Free-variable Tableaux for Propositional Modal Logics

Bernhard Beckert and Rajeev Goré

Abstract. Free-variable semantic tableaux are a well-established technique for first-order theorem proving where free variables act as a meta-linguistic device for tracking the *eigenvariables* used during proof search. We present the theoretical foundations to extend this technique to propositional modal logics, including non-trivial rigorous proofs of soundness and completeness, and also present various techniques that improve the efficiency of the basic naive method for such tableaux.

Keywords: automated deduction, modal logics, modal theorem proving, free-variable tableaux

1. Introduction

Free-variable semantic tableaux are a well-established technique for first-order theorem proving where free variables act as a meta-linguistic device for tracking the *eigenvariables* used during proof search [27, 13]. By allowing the choice of these free-variables to be deferred until more information becomes available, free variables reduce the search space and reduce the non-determinism inherent in automated proof search. Free variables are the most important refinement of traditional tableaux for classical logic from the automated theorem proving perspective. It is therefore natural to investigate whether free-variables can be used in automated theorem proving for non-classical logics.

Kanger's meta-linguistic indices for non-classical logics [20] have already been generalised by Gabbay into Labelled Deductive Systems [15], and recently, Massacci [22] and Russo [28] have shown the utility of using *ground* labels for obtaining *modular* modal tableaux and natural deduction systems (respectively); see [16] for an introduction to labelled modal tableaux.

Here we present a rigorous account of free-variable tableaux for propositional modal logics. We show how to use this theory to obtain modular proof systems based upon free-variable tableaux for all 15 basic modal logics. This paper is a full version of [2]. It does not include further work on decidability issues which we have reported in [3, 4] as these aspects have not been fully worked out yet.

Our object language uses *labelled formulae* like $\sigma : A$, where σ is a label and A is a formula, with intuitive reading "the possible world σ satisfies the formula A"; see [12, 24, 16] for details. Thus, $1 : \Box p$ says that the possible world 1 satisfies the formula $\Box p$. Our box-rule then reduces the formula $1 : \Box p$ to the labelled formula $1.(x) : p$ which contains the *universal* variable x in its label and has an intuitive reading "the possible world $1.(x)$ satisfies the formula p". Since different instantiations of x give different labels, the labelled formula $1.(x) : p$ effectively says that "all successors of the possible world 1 satisfy p", thereby capturing the usual Kripke semantics for $\Box p$ (almost) exactly. But the possible world 1 may have *no* successors; so we enclose the variable in parentheses and read $\sigma{:}A$ as “for all instantiations of the variables in σ, if the world corresponding to that instantiation of σ exists then the world satisfies the formula A”.

The use of variables as place-markers for *eigenvariables* has been used by various authors in the past where it has also been used in conjunction with specialist unification algorithms to close branches. The earliest such work is probably that of Wallen [29], but Wallen catered for only a very few simple modal logics. Similar approaches using labels containing variables have been explored by Governatori [17], D’Agostino et al. [10] and Pitt and Cunningham [26]. D’Agostino et al. relate the labels to modal algebras, instead of to first-order logic as we do. Both Pitt and Cunningham and Governatori use string unification over labels to detect complementary formulae, whereas we just use matching. Consequently, our variables are of a simpler kind: they capture all *immediate* children of a possible world (in a rooted tree model), but do not capture *all R-successors*; see [22, 16]. Note, however, that extensions of our calculi using string unification are perfectly feasible. Indeed, such work is currently being persued by Bonnette [9].

The following techniques, in particular, are crucial:

Free variables: Applying the traditional ground box-rule requires guessing the correct *eigenvariables*. Using (free) variables in labels as “wildcards” that get instantiated “on demand” during branch closure allows more intelligent choices of these *eigenvariables*. To preserve soundness for worlds with no R-successors, variable positions in labels must be conditional.

Universal variables: Under certain conditions, a variable x introduced by a formula like $\Box A$ is “universal” in that an instantiation of x on one branch need not affect the value of x on other branches, thereby localising the effects of a variable instantiation to one branch. The technique entails creating and instantiating local duplicates of labelled formulae instead of the originals.

Finite diamond-rule: Applying the diamond-rule to $\Diamond A$ usually creates a new label. By using (a Gödelisation of) the formula A itself as the label instead, we guarantee that only a finite number of different labels (of a certain length) are used in the proof. In particular, different (identically labelled) occurrences of $\Diamond A$ generate the same unique label.

The paper is structured as follows: In Sections 2 and 3 we introduce the syntax and semantics of labelled modal tableaux. In Section 4 we introduce our calculus and present an example; we prove its soundness and completeness in Sections 5 and 6, respectively. In Section 7 we present our conclusions and discuss future work.

2. Syntax

The formulae of modal logics are built in the usual way from a denumerable non-empty set $\mathcal{P}$ of primitive propositions, the classical connectives $\wedge$ (conjunction), $\vee$ (disjunction), $\neg$ (negation), $\rightarrow$ (implication), and the non-classical unary modal connectives $\Box$ ("box") and $\Diamond$ ("diamond").

To reduce the number of tableau rules and the number of case distinctions in proofs, we restrict all considerations to implication-free formulae in negated normal form (NNF); thus negation signs appear in front of primitive propositions only. Using NNF formulae is no real restriction since every formula can be transformed into an equivalent NNF formula in linear time.

Labels are built from natural numbers and variables, with variables intended to capture the similarities between the $\forall$ quantifier of first-order logic and the $\Box$ modality of propositional modal logic. However, whereas first-order logic forbids an empty domain, the $\Box$ modality tolerates possible worlds with no successors.[1] To capture this (new) behaviour, variable positions in labels are made "conditional" on the existence of an appropriate successor by enclosing these conditional positions in parentheses.

DEFINITION 2.1. Let *Vars* be a set of variables. Then, m is a *label* for $m \in \mathbb{N}$; and if σ is a label, then so are $\sigma.m$ and $\sigma.(l)$ for $m \in \mathbb{N}$ and $l \in Vars \cup \mathbb{N}$. The *length* $|\sigma|$ of a label σ is the number of dots it contains plus one. The constituents of a label σ are called *positions* in σ and terms like "the 1st position" or "the n-th position" are defined in the obvious way. A position is *conditional* if it is of the form (l), and a label is conditional if it contains a conditional position. By $ipr(\sigma)$ we mean the set of all non-empty *initial prefixes* of a label σ, excluding σ itself. A label is *ground* if it consists of (possibly conditional) members of $\mathbb{N}$ only. Let $\mathcal{L}$ be the set of all ground labels.

