NOVI SAD J. MATH.
VOL. 30, NO. 2, 2000, 83-93

Proceedings of the TARA 2000 Conference
Novi Sad, Yugoslavia, September 6-7, 2000

ON TWO APPROACHES TO MODAL THEOREM PROVING

Zoran Ognjanović[1], Tatjana Timotijević[2]

Abstract. Two approaches to modal theorem proving are presented. The first approach is a direct one, developed specifically for modal logics. In the second approach a translation procedure is included and a classical first-order prover is used to the examined modal formulas. Some comparative results and possible extensions are discussed.

AMS Mathematics Subject Classification (1991): 68T15, 03B60

Key words and phrases: theorem proving, modal logic, translation of logics, classical logic

1. Introduction

In this paper we present two approaches to modal theorem proving. In the first one we use a prover dedicated to modal logics [5, 6, 7]. In the second approach, instead of working on the modal syntax including modal operators, formulas are translated into the classical first-order syntax, and then a classical prover [8] is applied. The idea behind the translation is to introduce special relational symbols that represent modal accessibility relations [2]. In Section 2 we give an overview of the considered proving procedure. The full descriptions, as well as the proofs of the given statements, can be found in [6, 8]. In Section 3, the translation procedure is given, and some results of the provers are considered. Section 4 contains a conclusion and some directions of further investigation.

2. A modal and a classical proving procedure

2.1. Dual tableau for normal modal logics

The propositional modal language L_M consists of unary logical operators ($\neg$, $\Diamond$ and $\Box$), binary logical operators ($\wedge$, $\vee$, and $\rightarrow$), a set $\Phi = \{p, q, r, p_1, \ldots\}$

[1]Institute of mathematics, Kneza Mihaila 35, 11000 Beograd, Yugoslavia, zorano@mi.sanu.ac.yu

[2]Faculty of Sciences, Radoja Domanovića 12, 34000 Kragujevac, Yugoslavia, shak@ptt.yu

of propositional variables and auxiliary symbols ("(" and ")"). The atomic formulas and formulas are defined as usual. For example p, $\neg(p \to \Box(p \vee \neg q))$, and $p \wedge \Diamond\Box p$ are formulas. A unifying notation which makes it easier to handle similar kinds of formulas in the same way is introduced in [1, 10]. We assume that T and F are new formal symbols, and if A is a formula of propositional modal language, then $T\ A$ and $F\ A$ are signed formulas. Signed non-atomic formulas are grouped in α, β, ν, and π formulas. Figure 1 contains these types of formulas and their respective components. Intuitively, an α-formula is true iff its components are also true. A β-formula is true iff either the corresponding β_1 or β_2-formula is true. To understand ν and π-formulas we need the notion of satisfiability in Kripke models [1, 4].

α	α_1	α_2
$T\ A \wedge B$	$T\ A$	$T\ B$
$F\ A \vee B$	$F\ A$	$F\ B$
$F\ A \to B$	$T\ A$	$F\ B$
$T\ \neg A$	$F\ A$	$F\ A$
$F\ \neg A$	$T\ A$	$T\ A$

β	β_1	β_2
$F\ A \wedge B$	$F\ A$	$F\ B$
$T\ A \vee B$	$T\ A$	$T\ B$
$T\ A \to B$	$F\ A$	$T\ B$

ν	ν_0
$T\ \Box A$	$T\ A$
$F\ \Diamond A$	$F\ A$

π	π_0
$T\ \Diamond A$	$T\ A$
$F\ \Box A$	$F\ A$

Figure 1: Signed formulas

Definition 1. *A Kripke model is a tuple $M = \langle W, R, v\rangle$, where W is a non-empty set of elements called worlds, R is a relation over $W \times W$ called visibility (or accessibility) relation, and $v : W \times \Phi \mapsto \{\top, \bot\}$ is a propositional valuation. The pair $\langle W, R\rangle$ is called a frame.*

A signed modal formula X is satisfied in a world $w \in W$ from the model M (or true in w), denoted by $w \models_M X$, if the following hold:

