Theoretical Computer Science 129 (1994) 167–186
Elsevier

A tableau-like proof procedure for normal modal logics

Zoran Ognjanović
Mathematical Institute, Kneza Mihaila 35, 11000 Belgrade, Yugoslavia

Communicated by M. Nivat
Received February 1992
Revised November 1992

Abstract

Ognjanović, Z., A tableau-like proof procedure for normal modal logics, Theoretical Computer Science 129 (1994) 167–186.

In this paper a new proof procedure for some propositional and first-order normal modal logics is given. It combines a tableau-like approach and a resolution-like inference. Completeness and decidability for some propositional logics are proved. An extension for the first-order case is presented.

1. Introduction

The tableau approach was emphasized as a useful basis for theorem proving [18, 34], especially in the field of nonclassical logics [3, 10, 12, 15]. Some automated theorem provers for modal logics based on the tableau approach have already been implemented [6, 12, 37] and some distinct extensions of resolution [32] for modal logics proposed [1, 2, 5, 7–9, 13, 14, 19, 24, 29]. To complete the survey, we mention that there are still several different approaches in modal theorem proving: natural deduction [33, 36], the translation of modal formulas into classical logic [4, 30], the connection method [35], etc.

In this article we consider a new proof procedure for propositional and first-order S4, some other normal modal logics and intuitionistic logic. The procedure is based on the dual tableau for classical logic [18] (including a procedure dual to the propositional resolution) and on Kripke models [21, 22]. The standard propositional

Correspondence to: Z. Ognjanović, Mathematical Institute, Kneza Mihaila 35, 11000 Belgrade, Yugoslavia. Email: eognjano@yubgef51.bitnet.

SSDI 0304-3975(93)E0012-S

modal language and closed first-order language without functional symbols are discussed.

Our proof procedure could be understood as follows. Supposing that a formula is valid, we analyze its behavior in an arbitrary world of some arbitrary model. We generate a tree called dual tableau and some sets of subformulas of the studied formula. The subformulas are associated with some worlds of the model. By properly chosen reduction rules (reflecting semantic laws of valuation) we always need at least one of those sets to be satisfied to acknowledge that the examined formula is really valid. In an attempt to establish that at least one of the sets of subformulas is satisfied, a procedure dual to the classical propositional resolution is used. If we do not succeed in proving the formula we are in a position to construct its counter model.

This paper is organized as follows. We begin with propositional modal logics and emphasize the case of S4. In Section 2 the basic definitions are given. Section 3 contains the rules of our system and the completeness theorem for S4. In Section 4 some other logics and appropriate changes in the rules are considered. In Section 5 we show that the procedure described here is a decision procedure. Extensions to the first-order case are considered in Section 6. Conclusions are summarized in Section 7.

2. Preliminaries

Suppose that the propositional modal language for S4 consists of logical operators ($\neg$, $\wedge$, $\vee$, $\rightarrow$, $\Diamond$ and $\Box$), propositional variables and auxiliary symbols ("(" and ")"). The atomic formulas and formulas are defined as usual. We assume that T and F are new formal symbols [10, 12, 34], and if X is a formula of propositional modal language then T X and F X are signed formulas.

Signed nonatomic formulas are grouped in α, β, ν, π and negative and positive formulas. Figure 1 defines the formulas and their respective components. Intuitively, an α-formula is true iff its components are also true. The same holds for negative and corresponding positive formulas. A β-formula is true iff either the corresponding β_1- or β_2-formula is true. To understand ν- and π-formulas we require the notion of Kripke models [21].

Definition 2.1. Let W be a nonempty set of elements called worlds and let P be a set of signed propositional modal formulas. The triple $\langle W, R, \Vdash \rangle$ is called a Kripke propositional model if

(1) R is a relation over $W \times W$ called the visibility (or accessibility) relation and

(2) $\Vdash$ is a relation over $W \times P$ called valuation, and the following conditions are met for every $w \in W$:

(a) for every unsigned modal formula X, either $w \Vdash \mathrm{T}\,X$ or $w \Vdash \mathrm{F}\,X$,

(b) for every α-formula, $w \Vdash \alpha$ iff $w \Vdash \alpha_1$ and $w \Vdash \alpha_2$,

(c) for every β-formula, $w \Vdash \beta$ iff $w \Vdash \beta_1$ or $w \Vdash \beta_2$,

(d) for every ν-formula, $w \Vdash \nu$ iff $(\forall u \in W)(w\,R\,u \Rightarrow u \Vdash \nu_0)$,

α	α_1	α_2
$\mathrm{T}X \wedge Y$	$\mathrm{T}X$	$\mathrm{T}Y$
$\mathrm{F}X \vee Y$	$\mathrm{F}X$	$\mathrm{F}Y$
$\mathrm{F}X \rightarrow Y$	$\mathrm{T}X$	$\mathrm{F}Y$

β	β_1	β_2
$\mathrm{F}X \wedge Y$	$\mathrm{F}X$	$\mathrm{F}Y$
$\mathrm{T}X \vee Y$	$\mathrm{T}X$	$\mathrm{T}Y$
$\mathrm{T}X \rightarrow Y$	$\mathrm{F}X$	$\mathrm{T}Y$

ν	ν_0
$\mathrm{T}\Box X$	$\mathrm{T}X$
$\mathrm{F}\Diamond X$	$\mathrm{F}X$

π	π_0
$\mathrm{F}\Box X$	$\mathrm{F}X$
$\mathrm{T}\Diamond X$	$\mathrm{T}X$

Negative	Positive
$\mathrm{T}\neg X$	$\mathrm{F}X$
$\mathrm{F}\neg X$	$\mathrm{T}X$

Fig. 1.

(e) for every π-formula, $w \Vdash \pi$ iff $(\exists u \in W)(w R u \wedge u \Vdash \pi_0)$, and

(f) if A is a negative and B the corresponding positive formula, $w \Vdash A$ iff $w \Vdash B$. The pair $\langle W, R \rangle$ is called a frame.

Now, a ν-formula is true in world $w \in W$ iff ν_0 is true in every world visible from w. A π-formula is true in world $w \in W$ iff there is at least one world visible from w in which π_0 is true.

Definition 2.2. The modal signed formula X is valid in model $\langle W, R, \Vdash \rangle$ if $w \Vdash X$ for any $w \in W$. Formula X is valid in a collection of models if it is valid in every corresponding model. Formula X is valid in a frame $\langle W, R \rangle$ if it is valid in every model $\langle W, R, \Vdash \rangle$, for arbitrary valuation $\Vdash$.

For an unsigned formula Y, we say $w \Vdash Y$ if $w \Vdash \mathrm{T}\,Y$, and $w \Vdash \neg Y$ if $w \Vdash \mathrm{F}\,Y$. Since an unsigned modal formula Y behaves like $\mathrm{T}\,Y$, while $\neg Y$ behaves like $\mathrm{F}\,Y$, we say that Y is valid in a collection of models if the same holds for $\mathrm{T}\,Y$.