When dealing with ground labels, we often do not differentiate between the labels $\sigma.m$ and $\sigma.(m)$, and we use $\sigma.[m]$ to denote that the label may be of either form. Note that $\sigma.x$ (parentheses around x omitted) is not a label for $x \in Vars$: the parentheses mark the positions that contain variables, or that used to contain variables before a substitution was applied.

DEFINITION 2.2. A set Γ of labels is *strongly generated* if:

1. there is some (root) label $\rho \in \Gamma$ with $\rho \in ipr(\sigma)$ for all $\sigma \in \Gamma \setminus \{\rho\}$; and

2. $\sigma \in \Gamma$ implies $\tau \in \Gamma$ for all $\tau \in ipr(\sigma)$.

[1] To that extent, modal logics are similar to free logic, i.e., first-order logic where the domains of models may be empty [8].

Since we deal with mono-modal logics with semantics in terms of rooted frames (see Section 3), we always assume that our labels form a strongly generated set with root $\rho = 1$.

DEFINITION 2.3. A *labelled tableau formula* (or just tableau formula) ϕ is a structure of the form $X{:}\Delta{:}\sigma{:}A$, where X is a subset of $\mathit{Vars} \cup \mathbb{N}$, Δ is a set of labels, σ is a label, and A is a formula in NNF. If the set Δ is empty, we use $X{:}\sigma{:}A$ as an abbreviation for $X{:}\emptyset{:}\sigma{:}A$. A tableau formula $X{:}\Delta{:}\sigma{:}A$ is *ground* if σ and all labels in Δ are ground. If $\mathcal{F}$ is a set of labelled tableau formulae, then $\mathit{lab}(\mathcal{F})$ is the set $\{\sigma \mid X{:}\Delta{:}\sigma{:}A \in \mathcal{F}\}$.

The intuitions behind the different parts of our "tableau formulae" are as follows: The fourth part A is just a traditional modal formula. The third part σ is a label, possibly containing variables introduced by the reduction of $\Box$ modalities. If the label σ is ground, then it corresponds to a particular path in the intended rooted tree model; for example, the ground label 1.1.1 typically represents the leftmost child of the leftmost child of the root 1. If σ contains variables, then it represents all the different paths (successors) that can be obtained by different instantiations of the variables, thereby capturing the semantics of the $\Box$ modalities that introduced them. Our rule for splitting disjunctions allows us to retain these variables in the labels of the two disjuncts, but because $\Box$ does not distribute over $\vee$, such variables then lose their "universal" force, meaning that these "free" variables can be instantiated only *once* in a tableau proof. We use the first component X to record the variables in the tableau formula ϕ that are "universal", meaning that ϕ can be used multiply in the same proof with different instantiations for these variables. The free variables in ϕ (that do not appear in X) can be used with only one instantiation since they have been pushed through the scope of an $\vee$ connective. The second part Δ, which can be empty, has a significance only if our calculus is applied to one of the four logics **KB**, **K5**, **KB4**, and **K45** (that are non-serial, but are symmetric or euclidean). It is empty for the other logics. The intuition of Δ is that the formula A has to be true in the possible world called σ only if the labels in Δ name legitimate worlds in the model under consideration. This feature has to be used, if (a) rule applications may shorten labels, which is the case if the logic is symmetric or euclidean, and (b) the logic is non-serial and, thus, the existence of successor worlds is not guaranteed. The set Δ can contain both universal and free variables, and some of them may appear in σ.

DEFINITION 2.4. Given a tableau formula $\phi = X{:}\Delta{:}\sigma{:}A$, $\mathit{Univ}(\phi) = X$ is the set of *universal variables* of ϕ, while $\mathit{Free}(\phi) = \{x \mid x \text{ appears in } \sigma \text{ or } \Delta, x \notin X\}$ is the set of *free variables* of ϕ. These notions are extended in the obvious way to obtain the sets $\mathit{Free}(\mathcal{T})$ and $\mathit{Univ}(\mathcal{T})$ of free and universal variables of a given tableau $\mathcal{T}$ (see Def. 2.5).

DEFINITION 2.5. A *tableau* is a (finite) binary tree whose nodes are tableau formulae. A *branch* in a tableau $\mathcal{T}$ is a maximal path in $\mathcal{T}$ (where no confusion can arise, we identify a tableau branch with the set of tableau formulae it contains). A branch may be marked as being *closed*. If it is not marked as being closed, it is *open*. A tableau branch is *ground* if every formula on it is ground, and a tableau is ground if all its branches are ground.

Since we deal with propositional modal logics, notions from first-order logic like variables and substitutions are needed only for handling semantic notions like the accessibility relation between worlds. Specifically, whereas substitutions in first-order logic assign terms to variables, here they assign numbers or other variables (denoting possible worlds) to variables.

DEFINITION 2.6. A *substitution* is a (partial) function $\mu : \mathit{Vars} \to \mathbb{N} \cup \mathit{Vars}$. Substitutions are extended to labels and formulae in the obvious way. A substitution is *grounding* if its domain is the (whole) set *Vars* and its range is $\mathbb{N}$; that is, if it maps all variables in *Vars* to natural numbers. The *restriction* of a substitution μ to a set X of variables is denoted by $\mu_{|X}$. The *concatenation* $\mu \circ \nu$ of substitutions μ and ν is defined by $(\mu \circ \nu)(x) = \mu(\nu(x))$ for all variables $x \in \mathit{Vars}$.

Note, that applying the concatenation $\mu \circ \nu$ has the same effect as first applying ν and then applying μ, i.e., $o(\mu \circ \nu) = o\nu\mu$ for all objects o (labels, formulae, etc.).

DEFINITION 2.7. Given a tableau $\mathcal{T}$ containing a tableau formula $X : \Delta : \sigma : A$, a tableau formula $X' : \Delta' : \sigma' : A$ is a $\mathcal{T}$*-renaming* of $X : \Delta : \sigma : A$ if there is a substitution μ such that (a) $X' : \Delta' : \sigma' : A = (X : \Delta : \sigma : A)\mu$, (b) the domain of μ is X (all other variables remain unchanged), and (c) μ replaces the variables in X by distinct variables new to the tableau $\mathcal{T}$.

3. Semantics

In this section we first introduce the Kripke semantics for modal logics, and then extend these semantics to labelled tableau formulae and tableau.

DEFINITION 3.1. A Kripke *frame* is a pair $\langle W, R \rangle$, where W is a non-empty set (of possible worlds) and R is a binary relation on W. A Kripke *model* is a triple $\langle W, R, V \rangle$, where the valuation V is a mapping from primitive propositions to sets of worlds. Thus, $V(p)$ is the set of worlds at which p is "true" under the valuation V. We write wRw' iff $(w, w') \in R$, and we say that world w' is *reachable* from world w, and that w' is a *successor* of w.