- *if $X = T\ p$, $p \in \Phi$, then $w \models_M T\ p$ iff $v(w)(p) = \top$,*
- *if $X = F\ p$, $p \in \Phi$, then $w \models_M F\ p$ iff $v(w)(p) = \bot$,*
- *if X is a α-formula α, $w \models_M \alpha$ iff $w \models_M \alpha_1$ and $w \models_M \alpha_2$,*
- *if X is a β-formula β, $w \models_M \beta$ iff $w \models_M \beta_1$ or $w \models_M \beta_2$,*
- *if X is a ν-formula ν, $w \models_M \nu$ iff $(\forall u \in W)((wRu) \Rightarrow u \models_M \nu_0)$, and*
- *if X is an π-formula π, $w \models_M \pi$ iff $(\exists u \in W)((wRu) \wedge u \models_M \pi_0)$.*

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

Definition 2. *A modal signed formula X is valid in a model $M = \langle W, R, v\rangle$ if $w \models_M X$ for any $w \in W$. The formula X is valid in a collection of models if it is valid in every corresponding model.*

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

We are interested in some classes of models definable by the condition put on the accessibility relation that are given in Figure 2 (a relation R is ideal iff $(\forall x)(\exists y)xRy$). Let $L \in \{K, D, D4, DB, T, S4, B, S5\}$ be a class of models. A (signed, unsigned) modal formula is L-valid if it is valid in every L-model.

D	R is an ideal relation
D4	R is ideal and transitive
DB	R is ideal and symmetric

T	R is reflexive
S4	R is reflexive and transitive
B	R is reflexive and symmetric
S5	R is an equivalence relation

Figure 2: Classes of models

Dual tableau procedure [5, 6, 7] is introduced as a tool for deciding validity of formulas in the mentioned modal logics. To describe it, we have to define the following notion.

Definition 3. *A prefix is an integer. If X is a signed formula and k is a prefix, then $k\ X$ is a prefixed signed formula.*

A dual tableau is a labeled tree whose nodes contain 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 will be used as names of worlds in that frame. Since the rules depend on the considered type of accessibility relation, we present the procedure for the logic S4, while the rules and the corresponding statements for the other logics are similar. The construction rules (illustrated in Figure 3) are:

1. a formula $0\ T\ A$ is placed in the tableau's root, where 0 is a prefix. The relation ρ marks visibility between prefixes, and at the beginning contains only $(0, 0)$. After introducing a new prefix, the relation ρ will be updated,
2. depending on the type of the formula at a node, one of the following rules should be applied:
   (a) if the node contains an α-formula with the prefix k, the branch where the node is located extends with nodes containing the subformulas $k\ \alpha_1$ and $k\ \alpha_2$,
   (b) if the node contains a β-formula with the prefix k, the branch where the node is located branches with nodes containing the subformulas $k\ \beta_1$ and $k\ \beta_2$,

(c) if the node contains a ν-formula with the 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 extends with a node containing the 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 the 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 $k''\nu_0$,

(d) if the node contains a π-formula with the prefix k, let the 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 extends with a node containing the π_0-formula and the prefix k'; say that 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 k' π_0-formula. 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.

$$
\begin{array}{cccc}
k\alpha & k\beta & k\nu & k\pi \\
| & / \quad \backslash & | & / \quad | \quad \cdots \quad \backslash \\
k\alpha_1 & k\beta_1 \;\; k\beta_2 & k'\nu_0 & k_0\pi_0 \;\; k_1\pi_0 \;\; \ldots \;\; k_n\pi_0 \\
| & & & \\
k\alpha_2 & & &
\end{array}
$$

Figure 3: The modal tableau reduction rules

Every node is reduced by the rules at most once on a particular branch. After that the node is finished. π-nodes are the only exceptions. A π-node with a 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. The 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 S4-dual tableau is the first tree from the sequence containing only finished nodes.

Let $\mathcal{T}$ be an S4-dual tableau. An S4-interpretation I is a mapping from the set P of prefixes from $\mathcal{T}$ into a set W of worlds of some S4-model $M = \langle W, R, v\rangle$ such that $(\forall k, k' \in P)(k\rho k' \Rightarrow I(k)RI(k'))$. A set U of prefixed signed formulas is satisfied under an S4-interpretation I if for every $k\ X \in U$, $I(k) \models X$.