The modal logic S4 is characterized by models with a reflexive and transitive visibility relation. A formula X is S4-valid iff for any world w from any Kripke model $\langle W, R, \Vdash \rangle$ with transitive and reflexive visibility relation, $w \Vdash X$.

A prefix is an integer. If X is a signed formula and k is a prefix, then kX is a prefixed signed formula. For our purpose, the connection with the Kripke models is realized using prefixes as the names of worlds. This connection will be made clearer later on.

3. Dual tableau for S4 (propositional case)

A dual tableau is a tree whose nodes are labeled by prefixed signed subformulas of the examined formula. The tableau construction is followed by the construction of a frame which will present a paradigm of the class of frames and corresponding modal models in which the validity of the formula is investigated. Prefixes are the worlds of that frame.

The construction rules are:

(1) A formula $0\,\mathrm{T}\,A$ is placed in the tableau's root, where 0 is the prefix. The relation ρ marks the visibility between prefixes, and at the beginning contains only element $(0,0)$. After introducing a new prefix relation ρ will be updated.

(2) Depending on the type of formula in the node, one of the following rules should be applied.

(a) If the node contains an α-formula with prefix k, the branch where the node is located is extended with nodes containing subformulas α_1 and α_2, with the same prefix.

(b) If the node contains a β-formula with prefix k, the branch where the node is located branches with nodes containing subformulas β_1 and β_2, with the same prefix.

(c) If the node contains a negative formula with prefix k, the branch where the node is located is extended with the node containing the corresponding positive formula, with the same prefix.

(d) If the node contains a ν-formula with prefix k, and if the same rule has not been applied to the same formula and prefix k, the branch where the node is located is extended with the node containing formula ν_0, with the new prefix k'; the pair (k,k') is added to the relation ρ, and its reflexive and transitive closure is made. If the same pair (formula ν and prefix k) has already introduced a node containing prefix k'' and formula ν_0 at some other place, the branch containing the considered node is extended with the node containing the prefixed formula $k''\nu_0$.

(e) If the node contains a π-formula with prefix k, let a prefix k' be visible from k, and suppose that this rule has not been used at that node and prefix k'. If this rule has not been used at all at the examined π-node, the branch where the node is located is extended with a node containing the π_0-formula and prefix k'. Let this new node be the first π_0-descendant of the examined node. If this rule has been used at the examined π-node, the branch containing the node branches, and the new extension is a node with the corresponding π_0-formula and prefix k'. The branching is done in the predecessor of the first π_0-descendant of the examined π-node, so that every π_0-node (corresponding to the examined π-node) belongs to a different branch.

Figure 2 illustrates visually the idea behind the mentioned rules. Rules 2(a)–2(c) are, in fact, classical [10, 18, 34].

Starting with the root containing a formula with prefix 0 and sign T, and using the above rules, a sequence of trees is constructed. Each tree in the sequence extends the previous one. A node is reduced by the rules at most once on a particular branch. After

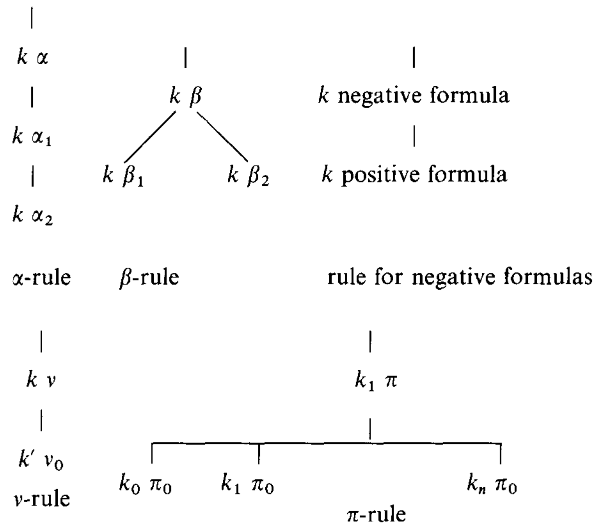


Fig. 2.

that the node is finished. π-nodes are an exception. A π-node with prefix k is finished if there cannot be any new prefix k'' visible from k and the π-rule has been applied to every prefix k' visible from k. Nodes containing signed atomic formulas are also finished, for no rules can reduce them. A branch is finished if it cannot be extended by the reduction rules.

The *dual tableau* is the first tree from the sequence containing only finished nodes.

Definition 3.1. An S4-interpretation I is a mapping from a set P of prefixes (from a tableau $\mathcal{T}$) into a set W of worlds of some S4-model $M = \langle W, R, \Vdash \rangle$, i.e. $I : P \to W$, when the following is satisfied:

$$(\forall k \in P)(\forall k' \in P)(k \, \rho \, k' \Rightarrow I(k) \, R \, I(k')).$$

Definition 3.2. The signed formula X with prefix k is satisfied under an S4-interpretation I which maps a set of prefixes into a set of worlds of some model $\langle W, R, \Vdash \rangle$ if $I(k) \Vdash X$. A set of signed formulas is satisfied under an S4-interpretation I if each formula from this set is satisfied under the interpretation I.

Lemma 3.3. *Let $\mathcal{T}$ be a tableau whose root contains formula $0\,\mathrm{T}\,X$. Formula X is S4-valid iff for each S4-model $\langle W, R, \Vdash \rangle$ and for each S4-interpretation I which maps the tableau's prefixes into the worlds from W there exists at least one branch of the tableau whose set of all prefixed atomic signed formulas is satisfied under the interpretation I.*

Proof. ($\Leftarrow$) Suppose that formula X is not S4-valid. Then, there is a model $\langle W, R, \Vdash \rangle$ and a world $w \in W$ such that $w \Vdash \mathrm{T} X$ does not hold. Consider the interpretation I which satisfies the following statements:

(a) $I(0) = w$ and

(b) if prefix k' is introduced using the ν-rule from prefix k and $I(k) = w_1$ then

$$I(k') = \begin{cases} \text{any} \quad w_2 \colon w_1 \, R \, w_2, \text{ if } I(k) = w_1 \Vdash \nu, \\ \text{some } w_2 \colon w_1 \, R \, w_2, \text{ such that not } w_2 \Vdash \nu_0, \text{ otherwise.} \end{cases}$$

There is at least one branch whose set of all prefixed atomic signed formulas is satisfied under interpretation I. Using induction we shall show that the set B of all prefixed signed formulas from the branch is satisfied. If $k\alpha \in B$ (k is a prefix), then by the construction rules both $k\alpha_1$ and $k\alpha_2$ belong to B, and by the induction hypothesis they are satisfied under interpretation I, i.e. $I(k) \Vdash \alpha_1$ and $I(k) \Vdash \alpha_2$. By the definition of Kripke models, $I(k) \Vdash \alpha$, and the α-formula is also satisfied under interpretation I; similarly for β- and negative formulas. If $k\nu \in B$, then for some prefix k' the formula $k'\nu_0$ is also in B, and by the induction hypothesis $I(k') \Vdash \nu_0$. Using the definition of interpretation I, $I(k) \Vdash \nu$, and the $k\nu$ formula is satisfied under interpretation I. If $k\pi \in B$, then for some prefix k' the formula $k'\pi_0$ belongs to B, and $I(k') \Vdash \pi_0$. By the definition of Kripke models, $I(k) \Vdash \pi$, and the $k\pi$ formula is satisfied under interpretation I. However, the formula $0\mathrm{T}X$ also belongs to B and is satisfied under the interpretation, i.e. $I(0) = w \Vdash X$, which is a contradiction; hence X is S4-valid.