DEFINITION 3.2. Given some model $\langle W, R, V \rangle$, and some $w \in W$, we write $w \models p$ iff $w \in V(p)$. This satisfaction relation $\models$ is then extended to more complex formulae as usual. We say that w *satisfies* a formula A iff $w \models A$. A formula A is *satisfied*

Table I. Basic axioms and the corresponding property of the reachability relation.

Name	Axiom	Property
(K)	$\Box(A \to B) \to (\Box A \to \Box B)$	—
(T)	$\Box A \to A$	reflexive
(D)	$\Box A \to \Diamond A$	serial
(4)	$\Box A \to \Box\Box A$	transitive
(5)	$\Diamond A \to \Box\Diamond A$	euclidean[2]
(B)	$A \to \Box\Diamond A$	symmetric

by a model $\langle W,R,V\rangle$ if it is satisfied by some world $w \in W$; it is *valid* in $\langle W,R,V\rangle$, written as $\langle W,R,V\rangle \models A$, iff every world in W satisfies A. A formula A is *valid* in a frame $\langle W,R\rangle$, iff it is valid in every model $\langle W,R,V\rangle$ based on that frame. An axiom A is *valid* in a frame $\langle W,R\rangle$, iff every formula instance of it is valid in $\langle W,R\rangle$.

The first two columns of Table II show the axiomatisations of the 15 basic logics that can be formed from the axioms shown in Table I.

DEFINITION 3.3. Given one of the logics **L** listed in Table II, a frame $\langle W,R\rangle$ is an **L**-*frame* if each axiom of **L** is valid in $\langle W,R\rangle$. A model $\langle W,R,V\rangle$ is an **L**-*model* if $\langle W,R\rangle$ is an **L**-frame. A formula A is **L**-*satisfiable* if there is an **L**-model satisfying A.

The axioms listed in Table I are characterised by the properties of R listed next to them; see [16]. Thus, all **KT**-frames have a reflexive accessibility relation R, and if a frame has a reflexive accessibility relation then it will validate axiom (T). Therefore, we associate these properties with logics as well, and say, for example, that a logic **L** is serial if all **L**-frames have a serial accessibility relation. Some care is needed: for example the axiom (D) is not an axiom of **KT**, but it is valid in all **KT**-frames since it is implied by (T). Consequently the reachability relation R of all **KT**-models *is* serial.

As we shall soon see, ground labels capture a basic reachability relation between the worlds they name, where the world named by $\sigma.[n]$ is reachable from the world named by σ. A set of strongly generated ground labels can be viewed as a tree with root ρ, where $\sigma.[n]$ is an immediate child of σ (hence the name "strongly generated"). We formalise this as follows.

DEFINITION 3.4. Given a logic **L** and a set Γ of strongly generated ground labels with root $\rho = 1$, a label $\tau \in \Gamma$ is **L**-*accessible* from a label $\sigma \in \Gamma$, written as $\sigma \rhd \tau$, if

[2] Relation R is euclidean iff: for all $u,v,w \in R$, uRv and uRw implies vRw.

Table II. Basic logics, axiomatic characterisations, and **L**-accessibility $\triangleright$.

Logic	Axioms	$\sigma \triangleright \tau$	Logic	Axioms	$\sigma \triangleright \tau$
K	(K)	$\tau = \sigma.[n]$	**KT**	(KT)	$\tau = \sigma.[n]$ or $\tau = \sigma$
KB	(KB)	$\tau = \sigma.[n]$ or $\sigma = \tau.[m]$	**K4**	(K4)	$\tau = \sigma.\theta$
K5	(K5)	$\tau = \sigma.[n]$, or $\lvert\sigma\rvert \geq 2$, $\lvert\tau\rvert \geq 2$	**K45**	(K45)	$\tau = \sigma.\theta$, or $\lvert\sigma\rvert \geq 2$, $\lvert\tau\rvert \geq 2$
KD	(KD)	**K**-condition, or σ is a **K**-deadend and $\sigma = \tau$	**KDB**	(KDB)	**KB**-condition, or $\lvert\Gamma\rvert = 1$ and $\sigma = \tau = 1$
KD4	(KD4)	**K4**-condition, or σ is a **K**-deadend and $\sigma = \tau$	**KD5**	(KD5)	**K5**-condition, or $\lvert\Gamma\rvert = 1$ and $\sigma = \tau = 1$
KD45	(KD45)	**K45**-cond., or $\lvert\Gamma\rvert = 1$, $\sigma = \tau = 1$	**KB4**	(KB4)	$\lvert\Gamma\rvert \geq 2$
B	(KTB)	$\tau = \sigma$, or $\tau = \sigma.[n]$, or $\sigma = \tau.[m]$	**S4**	(KT4)	$\tau = \sigma.\theta$ or $\tau = \sigma$
S5	(KT5)	for all σ, τ			

the conditions set out in Table II for **L** are satisfied. A label $\sigma \in \Gamma$ is an **L**-*deadend*, if no $\tau \in \Gamma$ is **L**-accessible from σ.

The following lemma shows that the **L**-accessibility relation $\triangleright$ on labels captures the reachability relation R of **L**-frames; see [16] for a proof. In particular, $\triangleright$ has the properties like reflexivity, transitivity, etc. that are appropriate for the axioms of **L** (see Table I).

LEMMA 3.5. *If Γ is a strongly generated set of ground labels with root $\rho = 1$, then $\langle \Gamma, \triangleright \rangle$ is an* **L**-*frame.*

The traditional notion of satisfaction relates a world in a model with a formula or a set of formulae. When formulae are annotated with ground labels, the notion of satisfaction must be extended by a further "interpretation function" that maps ground labels to worlds; see [13, 16]. If the labels are allowed to contain free variables, and in particular, universal variables, then the notion of satisfaction must also allow for all possible instantiations of the universal variables, thus catering for many different "interpretation functions". As usual, we define "satisfiability" so that our tableau expansion rules preserve this notion, and such that a "closed tableau" is not satisfiable.

We proceed incrementally by defining satisfiability for: ground labels; ground tableau formulae; non-ground tableau formulae; and whole tableaux. But first we enrich models by the "interpretation function" that maps labels to worlds. Note that such interpretations give a meaning to *all* ground labels, not just to those that appear in a particular tableau. A label that does not correspond to a world in the model is mapped to the special symbol $\perp$.