Theorem 4. *Let $\mathcal{T}$ be an S4-dual tableau whose root contains a formula $0\ T\ A$, and P be the set of all prefixes from $\mathcal{T}$. The formula A is S4-valid iff for each S4-model $M = \langle W, R, v\rangle$, and for each S4-interpretation $I : P \to W$ there is at* *least one branch of the tableau whose set of all prefixed atomic signed formulas is satisfied under the interpretation I.*

To examine whether or not the tableau's branches are satisfied for every interpretation, we connect satisfaction under interpretation with validity in the classical propositional logic using independence of values of atomic formulas in different worlds of Kripke models. Every prefixed atomic signed formulas is considered as a classical propositional signed variable where the picture of the prefix becomes an index. For example, instead of $k\ T\ p$ ($k\ F\ p$) we use $T\ p_{I(k)}$ ($Fp_{I(k)}$). In that way the sets of all prefixed atomic signed formulas from finished branches become sets of classical atomic signed formulas called dual clauses. A dual clause is satisfied under a classical propositional interpretation J if every prefixed atomic signed formulas from the clause is satisfied under the interpretation J. A set of dual clauses is satisfied under a classical propositional interpretation J if at least one clause is satisfied under the interpretation J. A set of dual clauses is valid if it is satisfied under every classical propositional interpretation. The following rule is used to resolve dual clauses [3]:

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

If $\emptyset$ denotes the empty clause, and $R(S) = S \cup \{C : C$ is the resolvent of the two clauses from $S\}$, $R_0(S) = S$, $R_i(S) = R(R_{i-1}(S))$, and $R^*(S) = \cup\{R_i(S) : i \geq 0\}$, the following theorem holds:

Theorem 5. *A set S of dual clauses is valid iff $\emptyset \in R^*(S)$.*

A tree t containing $0\ T\ A$ in the root and constructed using the above rules is a proof if the empty clause belongs to the set $R^*(Cl(t))$, where $Cl(t)$ denotes the set of dual clauses that correspond to finished branches of t. From Theorem 4 and 5 the theorem completeness for the S4-dual tableau procedure is obtained:

Theorem 6. *A modal formula A is S4-valid iff it has a finite proof in the S4-dual tableau system.*

For some formulas (e.g. $\Diamond\Box p$) the S4-dual tableau construction, as described above, never terminates. Thus, for the mentioned formula (which is not S4-valid) any theorem prover which is a straightforward implementation of the S4-dual tableau system does not give the answer. There is a modification of the dual tableau construction rules [1, 6] which guarantees that the procedure is finite, and consequently a decision procedure. It is based on the finding of a periodic behavior in a chain of prefixes and sets of signed formulas associated to them. However, the modification is time-expensive, and it is not implemented in the prover we discuss here [5, 7].

2.2. Dual tableau for the full first-order logic

The first-order classical language L_{FO} consists of classical logical operators and quantifiers ($\neg$, $\wedge$, $\vee$, $\rightarrow$, $\forall$, and $\exists$), a set of variables and auxiliary symbols ",", "(" and ")"), and sets of relation and functional symbols. In addition to the α and β formulas, there are two additional types of signed formulas corresponding to quantified formulas that are given in Figure 4, where $A(t/x)$ denotes the formula obtained from A by replacing every free occurrence of the variable x in A by a term t which is free for x in A.

γ	$\gamma(t)$
$T\ (\forall x)A(x)$	$T\ A(t/x)$
$F\ (\exists x)A(x)$	$F\ A(t/x)$

δ	$\delta(t)$
$F\ (\forall x)A(x)$	$F\ A(t/x)$
$T\ (\exists x)A(x)$	$T\ A(t/x)$

Figure 4: γ- and δ-formulas

During the first-order dual tableau construction some new symbols are introduced. They are so-called dummy-variables (or dummies) and Skolem function symbols. We use X, X_1, X_2, ..., and f, f_1, f_1, ..., to denote dummies and Skolem symbols respectively. The first-order dual tableau for a closed formula A is constructed using the following rules (γ- and δ-rules are given in Figure 5):

1. the formula $T\ A$ is placed in the tableau's root,
2. depending on the type of formula in a node, one of the following rules should be applied:
   (a) if the node contains an α-formula, the branch where the node is located extends with nodes containing the subformulas α_1 and α_2,
   (b) if the node contains a β-formula, the branch where the node is located branches with nodes containing the subformulas β_1 and β_2,
   (c) if the node contains a δ-formula, the branch where the node is located extends with a node containing the $\delta_0(X)$ formula, where X is a new dummy, and
   (d) if the node contains a γ-formula, the branch where the node is located extends with a node containing the $\gamma_0(f(X_1, \ldots, X_n))$ formula, where f is a new Skolem function symbol, and X_1, ..., X_n are dummies whose scopes cover the reduced subformula (the scope of a dummy X is the scope of the quantifier whose reduction introduces X).