($\Rightarrow$) Suppose that X is an S4-valid formula and that $\mathscr{T}$ is a tableau containing $0\mathrm{T}X$ in its root. Let $M = \langle W, R, \Vdash \rangle$ be a Kripke model. Let I be an S4-interpretation mapping prefixes from tableau $\mathscr{T}$ into worlds from W, such that there is no branch from $\mathscr{T}$ whose set of all prefixed atomic signed formulas is satisfied under interpretation I. Let us consider the Kripke model $M_0 = \langle \mathrm{Pref}(\mathscr{T}), \rho, V \rangle$, where $\mathrm{Pref}(\mathscr{T})$ is the set of all prefixes from tableau $\mathscr{T}$, ρ the visibility relation between them and V a valuation defined as $V(j, Z) = \Vdash(I(j), Z)$ for the signed atomic formula Z. Then,

(a) since ρ is a reflexive and transitive relation, M_0 is an S4-model,

(b) $V(0, \mathrm{T}X) = \top$, since X is S4-valid and

(c) prefixed signed atomic formulas are satisfied under interpretation I iff they are satisfied under the S4-interpretation I_0 mapping $\mathrm{Pref}(\mathscr{T})$ into itself, such that $I_0(j) = j$ for every $j \in \mathrm{Pref}(\mathscr{T})$.

Now, we shall inductively choose a branch B whose set of all prefixed signed atomic formulas is satisfied under interpretation I_0. In the first step B contains only $0\mathrm{T}X$, because M_0 is an S4-model, X an S4-valid formula and, for any $k \in \mathrm{Pref}(\mathscr{T})$, $k \Vdash \mathrm{T}X$. After i steps we have chosen an initial segment of the branch. Consider the formula Y from that segment. If it is atomic, it is also satisfied by hypothesis. If Y is an α-formula satisfied under interpretation I_0, its α_1- and α_2-components are also satisfied, and they are added to the branch; similarly for ν- and negative formulas. If Y is a β-formula satisfied under interpretation I_0, at least one of its components is also satisfied. We choose that one and add it to the branch. If Y is a π-formula with prefix k, then by the construction rules, for every prefix k' visible from k, there is a branch

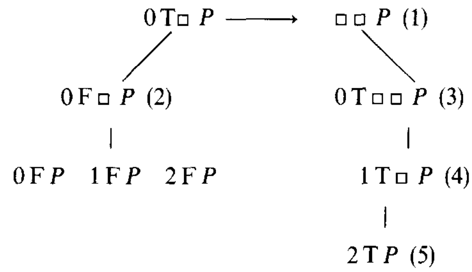


Fig. 3.

containing the formula $k'\pi_0$. However, only these prefixes (worlds from model M_0) are visible from k, and by the definition of Kripke models there is at least one k' such that $k \rho k'$ and $k' \Vdash \pi_0$. We add the node containing the formula $k'\pi_0$ to our branch. In this way we choose the tableau's branch whose set of all prefixed signed formulas is satisfied under interpretation I_0. The same holds for the set of all atomic formulas from that branch. However, atomic prefixed signed formulas are satisfied under interpretation I iff they are satisfied under interpretation I_0, and the branch B satisfies the lemma's requirement. □

Example 3.4. Figure 3 shows the tableau for the formula $\Box P \to \Box\Box P$. By applying the β-rule on node (1) we get nodes (2) and (3). By applying the ν-rule on node (3) we get a prefix 1 and a node (4). A new application of the ν-rule on node (4) introduces prefix 2 and a node (5). Prefixes 1 and 2 are visible from 0, and by applying the π-rule the branch containing node (2) branches into three new branches. Hence, for every S4-interpretation either 2 F P or 2 T P are satisfied, according to the previous lemma, and the formula is S4-valid.

After Lemma 3.3, the issue is how one should examine whether or not the tableau's branches are satisfied for every interpretation. We will connect satisfaction under interpretation with validity in the classical propositional logic using independence of values of atomic formulas in different worlds of Kripke models.

First, consider an arbitrary interpretation I which maps the tableau's prefixes into worlds of model $\langle W, R, \Vdash \rangle$. Next, consider an atomic formula with interpreted prefixes as propositional variables where the world, the picture of the prefix, becomes index. Then sets from branches become conjunctions of propositional variables. We call these conjunctions dual clauses induced by the tableau and the interpretation. In Example 3.4 dual clauses are: $\{\mathrm{F}\, P_{I(0)}\}$, $\{\mathrm{F}\, P_{I(1)}\}$, $\{\mathrm{F}\, P_{I(2)}\}$ and $\{\mathrm{T}\, P_{I(2)}\}$. The induced set of dual clauses is the set of all dual clauses corresponding to the sets of prefixed atomic formulas from the tableau's branches. Define an induced propositional valuation $\Vdash_p$ such that $\Vdash (I(k), X) = \Vdash_p X_{I(k)}$, where X is a signed atomic formula. For every model and every interpretation there is an induced valuation, and vice versa, for every propositional valuation and the frame $\langle W, R \rangle$ there is a model $\langle W, R, \Vdash \rangle$, where $\Vdash_p$ is induced by $\Vdash$. Obviously, the set of all prefixed signed atomic formulas from the tableau's branch is satisfied under interpretation I iff the corresponding dual clause is satisfied under valuation $\Vdash_p$. If an induced set of dual clauses is valid then, for every interpretation I which induces that set, at least one branch from the corresponding tableau satisfies the requirement of Lemma 3.3.

Now, the question is how one should establish the validity of all sets of dual clauses induced by the tableau $\mathscr{T}$ and all interpretations of the tableau's prefixes? The solution will be given in a few steps.

Let I_0 be an S4-interpretation from $\mathrm{Pref}(\mathscr{T})$ into worlds of model $\langle \mathrm{Pref}(\mathscr{T}), \rho, \Vdash \rangle$, where $\mathrm{Pref}(\mathscr{T})$ is the set of all prefixes of tableau $\mathscr{T}$, ρ is the visibility relation between them, $\Vdash$ is an arbitrary valuation and, for any $j \in \mathrm{Pref}(\mathscr{T})$, $I_0(j)=j$. In order to establish the satisfaction of at least one dual clause from the induced set the dual resolution rule [18] is used:

(DR) if S_1 and S_2 are dual clauses, $\mathrm{T}\,A \in S_1$ and $\mathrm{F}\,A \in S_2$ for an atomic formula A, then by resolving these clauses we get their resolvent $R(S_1, S_2, A) = (S_1 \backslash \{\mathrm{T}\,A\}) \cup (S_2 \backslash \{\mathrm{F}\,A\})$.