DEFINITION 3.6. An **L**-interpretation is a pair $\langle \mathbf{M}, \mathbf{I} \rangle$, where $\mathbf{M} = \langle W, R, V \rangle$, is a Kripke **L**-model and **I** is a function $\mathbf{I} : \mathcal{L} \to W \cup \{\perp\}$ interpreting ground labels such that:

(i) $\mathbf{I}(1) \in W$

(ii) $\mathbf{I}(\sigma.(n)) = \mathbf{I}(\sigma.n)$ for all $\sigma.n$ and $\sigma.(n)$ in $\mathcal{L}$

(iii) for all $\sigma \in \mathcal{L}$, if $\mathbf{I}(\tau) = \perp$ for some $\tau \in ipr(\sigma)$ then $\mathbf{I}(\sigma) = \perp$

(iv) if $\sigma \triangleright \tau$, $\mathbf{I}(\sigma) \in W$, and $\mathbf{I}(\tau) \in W$, then $\mathbf{I}(\sigma) R \mathbf{I}(\tau)$.

DEFINITION 3.7. An **L**-interpretation $\langle \mathbf{M}, \mathbf{I} \rangle$, where $\mathbf{M} = \langle W, R, V \rangle$, *satisfies* a ground label σ, if for all labels $\tau.n \in ipr(\sigma) \cup \{\sigma\}$ (that end in an unconditional label position): $\mathbf{I}(\tau) \in W$ implies $\mathbf{I}(\tau.n) \in W$. The **L**-interpretation $\langle \mathbf{M}, \mathbf{I} \rangle$ *satisfies* a ground tableau formula $X : \Delta : \sigma : A$, if

(a) $\mathbf{I}(\sigma) = \perp$, or $\mathbf{I}(\tau) = \perp$ for some $\tau \in \Delta$, or $\mathbf{I}(\sigma) \models A$; and

(b) if $\mathbf{I}(\tau) \in W$ for all $\tau \in \Delta$, then $\langle \mathbf{M}, \mathbf{I} \rangle$ satisfies σ.[3]

Thus, a tableau formula is satisfied by default if its label σ is undefined (that is, if $\mathbf{I}(\sigma) = \perp$) or if one of the labels in Δ is undefined. But because we deal only with strongly generated sets of labels with root 1, the twin requirements that every **L**-interpretation $\langle \mathbf{M}, \mathbf{I} \rangle$ define the label 1, and condition (b) in the above definition force the interpretation function **I** to "define" as many members of $ipr(\sigma)$ as is possible. However, for a conditional ground label of the form $\tau.(n)$, where n is parenthesised, it is perfectly acceptable to have $\mathbf{I}(\tau.(n)) = \perp$ even if $\mathbf{I}(\tau) \in W$.

EXAMPLE 3.8. If $\langle \mathbf{M}, \mathbf{I} \rangle$ satisfies $\sigma = 1.1.1$, then $\mathbf{I}(1)$, $\mathbf{I}(1.1)$, and $\mathbf{I}(1.1.1)$ must be defined. If $\sigma = 1.(1).1$, then $\mathbf{I}(1.(1))$ need not be defined; but if it is, then $\mathbf{I}(1.(1).1)$ must be defined.

The domain of every interpretation function **I** is the set of all *ground* labels $\mathcal{L}$, but our tableaux contain labels with variables. We therefore introduce a definition of satisfiability for non-ground tableau formulae capturing our intuitions that a label $\sigma.(x)$ stands for *all* possible successors of the label σ, and taking into account the special nature of universal variables.

[3] In particular, $\langle \mathbf{M}, \mathbf{I} \rangle$ must satisfy σ if $\Delta = \emptyset$, which is the most frequent case.

DEFINITION 3.9. Given an **L**-interpretation $\langle \mathbf{M}, \mathbf{I} \rangle$ and a grounding substitution μ, a (non-ground) tableau formula $\phi = X : \Delta : \sigma : A$ is *satisfied* by $\langle \mathbf{M}, \mathbf{I}, \mu \rangle$, written as $\langle \mathbf{M}, \mathbf{I}, \mu \rangle \models \phi$, if for all grounding substitutions λ, the ground formula $\phi\lambda_{|X}\mu$ is satisfied by $\langle \mathbf{M}, \mathbf{I} \rangle$ (Def. 3.7). A set $\mathcal{F}$ of tableau formulae is satisfied by $\langle \mathbf{M}, \mathbf{I}, \mu \rangle$, if every member of $\mathcal{F}$ is simultaneously satisfied by $\langle \mathbf{M}, \mathbf{I}, \mu \rangle$.

In the above definition, a ground formula $\phi\lambda_{|X}\mu$ is constructed from ϕ in two steps, such that the definition of satisfiability for ground formulae can be applied. To cater for the differences between the free variables and universal variables, we use two substitutions: a fixed substitution μ and an arbitrary substitution λ. The first step, applying $\lambda_{|X}$ to ϕ instantiates the universal variables $x \in X$. The second step, applying μ to $\phi\lambda_{|X}$, instantiates the free variables. Therefore, the instantiation of universal variables $x \in X$ is given by the arbitrary substitution λ, and the instantiation of free variables $x \notin X$ is given by the fixed substitution μ. Quantifying over all λ captures the universal nature of the members of X.

Note, that in the following definition of satisfiable tableaux, there has to be a single satisfying **L**-interpretation for *all* grounding substitutions μ.

DEFINITION 3.10. A tableau $\mathcal{T}$ is **L**-*satisfiable* if there is an **L**-interpretation $\langle \mathbf{M}, \mathbf{I} \rangle$ such that for *every* grounding substitution μ there is some *open* branch $\mathcal{B}$ in $\mathcal{T}$ with $\langle \mathbf{M}, \mathbf{I}, \mu \rangle \models \mathcal{B}$.

4. The Calculus

4.1. OVERVIEW

We now present an overview of our calculus, highlighting its main principles.

Refutation method. Our calculus is a refutation method: to prove that a formula A is a theorem of logic **L**, we first convert its negation $\neg A$ into NNF obtaining a formula B, and then test if B is **L**-unsatisfiable. To do so, we start with the initial tableau whose single node is $\emptyset : \emptyset : 1 : B$ and repeatedly apply the tableau expansion rules, the substitution rule, and the closure rule until a closed tableau results. Since our rules preserve **L**-satisfiability of tableaux, a closed tableau indicates that B is indeed **L**-unsatisfiable, and hence that its negation A is **L**-valid. Since **L**-frames characterise the logic **L** we then know that A is a theorem of logic **L**. Constructing a tableau for $\emptyset : \emptyset : 1 : B$ can be seen as a search for an **L**-model for B. Each branch is a partial definition of a possible **L**-model, and different substitutions give different **L**-models. Our tableau rules extend *one* particular branch using *one* particular formula, thus differing *crucially* from the systematic methods in [12, 16] where a rule extends *all* branches that pass through one particular formula.

