4	:	$(1, \neg[w][s](\neg Ld \wedge \neg Al))$			:	$0 \xrightarrow{l} 1$	: $(\langle l \rangle)$ on 2g
5	:	$(2, \neg[s](\neg Ld \wedge \neg Al))$			:	$1 \xrightarrow{w} 2$	: $(\langle w \rangle)$ on 4
6	:	$(3, \neg(\neg Ld \wedge \neg Al))$			:	$2 \xrightarrow{s} 3$	: $(\langle s \rangle)$ on 5
7	:	$(1, Ld)$			:		: (K_l) on 3
8	:	$(2, Ld)$			:		: (SF) on 7
9	:	$(2, \Box[s]\neg Ld)$			:		: $(4_\Box)$ on 2d
10	:	$(2, \Box(Ld \rightarrow [s]\neg Al))$			:		: $(4_\Box)$ on 2e
11	:	$(2, [s]\neg Ld)$			:		: $(T_\Box)$ on 9
12	:	$(2, Ld \rightarrow [s]\neg Al)$			:		: $(T_\Box)$ on 10
13	:	$(2, \neg Ld)$	$(2, [s]\neg Al)$				: $(\vee)$ on 12
14	:	$(2, \bot)$	$(3, \neg\neg Ld)$	$(3, \neg\neg Al)$			: $(\bot)$ on 8 and 13; $(\vee)$ on 6
15	:	closed	$(3, \neg Ld)$	$(3, \neg Al)$	:		: (K_s) on 11 ; (K_s) on 13
16	:		$(3, \bot)$	$(3, \bot)$	:		: $(\bot)$ on 14 and 15 ; idem
17	:		closed	closed	:		: all tableaux are closed

Figure 2: A $\mathcal{LAP}_{\rightsquigarrow}$-tableau proof of $\models_{\mathcal{LAP}_{\rightsquigarrow}} (KB \wedge IC) \rightarrow [l][w][s](\neg L \wedge \neg A)$.

logics (viz. PDL and extensions of it) have been used before, 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.[8] But none of these approaches unites *all* of our above features.

While nondeterministic actions can be handled in $\mathcal{LAP}_{\rightsquigarrow}$ 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 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 $\alpha \rightsquigarrow P$ then we must also have $\alpha \rightsquigarrow Q$. 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].

References

[CGH97] M.A. Castilho, O. Gasquet, and A. Herzig. Modal tableaux for reasoning about actions and plans. submitted to a conference, april 1997.

[CH96] M.A. Castilho and A. Herzig. An alternative to the iteration operator of propositional dynamic logic. Technical Report 96-05-R, Institut de recherche en informatique de Toulouse (IRIT), Université Paul Sabatier, 1996. http://www.irit.fr/ACTIVITES/EQ_ALG/Papers/-CH96a.ps.Z.

[DGL95] G. De Giacomo and M. Lenzerini. PDL-based framework for reasoning about actions. In *Proceedings of the Fourth 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 Ra-*

[8] 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 [CGH97] that our formalism has at least the same expressive power as the minimization-based action logic $\mathcal{A}$ [GL93].

tionality and Knowledge (TARK'96), pages 147–162. Morgan Kaufmann Publishers, 1996.

[Fit83] M. Fitting. *Proof methods for modal and intuitionistic logics.* D. Reidel, Dordrecht, 1983.

[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.

[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.

[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.

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

[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.

[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.

[Thi95] 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.