In derivations called the dual resolution procedure, we allow only dual clauses corresponding to the finished branches and their resolvents. If S is the induced set of dual clauses from a tableau $\mathscr{T}$, $\emptyset$ the empty clause, and $R(S) = S \cup \{C\colon C$ is the resolvent of two clauses from $S\}$, $R_0(S)=S$, $R_i(S)=R(R_{i-1}(S))$ and $R^*(S)=\bigcup\{R_i(S)\colon i \geqslant 0\}$, the following lemma holds.

Lemma 3.5. *A set of dual clauses is valid iff* $\emptyset \in R^*(S)$.

The proof of the lemma is given in [18].

What can we say about the other interpretations? Suppose that we can infer the empty clause from the set of dual clauses induced by I_0. Obviously, the same holds for any 1–1 interpretation. We just follow the same order of inference as for I_0. In fact, the only difference is in a change of the names of propositional variables from dual clauses. At last, there are non-1–1 interpretations. The problems arise in two situations:

(a) We use in inference (for I_0) the dual clause $\{X\,A_{I(l)}, X\,A_{I(j)}, \Psi\}$, where X is a sign and $I(j)=I(l)$; however, this clause is a subset of $\{X\,A_{I(l)}, X\,A_{I(j)}, \Psi\}$, where $I(l) \neq I(j)$, and we do not lose anything. It comes from $\models \{(A \wedge B), C\} \Rightarrow \models \{A, C\}$.

(b) We resolve dual clauses $C_1 = \{\Phi, \mathrm{T}\,A_{I(l)}, \mathrm{F}\,A_{I(j)}\}$ and $C_2 = \{\Psi, \mathrm{F}\,A_{I(l)}\}$ and get their resolvent $C_3 = \{\Phi, \Psi, \mathrm{F}\,A_{I(j)}\}$, where $I(j)=I(l)$; however, $C_2 \subset C_3$, and so, again, we do not lose anything. It follows that we can also infer the empty clause.

Now, if it is possible to infer the empty clause from the set of dual clauses induced by I_0, then for every valuation at least one set of atomic formulas is satisfied, and the formula whose tableau is examined is valid according to Lemma 3.3. If the formula is valid, then for any valuation a set of atomic formulas from at least one branch of its tableau is satisfied and it is possible to infer the empty clause by applying the dual resolution procedure. Using the introduced notation we can formulate the following lemma.

Lemma 3.6. *A formula X is S4-valid iff the empty clause can be inferred from the set of all dual clauses induced by the interpretation I_0 and the formula's tableau $\mathscr{T}$.*

Example 3.7. In Example 3.4 the dual clauses are $\{\mathrm{F}\, P_{I(0)}\}, \{\mathrm{F}\, P_{I(1)}\}, \{\mathrm{F}\, P_{I(2)}\}$ and $\{\mathrm{T}\, P_{I(2)}\}$. An application of the (DR)-rule on the third and fourth clauses infers the empty clause. According to Lemma 3.6, formula $\Box\, P \to \Box\Box\, P$ is S4-valid.

An appropriate choice of the order in which the construction rules should be applied will guarantee that every node will be processed, and consequently that for every formula there is its dual tableau.

Since all the rules introduce only a finite number of nodes and prefixes, after a finite number of reduction rules are applied, there would be only finite numbers of nodes and prefixes suitable for reduction. Consider a situation in an arbitrary stage of a tableau construction. Let there be N reducible nodes and M π-nodes between them, and let there be K prefixes which are applicable on the studied π-nodes. We will not reduce anything else until we finish with these nodes and prefixes. We can reduce $N-M$ non-π-nodes in $N-M$ steps and M π-nodes in less than $M*K+1$ steps. As K is finite, all the N nonreduced nodes can be processed in a finite number of steps. So, for every stage of the reduction we can guarantee that all immediately reducible nodes will be processed.

Let t be a tree from a sequence T of trees constructed by the application of rules 1 and 2 on formula X. Let S be the set of dual clauses induced by interpretation I_0 and finished branches from t, and let $R^*(S)$ be the closure of set S under (DR).

Definition 3.8. The tree t is the *proof* for formula X in the system of dual tableaux if the empty clause belongs to $R^*(S)$.

Completeness theorem for S4 (propositional case). *A modal formula X is S4-valid iff it has a finite proof in the system of dual tableaux.*

Proof. ($\Leftarrow$) If formula X has a finite proof, then according to ($\Leftarrow$) of Lemma 3.3 it is S4-valid.

($\Rightarrow$) Suppose that formula X is S4-valid. By ($\Rightarrow$) of Lemma 3.3, under arbitrary S4-interpretation there is a branch in the tableau whose set of all prefixed signed atomic formulas is satisfied under that interpretation. According to Lemmas 3.5 and 3.6, the set of all dual clauses induced by the tableau and interpretation I_0 is valid. Because of the compactness of the propositional calculus, in the valid set of dual clauses there is a finite subset S with the same property. Let n be the largest among prefixes from the tableau's branches corresponding to the dual clauses from S.

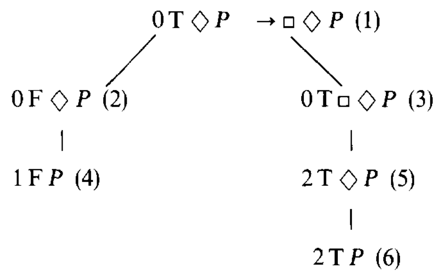


Fig. 4.

Let T be the sequence of trees defining the formula's tableau. Let $t_m \in T$ be the last tree containing no prefix greater than n. This tree is finite, because the studied formula X has only a finite number of subformulas. There are only a finite number of prefixes in t_m. If all the mentioned branches are in t_m, it is a finite proof for X. Otherwise, we construct a new sequence T' which begins with the first m elements from sequence T and continues with the elements made by reduction of branches with no prefixes greater than n. There is no problem in doing this, because all these prefixes have already been introduced. Since the number k of reduction steps in finite, so is the number k of new trees from T'. It follows that t_{k+m} is a finite proof of the examined formula X.

Example 3.9. Figure 4 shows the tableau for formula $\Diamond P \to \Box \Diamond P$. By applying the β-rule on node (1) we get nodes (2) and (3). By applying the ν-rule on nodes (2) and (3) we get prefixes 1 and 2 and nodes (4) and (5). Then the π-rule is applied to node (2) to obtain node (6). The tableau does not contain any more unfinished nodes, and a set of dual clauses $\{\{\mathrm{F}\,P_1\}, \{\mathrm{T}\,P_2\}\}$ is obtained which, obviously, cannot produce the empty clause; so the formula is not S4-valid.