Universal variables. In the following we explain the relationship between universal variables in first-order logic (resp. universal quantifiers in first-order logic) and modal logics (resp. box-formulae in modal logics). To emphasise the similarities, we use in this explanation a simplified notation $\sigma : \Box A$ instead of $X : \Delta : \sigma : A$ for tableau formulae.

In first-order tableaux, a free variable x is used as a place marker for an *eigenvariable* whose value is unknown when reducing a formula $\forall x \varphi(x)$ to $\varphi(x)$. In our calculus, a free variable x is used as a place marker for a successor world whose value is unknown when reducing a labelled formula $\sigma : \Box A$ to $\sigma.x : A$. Because a world may have no successor, variable positions in labels in our calculus must be conditional to preserve soundness for non-serial logics, but we ignore this aspect for the moment to simplify the exposition. In each calculus, the free variable x is used so that the actual *value* of x does not have to be guessed at the point where $\forall x \varphi(x)$ or $\sigma : \Box A$ is reduced. Instead, we defer the choice of x until enough information is available to make a choice that *immediately* closes a branch of the respective tableau.

Since the free variable x is a proxy for one instance of x, or one particular successor of σ respectively, x must be instantiated to the same value on all branches. Moreover, one single instantiation of the free variables has to be found that allows us to close all branches of a tableau *simultaneously*, and instantiating a free variable (in the wrong way) to close one branch, can make it impossible to close other branches.

However, both $\forall x \varphi(x)$ and $\sigma : \Box A$ have a universal nature, and we may require additional renamings $\varphi(x_1), \ldots, \varphi(x_n)$ and $\sigma.x_1 : A, \ldots, \sigma.x_n : A$ of the respective reduced formulae to construct a close tableau, corresponding to multiple *different* instances of $\forall x \varphi(x)$ and *multiple different* successors of $\sigma : \Box A$ respectively.

Beckert and Hähnle [5] noticed that under certain conditions, the free variable x can be instantiated in one way to close one branch, but this binding can be undone, and x can be instantiated in a different way to close another branch, without losing soundness. Such a variable x is said to be "universal" since it allows the *single* formula $\varphi(x)$ or $\sigma.x : A$ respectively to stand in for the *multiple* renamings $\varphi(x_1), \ldots, \varphi(x_n)$ and $\sigma.x_1 : A, \ldots, \sigma.x_n : A$ mentioned above. Indeed, if the number of required renamings n is large, then using a variable in this "universal" manner can shorten the tableau branches considerably. The example below illustrates this point for our modal tableaux.

EXAMPLE 4.1. Suppose that we are given a tableau branch containing the tableau formulae $1 : \Diamond\neg p \vee \Diamond\neg q$ and $1 : \Box(p \wedge q)$.

In Figure 1(a) we reduce the second formula to obtain $1.x_1 : p \wedge q$, creating the free variable x_1. Applying the conjunctive rule to this formula gives $1.x_1 : p$ and $1.x_1 : q$. Next we apply the disjunctive rule to the root and split the tableau into two branches, the first containing $1 : \Diamond\neg p$ and the second containing $1 : \Diamond\neg q$. We apply the diamond-rule to obtain $1.1 : \neg p$ on the first branch where 1.1 is a

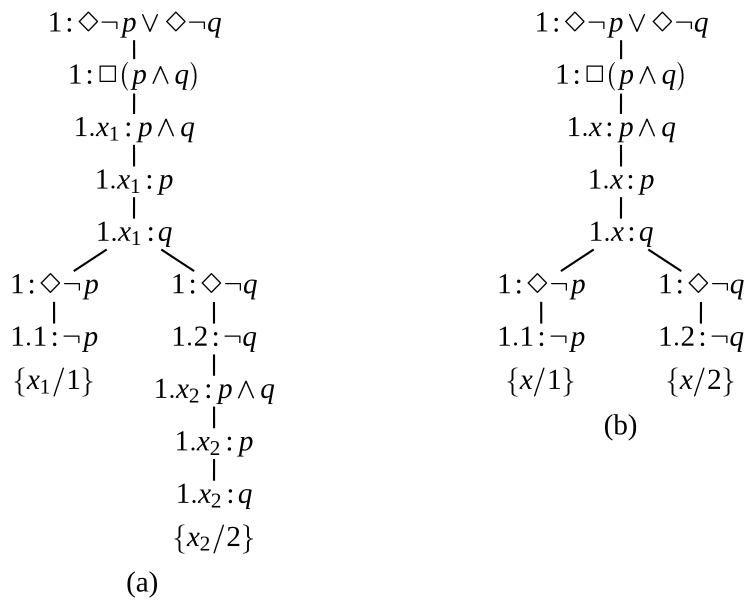

Figure 1. Tableau proof (a) without and (b) with universal variables (see Example 4.1).

label new to the tableau. We can now close the left branch by setting $x_1 := 1$. All occurrences of x_1 are now bound to 1.

To continue, we apply the diamond-rule to $1 : \Diamond\neg q$ on the second branch to obtain $1.2 : \neg q$ where the label 1.2 is new to the tableau.

The only potential closure on this second branch is between $1.2 : \neg q$ and $1.x_1 : q$ but this closure is not possible since x_1 is bound to 1 so that the latter formula is actually $1.1 : q$, and this does not contradict $1.2 : \neg q$. We therefore have to apply the box-rule once again to $1 : \Box(p \wedge q)$ to obtain $1.x_2 : p \wedge q$ on the second branch thereby creating a second free variable x_2. Applying the conjunctive rule to this formula allows us to close the second branch by putting $x_2 := 2$ as shown.

In Figure 1(b) we proceed in exactly the same manner until we close the first branch, except that we create the free variable x rather than x_1. But before proceeding to process the second branch, we undo the binding for x. Consequently we can close the second branch much sooner by putting $x := 2$, without generating the extra renaming $1.x_2 : p \wedge q$ required in Figure 1(a). Thus the variable x is instantiated in multiple ways.

How to detect such "universal" variables and use them only in a sound manner? Here is one way. Every variable x is introduced into a label by the reduction of a box-formula like $\Box A$. At some later stage of the tableau construction, in some particular formula $\varphi(x)$ on some particular tableau branch $\mathcal{B}$, the variable x will be "universal" if a renaming $\varphi' = \varphi\{x := x'\}$ of φ could be added to $\mathcal{B}$ without generating additional branches. That is, the modified tableau would be no more difficult to close than the original. One way to generate the renaming is to repeat the rule applications that lead to the generation of φ, starting from the box-rule application that created x. Once renaming φ' is present on $\mathcal{B}$, the variable x never has to be instantiated to close $\mathcal{B}$ because φ' could be used instead of φ, thus instantiating x' instead of x. However, if x occurs on two separate branches in the tableau then repeating these rule applications would generate at least one additional branch, making the new tableau harder to close than the original one.