Let $\mathcal{T}$ be a dual tableau, and Dvar($\mathcal{T}$) the set of all dummy variables from $\mathcal{T}$. The Herbrand universe H($\mathcal{T}$) contains all well-formed ground terms over the symbols that appear in $\mathcal{T}$, i.e., the terms from H($\mathcal{T}$) do not contain dummies

$$\begin{array}{ccc} \delta & \qquad & \gamma \\ | & & | \\ \delta_0(X) & & \gamma_0(f(X_1, \ldots, X_n)) \end{array}$$

Figure 5. γ- and δ-rules

or variables. If $\mathrm{H}(\mathcal{T})$ contains no constant, a constant λ is added to it. A replacement ω is a mapping from $\mathrm{Dvar}(\mathcal{T})$ to $\mathrm{H}(\mathcal{T})$. This definition is extended in an obvious way, so that every replacement maps formulas to formulas. For example $\omega(T\ A \vee B) = \omega(T\ A) \vee \omega(T\ B)$. Let X be a signed formula which appears in a tableau $\mathcal{T}$. Let I be a classical first order interpretation. We say that $I \models X$ if there is a replacement ω, such that $I \models \omega(X)$. A finite set of formulas is satisfied under an interpretation I if the conjunction of all formulas from the set is satisfied under than I. Similar to Theorem 4 we have:

Theorem 7. *Let $\mathcal{T}$ be a first-order dual tableau whose root contains a formula $T\ A$. The formula A is valid iff for each interpretation I there is at least one branch of the tableau whose set of all atomic signed formulas is satisfied under the interpretation I.*

Again, let a dual clause be a set of all signed atomic formulas from a finished tableau branch, $\mathcal{T}$ be a finished dual tableau, and $Cl(\mathcal{T})$ be the corresponding set of dual clauses. To examine validity of a set $Cl(\mathcal{T})$ we use the following dual resolution rule

> if S_1 and S_2 are dual clauses, $L_1 \subset S_1$, and $L_2 \subset S_2$, and μ is the most general unifier such that $\mu(L_1) = \neg\mu(L_2) = \{L\}$, where L is a signed atomic formula, a resolvent of S_1 and S_2 is
>
> $$R(S_1, S_2, L) = \mu((S_1 \setminus L_1) \cup (S_2 \setminus L_2))$$

and have the following statements:

Theorem 8. *A set S of dual clauses is valid iff $\emptyset \in R^*(S)$.*

Theorem 9. *A formula A is valid iff it has a finite proof in the first-order dual tableau system.*

3. The translation procedure

It is already known that the mentioned propositional modal logics can be embedded in the classical first-order logic by the translation τ (we use the following notation: let $x \in \{x_1, x_2\}$; if $x = x_1$, then $x' = x_2$, and vice versa) [2]:

- $\tau(p, x) = P(x)$, for every propositional variable $p \in \Phi$,

- $\tau(\neg A, x) = \neg\tau(A, x)$,
- $\tau(A \wedge B, x) = \tau(A, x) \wedge \tau(B, x)$,
- $\tau(A \vee B, x) = \tau(A, x) \vee \tau(B, x)$,
- $\tau(A \rightarrow B, x) = \tau(A, x) \rightarrow \tau(B, x)$,
- $\tau(\Box A, x) = (\forall x')(R(x, x') \rightarrow \tau(A, x'))$, and
- $\tau(\Diamond A, x) = (\exists x')(R(x, x') \wedge \tau(A, x'))$.