We do not infer the empty clause in Example 3.9, and $\Diamond P \to \Box \Diamond P$ is not a theorem of our system. Now we are in a position to construct the formula's counter model. We consider a model M and an interpretation I so that no tableau branch is satisfied under the interpretation. We choose the model $M = \langle \mathrm{Pref}(\mathscr{T}), \rho, \Vdash \rangle$, where $\mathrm{Pref}(\mathscr{T})$ is the set of prefixes from the tableau, i.e. $\mathrm{Pref}(\mathscr{T}) = \{0, 1, 2\}$, ρ is its visibility relation, i.e. $\rho = \{(0,0), (0,1), (0,2), (1,1), (1,2), (2,2)\}$, and $\Vdash$ is a valuation so that under the identity interpretation I_0 there is no satisfied branch from the tableau. For instance, we can use an arbitrary valuation $\Vdash$, so that $1 \Vdash \mathrm{T}\,P$, and $2 \Vdash \mathrm{F}\,P$. Obviously, then $0 \Vdash \mathrm{T} \Diamond P$, and it is not $0 \Vdash \mathrm{T} \Box \Diamond P$. It follows that it is not $0 \Vdash \Diamond P \to \Box \Diamond P$, and model M is a counter model of the formula.

Such an examination can be applied whenever we have not proved a formula.

4. Other modal logics

By similar considerations it is possible to formulate formal systems of dual tableaux for some other normal modal logics (T, B, S5, D, D4, DB) and also for the intuitionistic logic. Only the visibility between prefixes in the tableaux should be changed. For example, for S5 it is the relation of equivalence, and for D-logic it is an ideal relation, i.e. $(\forall w)(\exists u) w \rho u$. For the intuitionistic logic, it is well known that it is interpretable into S4, so we convert an intuitionistic formula to its S4-equivalent and then use the dual tableau method for S4.

Example 4.1. Figure 5 shows the S5-tableau for the formula $\Diamond P \rightarrow \Box \Diamond P$. Different from Example 3.7, application of the π-rule on node (5) introduces three new nodes due to the symmetry of the visibility relation. A set of dual clauses is obtained, $\{\{\mathrm{F}\,\mathrm{P}_1\}, \{\mathrm{T}\,\mathrm{P}_0\}, \{\mathrm{T}\,\mathrm{P}_1\}, \{\mathrm{T}\,\mathrm{P}_2\}\}$, from which an empty clause is inferred by a single application of the (DR)-rule on the first and third clauses.

So far, our approach requires at least idealization to justify the introduction of new worlds by the ν-rule, and it is not directly suitable for the family of K-logics (K, KB, K4). Models of these logics might contain so-called dead ends [16]. They are worlds which are not related to any worlds at all, even to themselves. For an interpretation I the question arises when we have prefixes k and k', such that k' is visible from k, and $I(k)$ is a dead end. Since $I(k')$ should not be defined, the claims about our system would not hold. Following the definition of valuation we conclude that for every ν-formula and prefix k, such that $I(k)$ is a dead end, $I(k) \Vdash \nu$, but $I(k)$ does not satisfy any π-formula. Using these facts the dual tableau system can be adapted to the mentioned K-logics. We will discuss only K-logic, and the rest follows easily. First, we extend some definitions.

Definition 4.2. K-interpretation is a partial mapping from a set P of prefixes into a set W of worlds of some K-model $M = \langle W, R, \Vdash \rangle$, i.e. $I : P \rightarrow W$, when for every $k \in P$ the following is satisfied:

(a) if $I(k)$ is a dead end, then, for every prefix k' visible from k, $I(k')$ is not defined;

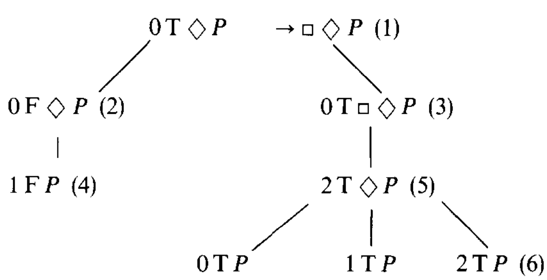


Fig. 5.

(b) if $I(k)$ is not defined, then, for every prefix k' visible from k, $I(k')$ is not defined;

(c) if $I(k)$ is defined and it is not a dead end, then, for every prefix k' visible from k, $I(k) R I(k')$.

Definition 4.3. For a prefix k, general atomic formulas (GAF) under a K-interpretation I are:

(a) if $I(k)$ is defined, then a prefixed atomic formula kA is a GAF, and

(b) if $I(k)$ is a dead end, the formulas $k\nu$ and $k\pi$ are GAFs.

Definition 4.4. A dual clause is acceptable under a K-interpretation I if it contains all atomic GAFs from a finished tableau branch and the branch does not contain any GAF π-formula (under interpretation I).

Hence, no conditions are placed on the visibility relation between worlds in K-models, the relation ρ (visibility between prefixes) contains (k, k') iff prefix k' is introduced by an application of the ν-rule to a formula $k\nu$. For the K4-logic we additionally require transitivity, and for the KB-logic symmetry. Now, we can reformulate the key Lemma 3.3. The proof of the new lemma is very much like the previous one, so we do not repeat it. However, we emphasize that satisfaction under K-interpretation has some extended meaning, as has been mentioned. The (DR)-rule, Lemmas 3.5 and 3.6 and the completeness theorem are applicable directly.

Lemma 4.5. *Let $\mathscr{T}$ be a tableau whose root contains the formula* $0\,\mathrm{T}\,X$. *A formula X is K-valid iff, for each K-model $\langle W, R, \Vdash \rangle$ and for each K-interpretation I which maps the tableau's prefixes into the worlds from W, there exists at least one branch of the tableau whose set of all GAFs is satisfied under interpretation I.*

If a dual clause acceptable under a K-interpretation contains no formulas (the corresponding branch contains only GAF ν-formulas, and no atomic or π-GAFs), we treat it as an empty clause. We divide the set of all K-interpretations into groups, such that every K-interpretation from a group maps the same prefixes into dead ends. In any group of interpretation we consider only one $1-1$ interpretation, and try to infer an empty clause from the set of acceptable clauses. If we succeed, the formula is a theorem of our system, in the other case it is not.

Example 4.6. Figure 6 represents the K-tableau of the K-axiom $\Box(P\to Q)\to(\Box P\to \Box Q)$. The only application of the ν-rule is in node (1). So, we have two groups of K-interpretations: the first contains all interpretations that map the prefix 0 into a dead end and the second contains interpretations that do not. The set of acceptable dual clauses corresponding to the first group contains only one element – the empty clause, which is induced by the branch that contains node (1), because there are π-GAFs in nodes (2) and (3). The set $\{\{\mathrm{F}\,P_1\}, \{\mathrm{T}\,Q_1\}, \{\mathrm{T}\,P_1, \mathrm{F}\,Q_1\}\}$ corresponds to the second group of interpretations. Obviously, by two applications of the (DR)-rule

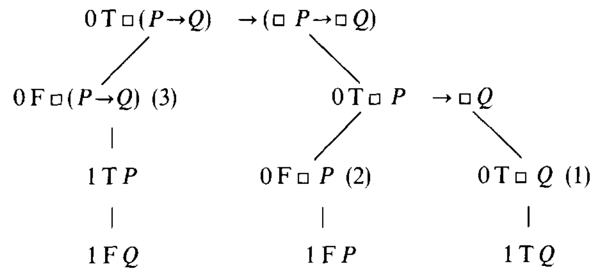


Fig. 6.

we can infer the empty clause. It follows that the formula is a theorem of our system, and consequently K-valid.