All variables are obviously universal when they are created. So how can a universal variable become non-universal? The crucial criterion is that the steps that generate the copy x' of x must not cause the creation of *additional* branches to the tableau. This simply means that all occurrences of the original x must occur on only one branch. Since the only rule that causes branching is the disjunctive rule, occurrences of x on different branches can be created only by a disjunctive rule application to a formula containing x. Therefore, an application of the disjunctive rule to a formula ψ causes the universal variables of ψ to become "non-universal" variables. That is, all "non-universal" variables are a result of a disjunction within the scope of a $\Box$, corresponding to the fact that $\Box$ does not distribute over $\vee$.

Consider $1 : \Box(p \vee q)$ and suppose we reduce this to $1.x : p \vee q$. At this point, x is universal, but it will become "non-universal" once we apply the disjunctive rule to obtain two branches, the first containing $1.x : p$ and the second containing $1.x : q$. An instantiation of x to 1 (say) on the first branch must now instantiate x to 1 on the second branch as well since x is no longer universal. But it may also be necessary to reduce $1 : \Box(p \vee q)$ once again on the second branch to obtain $1.y : p \vee q$ to obtain closure. There is clearly an interaction between the box-rule and the disjunctive rule. We capture this interaction by ensuring that when the disjunctive rule makes universal variables "non-universal", an additional renaming ($1.y : p \vee q$) of the disjunctive-formula ($1.x : p \vee q$) that created these "non-universal" variables is generated by the disjunctive rule itself (rather than the box-rule).

The diamond-rule. Our diamond-rule does not introduce a *new* label $\sigma.n$, when it is applied to $X : \Delta : \sigma : \Diamond A$. Instead, each formula $\Diamond A$ is assigned its own unique label $\lceil A \rceil$ which is a Gödelisation of A itself.[4] This rule is easier to implement than the traditional one; and it guarantees that (up to renaming of free variables) only a limited number of different labels can occur in a proof, which depends on the number of different sub-formulae in the input formula.

The box-rule for symmetric or euclidean logics can shorten labels. For example, the tableau formula $X' : \Delta' : 1 : A$ is obtainable from $X : \Delta : 1.(1) : \Box A$ for symmetric logics. The semantics for serial logics guarantee that all labels define worlds, but in non-serial logics, the label 1 may be defined when $1.(1)$ is undefined. To ensure that the formula $X' : \Delta' : 1 : A$ or one of its descendants is used to close a branch only if the label $1.(1)$ *is* defined, the label $1.(1)$ is made part of Δ' (see Section 4.3).

[4] Similar features were used in leanTAP [7] for first-order predicate logic, and in other labelled proof systems for modal logics, but within the context of ground labels; see [28].

Such problems do not occur when rule applications always lengthen labels since τ has to be defined if $\tau.l$ is defined.

All expansion rules are sound and *invertible* (some denominator of each rule is **L**-satisfiable *iff* the numerator is **L**-satisfiable). Thus, unlike traditional modal tableau methods where the order of (their non-invertible) rule applications is crucial [12, 16], the order of rule application is *immaterial*.

The differences in the calculi for different logics **L** is mainly in the box-rule, with different denominators for different logics. In addition, a simpler version of the closure rule can be used if the logic is serial.

4.2. Tableau Expansion Rules

There are four expansion rules, one for each class of complex (non-literal) formulae (conjunctive, disjunctive, box-, and diamond-formulae). If we wanted to avoid the restriction to NNF, we would have several types of formulae in each class and, accordingly, would need several (similar) rules for each class (and an extra rule for double negation). Since we assume that all our formulae are in NNF, we need just one rule for each of the four classes.

As usual, in each rule, the formula above the horizontal line is its *numerator* (the premiss) and the formula(e) below the horizontal line, possibly separated by vertical bars, are its *denominators* (the conclusions). All expansion rules (including the box-rule) are "destructive"; that is, once the (appropriate) rule has been applied to a formula occurrence to expand a branch, that formula occurrence is not used again to expand that branch. Note that we permit multiple occurrences of the same formula on the same branch (nevertheless, when a branch is identified with the set of formulae it contains, these occurrences collapse to one).

Definition 4.2. Given a tableau $\mathcal{T}$, a new tableau $\mathcal{T}'$ may be constructed from $\mathcal{T}$ by applying one of the **L**-*expansion rules* from Table III as follows: If the numerator of a rule occurs on a branch $\mathcal{B}$ in $\mathcal{T}$, then the branch $\mathcal{B}$ is extended by the addition of the denominators of that rule. For the disjunctive rule the branch splits and the formulae in the right and left denominator, respectively, are added to the two resulting sub-branches instead.

The box-rule(s) shown in Table III require explanation. The form of the rule is determined by the index **L** in the accompanying table. But some of the denominators have side conditions that determine when they are applicable. For example, the constraint $\sigma_5 = 1.l_5$ means that (5) is part of the denominator only when the numerator of the box-rule is of the form $X : \Delta : 1.l_5 : \Box A$. Similarly, the constraints $\sigma_2 = \tau_2.l_2$ and $\sigma_4 = \tau_4.l_4$ for the (4^r) and (B) denominators mean these rules can be used only for a numerator of the form $X : \Delta : \sigma : \Box A$ where $|\sigma| \geq 2$, thereby guaranteeing that the *strictly shorter* labels τ_2 and τ_4 that appear in the respective denominators are properly defined. The table indicates that the rules for a logic **L** and its serial version **LD** are identical because these logics are distinguished by

Table III. Tableau expansion rules.