The translation of a modal formula A describes the corresponding conditions on Kripke models that must be satisfied, if A is valid. We use $Des(L)$ to denote a formula which represents the conditions on the accessibility relation for L. For example, $Des(\mathrm{S4}) = (\forall x_1)R(x_1, x_1) \wedge (\forall x_1)(\forall x_2)(\forall x_3)(R(x_1, x_2) \wedge R(x_2, x_3) \rightarrow R(x_1, x_3))$. The following theorem holds:

Theorem 10. *Let L be one of the mentioned modal logics. A propositional modal formula A is L-valid if and only if* $Des(L) \rightarrow (\forall x_1)\tau(A, x_1)$ *is a valid first-order formula.*

Hence, instead of working on the original modal formulas, we can translate these formulas so that classical proof systems can be used. Note that in a translated formula propositional variables become unary relational symbols, and that a new binary relation symbol R appears. The symbol R represents the accessibility relation. Table 1 contains some results obtained by the implementations [7] and [8] of the dual tableau procedure for the modal logic S4 and the first-order dual tableau procedure which is applied on the translated formulas. Both provers have been compiled as 32-bit applications using the Visual C compiler and run on a PC compatible computer with an Intel 133MHz Pentium processor, 32MB RAM and Windows 98SE. The signs (r) and ∞ denote that the corresponding formula is not valid, and that there is no enough memory to finish a proof, respectively. The time unit is 10^{-3}sec.

Figure 6 contains the modal dual tableau for the formula $\Box P \rightarrow P$ and the first-order dual tableau for the corresponding translation $Des(L) \rightarrow (\forall x_1)((\forall x_2)(R(x_1, x_2) \rightarrow P(x_2)) \rightarrow P(x_1))$ of the formula. The letters α, β, γ, δ, ν and π denote which of the reduction rules is applied. It is obvious that, due to the appearance of the $Des(L)$ subformula and the first-order nature of the right-hand part of the translated formula, the second tableau is considerably greater than the first one. There are also more clauses in the second approach and the corresponding proof is almost 5 times longer than the first one. Results are similar in the last two examples given in Table 1. However, note that the execution times in these examples are not very long, i.e. they are less than 3sec.

Modal formula	time	First order translation	time
$\Box P \to P$	60	$Des(L) \to (\forall x_1)((\forall x_2)(R(x_1,x_2) \to P(x_2)) \to P(x_1))$	280
$\Box(\Box P \to P) \to \Box P$	∞	$Des(L) \to (\forall x_1)((\forall x_2)(R(x_1,x_2) \to ((\forall x_1)(R(x_2,x_1) \to P(x_1)) \to P(x_2))) \to (\forall x_2)(R(x_1,x_2) \to P(x_2)))$	(r) 1650
$\Box\Diamond P \to \Box\Diamond\Box\Diamond P$	∞	$Des(L) \to (\forall x_1)((\forall x_2)(R(x_1,x_2) \to (\exists x_1)(R(x_2,x_1) \wedge P(x_1))) \to (\forall x_2)(R(x_1,x_2) \to (\exists x_1)(R(x_2,x_1) \wedge (\forall x_2)(R(x_1,x_2) \to (\exists x_1)(R(x_2,x_1) \wedge P(x_1))))))$	2910
$\Diamond\Box P$	∞	$Des(L) \to (\forall x_1)((\exists x_2)(R(x_1,x_2) \wedge (\forall x_1)(R(x_2,x_1) \to P(x_1))))$	(r) 440
$P \to \Diamond P$	110	$Des(L) \to (\forall x_1)(P(x_1) \to (\exists x_2)(R(x_1,x_2) \wedge P(x_2)))$	270
$\Box(P \vee Q) \to (\Box P \vee \Box Q)$	(r) 220	$Des(L) \to (\forall x_1)((\forall x_2)(R(x_1,x_2) \to (P(x_1) \vee Q(x_2))) \to ((\forall x_2)(R(x_1,x_2) \to P(x_2)) \to (\forall x_2)(R(x_1,x_2) \to Q(x_2))))$	(r) 1380

Table 1: Some results of the direct and indirect modal theorem proving

On the other hand, there are cases where the indirect approach is more suitable than the direct one. They are illustrated by the examples 2 – 4. As we have already mentioned, the periodicity test is not implemented in the modal prover. Thus, if there is a $\Diamond\Box$ combination of operators in the considered formula the indirect approach gives better results.