5. Decidability

A proof procedure could be described in the following way:

```
put the formula in the tableau root;
S:=empty set;
while not(end of reduction) and not(empty clause∈R*(S)) do
begin
   apply one of the reduction rules;
   if (a branch is finished) then
   begin
   add the dual clause (corresponding to the finished branch and
   induced by interpretation I0) to the set S;
   S:=R*(S);
   end;
end;
```

Set S is a set of dual clauses induced by interpretation I_0 and the finished tableau branches. $R^*(S)$ is the closure of set S under the (DR)-rules. The order of the application of the construction rules must guarantee that every node will be processed.

The formula in question is a theorem iff the empty clause is inferred.

If every dual tableau is finite, so is our proof procedure, because dual resolution over a finite set of clauses should be finished in finite time. Every tableau's branch is finite because the reduction rules decrease the number of logical operators of formulas. A dual tableau is finite if every π-node has only a finite number of descendants, or if every prefix can access only a finite number of prefixes.

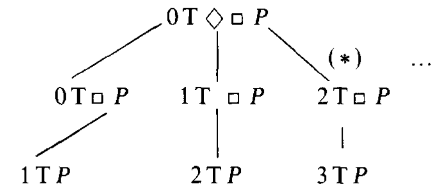


Fig. 7.

If the visibility relation is not transitive, any prefix can access only prefixes which it has introduced by the ν-rule, itself (if visibility is reflexive) and the prefix which has introduced it (if visibility is symmetric). Since a formula has only a finite number of subformulas, any prefix can introduce only a finite number of new prefixes, so any prefix sees only a finite number of prefixes. It follows that the dual tableau is a decision procedure for nontransitive logics.

If the visibility relation is transitive the procedure may not terminate, as is shown in Example 5.1. What has happened? The prefix 0 sees not only prefix 1, which it introduced, but also prefixes 2, 3, etc., because of transitivity. It is possible to modify the dual tableau system to go around the problem, keeping the completeness. We just follow the idea of [10].

Example 5.1. For the formula $\Diamond \Box P$ the S4-tableau is an infinite one (see Fig. 7, where $(*)$ denotes that the construction will never terminate).

A chain is a sentence of prefixes where every prefix sees its successors and is seen by its predecessors. A chain can be infinite if the visibility relation is transitive, as in Example 5.1, where the chain is composed of prefixes 0, 1, 2, etc. Since any formula has only a finite number of subformulas, in any infinite chain there must be at least two prefixes corresponding to the same set of signed subformulas of the studied formula (correspondence means that the prefix stands beside the subformula in the formula's tableau). Let them be prefixes p_i and p_j, where $i<j$. After p_j the chain becomes periodic and we do not get more information than from the initial segment concluded with p_j. The initial segments begin with prefix 0 and are finite. Also, for any tableau there are only finite numbers of chains, because prefix 0 can introduce only a finite number of prefixes. In every π-node we discard prefixes whose corresponding predecessors have already been used. We cannot lose anything because dual clauses which are not generated have the same shape as the used ones. The only difference is in indexes, but they are also periodical like the prefixes. So, we can infer the empty clause from a reduced set of dual clauses iff we can do it with the whole induced set. Hence, in every π-node we keep only a finite number of descendants, the tableau is finite and the modified procedure is actually a decision procedure.

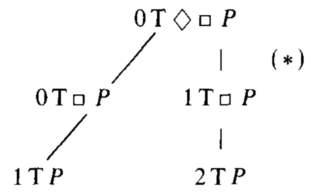


Fig. 8.

Example 5.2. For the formula $\Diamond \Box P$ the modified S4-tableau is a finite one (see Fig. 8, where $(*)$ denotes where the construction is changed).

6. First-order modal logics

In the first-order case the language is extended with the new symbols: quantifiers $(\forall, \exists)$, constants and relation symbols. We have got two new groups of signed formulas. They are shown in Fig. 9, where c is a constant and $A(c)$ is an instance of the formula $A(x)$. Constant symbols are rigid, meaning the same thing in each world. We assume that we have first-order Kripke models defined as usual, with nonempty quantifier domains associated with the model worlds. The domains can meet the following conditions:

(a) every world has the same domain, and

(b) if world w is visible from world u, then the domain of w is a superset of the domain of u.

In the former case the models are constant domain models, and in the latter models have monotonic domains. Assuming that all formulas appearing in a tableau have no free variables, as well as in [10, 34], we start with a discussion of constant domain models.

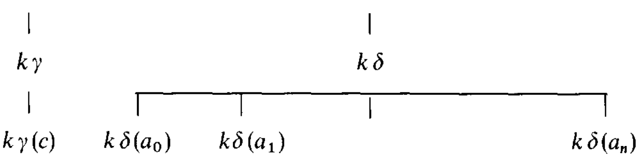
The dual tableaux for first-order modal logics are simply extensions of the propositional modal dual tableaux. The additional γ- and δ-rules are shown in Fig. 10. By the γ-rule a new constant is introduced. By the δ-rule, a branch containing a δ-node branches for every constant occurring in the tree. These rules are, in fact, independent of the other rules and the types of modal logic.

We eventually allow that the first constant in a tableau could be introduced by the δ-rule. This happens when we have no constant in the tableau and we cannot apply any other rule. The rules for quantifiers are the same as those for classical quantifiers [18] and similar to the modal ν- and π-rules. It is worth noting that the subformulas of the examined formula are reduced to the atomic level. An atomic sentence could be treated as a propositional variable, and dual resolution is done over dual clauses that contain only atomic sentences. So, completeness proofs for first-order modal logics are combinations of the proofs given in [18] and in this paper, and we omit them.

γ	$\gamma(c)$
$\mathrm{T}(\forall x)A(x)$	$\mathrm{T}\,A(c)$
$\mathrm{F}(\exists x)A(x)$	$\mathrm{F}\,A(c)$

δ	$\delta(c)$
$\mathrm{T}(\exists x)A(x)$	$\mathrm{T}\,A(c)$
$\mathrm{F}(\forall x)A(x)$	$\mathrm{F}\,A(c)$

Fig. 9.



c is a new constant $a_0, a_1, \ldots, a_n$ occur already in the tree

Fig. 10.