$\dfrac{X:\Delta:\sigma:A\land B}{\begin{array}{c}X:\Delta:\sigma:A\\ X':\Delta':\sigma':B\end{array}}$	*Conjunctive rule.* $X':\Delta':\sigma':B$ is a $\mathcal{T}$-renaming of $X:\Delta:\sigma:B$.
$\dfrac{X:\Delta:\sigma:A\lor B}{\begin{array}{c\|c}\emptyset:\Delta_1:\sigma_1:A & \emptyset:\Delta_1:\sigma_1:B\\ X_2:\Delta_2:\sigma_2:A\lor B & X_3:\Delta_3:\sigma_3:A\lor B\end{array}}$	*Disjunctive rule.* The $X_i:\Delta_i:\sigma_i:A\lor B$ are $\mathcal{T}$-renamings of $X:\Delta:\sigma:A\lor B$ (for $1\le i\le 3$) with disjoint X_i. If $X=\emptyset$ then the renamings for $i=2$ and $i=3$ are omitted.
$\dfrac{X:\Delta:\sigma:\Diamond A}{X:\Delta:\sigma.\lceil A\rceil:A}$	*Diamond-rule.* $\lceil\cdot\rceil$ is an arbitrary but fixed bijection from the set of formulae to $\mathbb{N}$.
$\dfrac{X:\Delta:\sigma:\Box A}{\begin{array}{cc}X\cup\{x\}:\Delta:\sigma.(x):A & (K)\\ X_1\cup\{x_1\}:\Delta_1:\sigma_1.(x_1):\Box A & (4)\\ X_2:\Delta_2\cup\{\sigma_2\}:\tau_2:\Box A & (4^r)\\ X_3:\Delta_3:\sigma_3:A & (T)\\ X_4:\Delta_4\cup\{\sigma_4\}:\tau_4:A & (B)\\ X_5\cup\{x_5\}:\Delta_5\cup\{\sigma_5\}:1.(x_5):\Box A & (5)\end{array}}$	*Box-rule.* The formulae $X_i:\Delta_i:\sigma_i:\Box A$ are $\mathcal{T}$-renamings of $X:\Delta:\sigma:\Box A$ for $1\le i\le 5$. The variables $x, x_1, x_5 \in \mathit{Vars}$ are new to $\mathcal{T}$. The sets $X\cup\{x\}$, $X_1\cup\{x_1\}$, X_2, X_3, X_4, $X_5\cup\{x_5\}$ are disjoint. Also, $\sigma_2=\tau_2.l_2$, $\sigma_4=\tau_4.l_4$, and $\sigma_5=1.l_5$. The form of the denominator depends on the logic **L**, and is determined by including every denominator corresponding to the entry for **L** in the table below.

Logics	Box-rule denominator
K, **D**	(K)
T	(K), (T)
KB, **KDB**	(K), (B)[a]
K4, **KD4**	(K), (4)
K5, **KD5**	(K), (4)[a], (4^r)[a], (5)[b]

[a] Only included if $|\sigma|\ge 2$.

Logics	Box-rule denominator
K45, **K45D**	(K), (4), (4^r)[a]
K4B,	(K), (B)[a], (4), (4^r)[a]
B	(K), (T), (B)[a]
S4	(K), (T), (4)
S5	(K), (T), (4), (4^r)[a]

[b] Only included if $|\sigma|=2$.

the form of our closure rule; see Definition 4.5. Various other ways to define the calculi for serial logics exist; see [16].

The expansion rules rename the universal variables in the denominators to ensure that no two literals in a tableau, which may be used for closure, share the same universal variables. This is important because our closure rule uses a single instantiation of all universal variables. We could do without renamings if the closure rule used separate instantiations of the universal variables of the closing literals (including those literals used for justification). This is a technicality since all proofs would still go through with minor changes.

4.3. THE SUBSTITUTION RULE AND THE CLOSURE RULE

By definition, the substitution rule allows us to apply *any* substitution at *any* time to a tableau. In practice, however, it makes sense to apply only "useful" substitutions; that is, those most general substitutions which allow to close a branch of the tableau.

DEFINITION 4.3. *Substitution rule:* Given a tableau $\mathcal{T}$, a new tableau $\mathcal{T}' = \mathcal{T}\mu$ may be constructed from $\mathcal{T}$ by applying a substitution μ to $\mathcal{T}$ that instantiates free variables in $\mathcal{T}$ with other free variables or natural numbers.

In tableaux for modal logics without free variables as well as in free-variable tableaux for first-order logic, a tableau branch is closed if it contains complementary literals since this immediately implies the existence of an inconsistency. Here, however, this is not always the case because the labels of the complementary literals may be conditional. For example, the (apparently contradictory) pair $\emptyset : 1.(1) : p$ and $\emptyset : 1.(1) : \neg p$ is not necessarily inconsistent since the world $\mathbf{I}(1.(1))$ may not exist in the chosen model. Before declaring this pair to be inconsistent, we therefore have to ensure that $\mathbf{I}(1.(1)) \neq \bot$ for all $\mathbf{L}$-interpretations satisfying the tableau branch $\mathcal{B}$ that is to be closed. Fortunately, this knowledge can be deduced from other formulae on $\mathcal{B}$. Thus in our example, a formula like $\psi = X : 1.1 : A$ on $\mathcal{B}$ would "justify" the use of the literal pair $\emptyset : 1.(1) : p$ and $\emptyset : 1.(1) : \neg p$ for closing the branch $\mathcal{B}$ since any $\mathbf{L}$-interpretation $\langle \mathbf{M}, \mathbf{I} \rangle$ satisfying $\mathcal{B}$ has to satisfy ψ, and, thus, $\mathbf{I}(1.(1)) = \mathbf{I}(1.1) \neq \bot$ has to be a world in the chosen model $\mathbf{M}$. The crucial point is that the label 1.1 of ψ is *unconditional* exactly in the *conditional* positions of $\emptyset : 1.(1) : p$ and $\emptyset : 1.(1) : \neg p$. These observations are now extended to the general case of arbitrary *ground* labels.

DEFINITION 4.4. A ground label σ with j-th position $[n_j]$ ($1 \leq j \leq |\sigma|$) is *justified* on a branch $\mathcal{B}$ if there is some set $\mathcal{F} \subseteq \mathcal{B}$ of tableau formulae such that for every j:

1. some label in $lab(\mathcal{F})$ has (an unconditional but otherwise identical) j-th position n_j; and

2. for all $\tau \in lab(\mathcal{F})$: if $|\tau| \geq j$ then the j-th position in τ is n_j or (n_j).

DEFINITION 4.5. Given a tableau $\mathcal{T}$ and a substitution $\lambda : Univ(\mathcal{T}) \to \mathbb{N}$ that instantiates universal variables in $\mathcal{T}$ with natural numbers, the $\mathbf{L}$-*closure rule* allows to construct a new tableau $\mathcal{T}'$ from $\mathcal{T}$ by marking $\mathcal{B}$ in $\mathcal{T}$ as closed provided that:

1. the branch $\mathcal{B}\lambda$ of $\mathcal{T}\lambda$ contains a pair $X : \Delta : \sigma : p$ and $X' : \Delta' : \sigma : \neg p$ of complementary literals; and

2. (a) the logic $\mathbf{L}$ is serial, or (b) all labels in $\{\sigma\} \cup \Delta \cup \Delta'$ are ground and justified on $\mathcal{B}\lambda$.