$T\ \Box P \to P$ — / \ β: $F\ \Box P$ and $T\ A$; $F\ \Box P$ — | π — $F\ P$

$T\ Des(L) \to (\forall x_1)((\forall x_2)(R(x_1,x_2) \to P(x_2)) \to P(x_1))$

/ \ β

$F\ Des(L)$ and $T\ (\forall x_1)((\forall x_2)(R(x_1,x_2) \to P(x_2)) \to P(x_1))$

| γ

$T\ (\forall x_2)(R(f(),x_2) \to P(x_2)) \to P(f())$

/ \ β

$F\ (\forall x_2)(R(f(),x_2) \to P(x_2))$ and $T\ P(f())$

| δ

$F\ R(f(),X) \to P(X)$

| α

$T\ R(f(),X)$

|

$F\ P(X)$

Fig. 6. Two tableaus for $\Box P \to P$ and its first-order translation

4. Conclusion

We have presented two approaches to modal theorem proving. In the direct one, the modal dual tableau and the resolution procedure are directly implemented in the source code of the prover. On the other hand, in the indirect approach, the prover corresponds to the classical first-order logic, but a translation procedure from the modal to the classical framework is included. There is no doubt that the direct prover is more efficient because the semantics of the accessibility relation is directly implemented in its code, while the indirect prover handles it at the syntactical level of formulas. However, the second approach is more flexible. For example, we can easily change the type of the accessibility relation by giving another $Des(L)$ part of the translation, while in the direct approach we have to modify the source code of the prover. There are modal logics where many $\Box$-operators are considered. For example, in logics of knowledge, $\Box_i$ is read as 'agent i knows'. In the indirect approach we can handle such a situation. The only thing we have to do is to change the translation procedure so that a different relation symbol R_i corresponds to every accessibility relation and operator $\Box_i$.

There is another place in the direct modal theorem proving which is convenient for the application of dummies and Scolemization. Modal operators can be interpreted as quantifiers over possible worlds. We hope that dummies and Scolemization can reduce the number of new worlds in the application of the modal ν and π reduction rules. Also, some strategies for the classical resolution that work well under the assumption of the considered accessibility relation can be developed to give more efficient proofs.

While we have been finishing this paper, we have found the paper [9] where the similar ideas are considered. The paper is, however, more abstract, and no information about results of any theorem prover is presented. On the other hand, an extension of the above translation is given so that first-order modal logics with constant or flexible domains, and rigid or non-rigid functional symbols can be examined.

References

[1] Fitting, M., Proof Methods for Modal and Intuitionistic Logic, D.Reidel Publishing Co., Dordecht, 1983.

[2] Fitting, M., Basic modal logics, in: Handbook of logic in artificial intelligence and logic programming, eds.: D. Gabbay, C. J. Hogger, J. A. Robinson, J. Sickmann. Clarendon Press, 365 – 448, 1993.

[3] Kapetanović, M., Krapež, A., A proof procedure for the first order logic, Publications de L'Institut Mathematique, vol. 59, 3-5, 1989.

[4] Kripke, S., Sematical analysis of modal logics I, normal propositional calculi, Zeitschrift für mathematische Logik und Grundlagen der Mathematik, 9, 67-96, 1963.

[5] Ognjanović, Z., Theorem prover in the modal calculus S4 based on the dual-tableaux method (in Serbian), M.Sc. Thesis, Mathematical Faculty, Belgrade, 1993.

[6] Ognjanović, Z., A tableau-like proof procedure for normal modal logics, Theoretical computer science, 129, 167 – 186, 1994.

[7] Ognjanović, Z., Urošević, D., Petrović, T., Ristović, Z., Tableaux Related Method for Modal Theorem Proving, *Proc. of the 1st Workshop on Theorem Proving with Analytic Tableaux and Related Methods*, Lautenbach, Germany, 1992.

[8] Ognjanović, Z., Urošević, D., Božin, V., A tableaux related method for the full first order logic, Lira '92, Proceedings of the 6th Conference on logic and computer science, Novi Sad, 144 – 145, 1992.

[9] Ohlbach, H. J., Semantic-based translation methods for modal logics, Journal of logic and computation, vol. 1, no. 5, 691 – 746, 1991.

[10] Smullyan, R., First-Order Logic, Sprin[illegible]