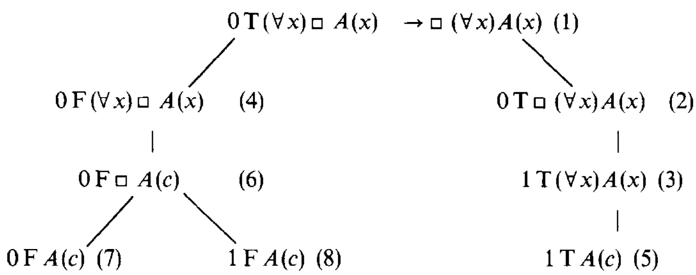


Fig. 11.

Example 6.1. Figure 11 represents the first-order S4 (or T, or S5)-tableau of the so-called Barcan formula. By applying the propositional rules we get nodes (2), (3) and (4). Then, by the γ-rule we introduce node (5) and the constant c. Hence, the model is with constant domain, the constant c exists in the domain of world 0 and we apply the δ-rule on node (4). By the π-rule nodes (7) and (8) are created. The set of dual clauses $\{\{\mathrm{F}\,A(c)_0\}, \{\mathrm{F}\,A(c)_1\}, \{\mathrm{T}\,A(c)_1\}\}$ is obtained. By a single application of the (DR)-rule we could infer the empty clause, and the examined formula is a theorem of our system.

When Kripke models with monotonic domain are considered, the Barcan formula is no longer valid. To suit such models, we have to change our π-rule. As in [10], we associate prefixes and constants. A prefix k is associated with a constant c if the constant is introduced with a $k\gamma$-formula, or if there is a prefix k', such that $k'\,\rho\,k$ and k' is associated with c. The π-rule should be applied to a $k\pi$-formula only for constants that are associated with prefix k. In Example 6.1, we cannot use the constant c with the prefix 0, and we cannot infer the empty clause. It follows that the Barcan formula is not a theorem of the system of dual tableaux for models with monotonic domains.

7. Conclusion

Our modal dual tableau method is an extension of the dual tableau proof procedure for classical logic. We have an integrated notion of Kripke models in the rules from [18] and have a combination of a tableau-like approach and a resolution-like inference over classical propositional clauses. Since we do not use it, we do not worry that the normal modal form does not exist for most modal logics. During the tableau reduction of a formula we eliminate modal operators supposing that we jump to new worlds of a model. These worlds are explicitly denoted by their names, i.e. prefixes. Such an approach ensures that the system is suitable for many kinds of modal logic, including logics whose models require symmetry. Step-by-step reduction of a formula enables staightforward extension to the first-order modal logic.

There are some other proof procedures for modal logics, more or less similar to the dual tableau method. Among them the best known are: modal tableau and the so-called destructive resolution given by Fitting [10, 12, 13], modal resolution in clausal form, from Farinas, Auffray, Enjalbert and Hebrard [5, 9], its first-order extension from Cialdea [7, 8] and nonclausal resolution from Abadi and Manna [1]. Different from the dual tableaux all these methods are refutation procedures.

The resolution of Farinas and others differs from that of Abadi and Manna in many ways; for instance, the first one contains formula translation to some sort of normal form, while the second does not need it, etc. However, both of them include many inference rules that are, let us say, complicated (at least, they are so judged in [25]) and are hard to implement [4]. It seems that the nonexistence of a reasonable normal form for most modal logics is the main source of problems for modal resolution in clausal form. We reduce the resolution at the classical propositional level, where all the problems are solved and intuition is clear. So, we believe that our system overcomes such problems and has an advantage as far as simplicity is concerned.

The most attractive methods for modal theorem proving have been given by Fitting. His destructive and prefixed tableaux became almost "classical" modal tableaux. A generalization of them, the prefixed tableau, is an extension of Smullyan's tableau [34] and has had great influence on our system, which extends the dual tableau in the same way. Prefixed and dual tableaux are really dual to each other. When one requires atomic closure of branches in the former, both of them have the same number of nodes. It is because starting formulas are mutually conjugate, the classical reduction rules are the same, and the ν- and π-rules are symmetrical. Both systems are particularly appropriate for symmetric modal logics and first-order logics with constant domains, but they are also suitable for other kinds of modal logics. The destructive tableau covers logics that do not involve symmetry and first-order logics with monotonic domains. For such logics, when one switches the context, i.e. when one leaves one world and jumps to another, the old context could be destroyed. In destructive tableaux, these jumps are done when π-formulas are reduced to their π_0-components, and some other formulas are forgotten. The jumps correspond to a pruning of parts of the (prefixed) tableau tree. However, the choice of a π-formula for the π-rule is ambiguous. These choice points must be remembered, and if a proof is not obtained one has to backtrack to the last one and try again. An alternative way is to create parallel processes whenever the π-rule is applied. So, there is some doubt about the benefit of destructive context switching. It seems to us that an extension of the classical dual tableau in such a "destructive" way is straightforward, but so far we have not tried that.

Fitting's destructive resolution was designed with a strong bias toward the destructive tableau. This is obvious when one analyzes the rules. In destructive resolution the reduction and resolution rules can be intermingled, and in this sense the system is nonclausal. At the classical propositional level the set of clauses generated by destructive resolution corresponds to the conjunctive normal form of a negated formula. Hence, we start with a formula itself (not a negated one), and in the dual tableau [18] we generate in fact the same set of clauses, although the corresponding literals are mutually conjugate. Then one may proceed with resolution inferences in the same manner. The differences appear when modal operators are included. Thanks to the destructive nature of its modal reduction rules, Fitting's method works with a smaller number of clauses than the dual tableau. Also, the clauses themselves may be smaller than the dual clauses. However, the benefit of context switching is under suspicion, as in the destructive tableau method. There is an additional rule in destructive resolution that also involves branching points in proofs. It is the Special Case Rule and it could not be eliminated. By this rule many branches of proofs may be created and every one must produce the empty clause. The problem arises with the first-order case when additional communication between parts of clause sets is also required. There are no such issues in the dual tableau. Destructive resolution includes skolemizing during resolution inference, which rapidly decreases the number of clauses. In the dual tableau we use only ground clauses and there is a real danger of combinatorial explosion. On the other hand, the substitution rule which corresponds to skolemization is problematic when it comes to implementation.

Finally, the great advantage of Fitting's systems (compared with ours) is the elegant way in which some weak logics, like K-logic, are solved. It is a consequence of the definition of valuation in modal models, where ν- and π-formulas are not completely symmetrical.

We believe that one can understand our system of dual tableaux easily, even more easily than most present systems. The clarity of the system has made it possible that a modal theorem prover based on the procedure described in Section 5 has been realized very simply. However, this does not guarantee efficient implementation. The biggest problem corresponds to the possibility of combinatorial explosion of ground dual clauses and we try to reduce their number by discarding superset clauses. Any other improvement would be recommended, but the most promising solution is skolemization. We shall first incorporate such an approach in the classical dual tableau. The next step may be the application of that procedure to the tableau prefixes. It would be straightforward because ν- and π-formulas behave almost like classical γ- and δ-formulas. Other approaches may be the world-unification method introduced in [3, 17, 36], or the dummy variables method [31].

An important advantage of the dual tableau proof procedure is its suitability for parallel execution. We have found two main possibilities for parallelization: tableau construction is independent of the dual resolution procedure up to the generation of dual clauses, and dual clauses are ground clauses and can be resolved in a fully distributed manner, without any additional communication overhead. A parallel theorem prover based on dual tableaux and the ideas mentioned was presented in [28]. It is an extension of the classical version described in [20].

In addition to the discussed improvements of the existing system we hope to extend the dual tableau method to some other modal logics acknowledging its generality.

Acknowledgment

This work was supported by Science Fund Serbia, grant number 0403, through Mathematical Institute, Belgrade. I am grateful to Miodrag Kapetanović, Aleksander Krapež, and an anonymous referee, for helpful comments, and to Zsolt Zolnai and Diana Kramer for their efforts in improving my poor knowledge of English.

References

[1] M. Abadi and Z. Manna, Non-clausal temporal deduction, in: R. Parikh, ed., *Logics of Programs*, Lecture Notes in Computer Science, Vol. 173 (Springer, Berlin, 1985) 1–15.

[2] M. Abadi and Z. Manna, Modal theorem proving, in: *Proc. 8CAD*, Lecture Notes in Computer Science, Vol. 230 (Springer, Berlin, 1986) 172–186.

[3] T. Arragi, A uniform prefixed tableau method for positive first-order definable systems, in: *Proc. 1st Workshop on Theorem Proving with Analytic Tableaux and Related Methods*, Lautenbach, Germany (1992).

[4] Y. Auffray and P. Enjalbert, Modal theorem proving: an equational viewpoint, *J. Logic Comput.* **3** (1992) 243–295.

[5] Y. Auffray, P. Enjalbert and J. Hebrard, Strategies for modal resolution: results and problems, *J. Automat. Reason.* **6** (1990) 1–38.

[6] L. Catach, Tableaux: a general theorem prover for modal logics, *J. Automat. Reason.* **7** (1991) 489–510.

[7] M. Cialdea, Une méthode de déduction automatique en logique modale, Thèse de l'Université P. Sabatier, Toulouse, France, 1986.

[8] M. Cialdea, Towards first order resolution for the modal systems T and S4, Tech. Report, LSI, Université P. Sabatier, Toulouse, France, 1989.

[9] P. Enjalbert and L. Farinas, Modal resolution in clausal form, *Theoret. Comput. Sci.* **1** (1989) 1–33.

[10] M. Fitting, *Proof Methods for Modal and Intuitionistic Logic* (Reidel, Dordrecht, 1983).

[11] M. Fitting, Resolution for intuitionistic logics, in: Z.W. Ras and M. Zemankova, eds., *Methodologies for Intelligent Systems* (North-Holland, Amsterdam, 1987) 400–407.

[12] M. Fitting, First-order modal tableaux, *J. Automat. Reason.* **4** (1988) 191–213.
[13] M. Fitting, Destructive modal resolution, *J. Logic Comput.* **1** (1990) 83–97.
[14] C. Geissler and K. Konolige, A resolution method for quantified modal logics of knowledge and belief, in: J.Y. Halpern, ed., *Theoretical Aspects of Reasoning about Knowledge* (Morgan Kaufmann, Los Altos, CA, 1986).
[15] G.E. Hughes and M.J. Cresswell, *An Introduction to Modal Logic* (Methuen, London, 1972).
[16] G.E. Hughes and M.J. Cresswell, *A Companion to Modal Logic* (Methuen, London, 1984).
[17] P. Jackson and H. Reichgelt, *Logic-Based Knowledge Representation* (MIT Press, Cambridge, MA, 1989).
[18] M. Kapetanović and A. Krapež, A proof procedure for the first order logic, *Publ. Inst. Math.* **59** (1989) 3–5.
[19] K. Konolige, Resolution and quantified epistemic logics, Lecture Notes in Computer Science, Vol. 230 (Springer, Berlin, 1986) 199–208.
[20] A. Krapež, A. Kapetanović, Z. Ognjanović and T. Petrović, Prover 91 – a parallel theorem prover, in: *Proc. 1st Workshop on Theorem Proving with Analytic Tableaux and Related Methods*, Lautenbach, Germany (1992).
[21] S. Kripke, Semantical analysis of modal logics I, normal propositional calculi, *Z. Math. Logik Grundlag. Math.* **9** (1963) 67–96.
[22] S. Kripke, Semantical consideration on modal logics, *Acta Philos. Fenn.* (1963) 83–94.
[23] G.E. Mints, Structural synthesis with independent subproblems and the modal logic S4, *Eesti NSV Tead. Akad. Toimetised Füüs.-Mat.* **33** (1984) 147–151.
[24] G.E. Mints, Resolution calculi for modal logics, *Eesti NSV Tead. Akad. Toimetised Füüs.-Mat.* **35** (1986) 279–290.
[25] G.E. Mints, Review of "Some remarks on the possibility of extending resolution proof procedures to intuitionistic logic", by M. Cialdea, *Math. Rev.*, review 87j:03016 (1987).
[26] G.E. Mints, Cutfree formalizations and resolution method for propositional modal logics, in: *Proc. LMPS Congr.*, Moscow (1987).
[27] G.E. Mints and E. Tyugu, The programming system Priz, *J. Symbolic Comput.* (1987).
[28] Z. Ognjanović, D. Urošević, T. Petrović and Z. Ristović, A tableau related method for modal theorem proving, in: *Proc. 1st Workshop on Theorem Proving with Analytic Tableaux and Related Methods*, Lautenbach, Germany (1992).
[29] H.J. Ohlbach, A resolution calculus for modal logics, in: *Proc. CADE-9*, 500–516.
[30] H.J. Ohlbach, Semantic-based translation methods for modal logics, *J. Logic Comput.* **5** (1991) 691–746.
[31] S. Reevs, Semantic tableaux as a framework for automated theorem-proving, advances in artificial intelligence, in: J. Hallam and C. Mellish, eds., *Proc. 1987 ASIB Conf.* (Wiley, New York, 1987) 125–139.
[32] J.A. Robinson, A machine-oriented logic based on the resolution principle, *J. ACM* **12** (1965) 23–41.
[33] M. Sato, A study of Kripke-type models for some modal logic by Gentzen's sequential calculus, Tech. Report Institute for Math. Science, Kyoto University, 1977.
[34] R. Smullyan, *First-Order Logic* (Springer, Berlin, 1968).
[35] L. Wallen, Matrix proof methods for modal logics, in: *Proc. 10th ICALP*.
[36] L. Wallen, *Automated Proof Search in Non-Classical Logics* (MIT Press, Cambridge, MA, 1989).
[37] L. Wallen and G. Wilson, A computational efficient proof system for S5, advances in artificial intelligence, in: J. Hallam and C. Mellish, eds., *Proc. 1987 ASIB Conf.* (Wiley, New York, 1987).