Note: the substitution λ that instantiates universal variables is *not* applied to the tableau when the branch is closed; it must merely exist.

By definition, only complementary *literals* close tableau branches, but in theory, pairs of complementary *complex formulae* could be used as well (that, however, would lead to additional choice points in the proof search).

4.4. TABLEAU PROOFS

We now have the ingredients to define the notion of a tableau proof.

DEFINITION 4.6. A sequence $\mathcal{T}^0, \ldots, \mathcal{T}^r$ of tableaux is an **L**-*proof* for the **L**-unsatisfiability of a formula A if:

1. $\mathcal{T}^0$ consists of the single node $\emptyset : \emptyset : 1 : A$;
2. for $1 \leq m \leq r$, the tableau $\mathcal{T}^m$ is constructed from $\mathcal{T}^{m-1}$ by applying an **L**-expansion rule (Def. 4.2), the substitution rule (Def. 4.3), or the **L**-closure rule (Def. 4.5); and
3. all branches in $\mathcal{T}^r$ are marked as closed.

Theorems 4.7 and 4.9 state soundness and completeness for our calculus with respect to the Kripke semantics for logic **L**; proofs in Sections 5 and 6.

THEOREM 4.7. (Soundness). *Let A be a formula in NNF. If there is an* **L**-*proof $\mathcal{T}^0, \ldots, \mathcal{T}^r$ for the* **L**-*unsatisfiability of A (Def. 4.6), then A is* **L**-*unsatisfiable.*

We prove completeness for the non-deterministic and unrestricted version of the calculus, and also for all tableau procedures based on this calculus that deterministically choose the next formula for expansion (in a *fair* way) and that only apply most general closing substitutions.

DEFINITION 4.8. Given an open tableau $\mathcal{T}$, a *tableau procedure* Ψ deterministically chooses an open branch $\mathcal{B}$ in $\mathcal{T}$ and a non-literal tableau formula ψ on $\mathcal{B}$ for expansion.

The tableau procedure Ψ is fair if, in the (possibly infinite) tableau constructed using Ψ (where no substitution is applied and no branch is closed), every formula is used for expansion of every branch on which it occurs.

THEOREM 4.9. (Completeness). *Let Ψ be a fair tableau procedure, and let A be an* **L**-*unsatisfiable formula in NNF. Then there is a (finite) tableau proof $\mathcal{T}^0, \ldots, \mathcal{T}^r$ for the* **L**-*unsatisfiability of A, where $\mathcal{T}^i$ is constructed from $\mathcal{T}^{i-1}$ $(1 \leq i \leq r)$ by*

1. *applying the appropriate* **L**-*expansion rule to the branch $\mathcal{B}$ and the formula ψ on $\mathcal{B}$ chosen by Ψ from $\mathcal{T}^{i-1}$; or*

2. *applying a most general substitution such that the* **L***-closure rule can be applied to a previously open branch in* $\mathcal{T}^{i-1}$.

When a tableau proof is constructed according to Theorem 4.9, i.e., using a fair tableau procedure, the remaining choices are (1) whether a branch (that can be closed) is closed or further expanded and (2) in case there are different possibilities to close a branch, which of different most general closing substitutions is applied. An implementation has to resolve this non-determinism (for example, using a backtracking mechanism). As long as no branch is closed, the expansion is deterministic.

There is clearly an interaction between the box-rule, the closure rule and the disjunctive rule. This interaction can be used to restrict the search space by ensuring that when the disjunctive rule "frees" universal variables, an additional renaming $(1.y : p \vee q)$ of the disjunctive-formula $(1.x : p \vee q)$ that created these "non-universal" variables is generated by the disjunctive rule, but only if demanded by the closure rule (that is, when the free variable x gets instantiated during branch closure). Applying this same criterion to the copy $(1.y : p \vee q)$ will produce another copy $(1.z : p \vee q)$, but only upon demand, and so on.

EXAMPLE 4.10. We prove that $A = \Box(p \to q) \to (\Box p \to (\Box q \wedge \Box p))$ is a **K**-theorem. To do this, we first transform the negation of A into NNF; the result is

$$B \quad = \quad \mathrm{NNF}(\neg A) \quad = \quad \Box(\neg p \vee q) \wedge \Box p \wedge (\Diamond\neg q \vee \Diamond\neg p) \enspace .$$

The (fully expanded) tableau $\mathcal{T}$, that is part of the proof for the **K**-unsatisfiability of B is shown in Figure 2. The nodes of the tableau are numbered; a pair $[i; j]$ is attached to the i-th node, the number j denotes that node i has been created by applying an expansion rule to the formula in node j. Note, that by applying the disjunctive rule to 6, the nodes 11 to 14 are added; 13 and 14 are renamings of 6. The variable y_1 is no longer universal in 11 and 12.

When the substitution $\mu = \{y_1/\lceil\neg q\rceil\}$ is applied to $\mathcal{T}$, the branches of the resulting tableau $\mathcal{T}\mu$ can be closed as follows, thereby completing the tableau proof. The left branch $\mathcal{B}_1$ of $\mathcal{T}\mu$ can be closed using the universal-variable substitution $\lambda_1 = \{x/\lceil\neg q\rceil\}$ as $\mathcal{B}_1\lambda_1$ contains the complementary pair $\{\lceil\neg q\rceil\} : 1.(\lceil\neg q\rceil) : p$ and $\emptyset : 1.(\lceil\neg q\rceil) : \neg p$ in nodes 7 and 11, respectively. The label $1.(\lceil\neg q\rceil)$ of these literals is justified on $\mathcal{B}_1\lambda_1$ by label $1.\lceil\neg q\rceil$ of formula 10. In this case, the complementary literals contain conditional labels which are only justified by a third formula on the branch, so checking for justification is indispensable. The middle branch $\mathcal{B}_2$ of $\mathcal{T}\mu$ can be closed using the same universal-variable substitution $\lambda_2 = \lambda_1 = \{x/\lceil\neg q\rceil\}$ as for the left branch. The branch $\mathcal{B}_2\lambda_2$ then contains the complementary literals $\emptyset : 1.(\lceil\neg q\rceil) : \neg q$ and $\emptyset : 1.(\lceil\neg q\rceil) : q$ in nodes 10 and 12. The label is again justified by formula 10, which in this case is one of the complementary literals. Note that the middle branch $\mathcal{B}_2$ can be closed only by the substitution $\mu = \{y_1/\lceil\neg q\rceil\}$, other choices will not suffice. The right branch $\mathcal{B}_3$ of $\mathcal{T}\mu$ can be closed using

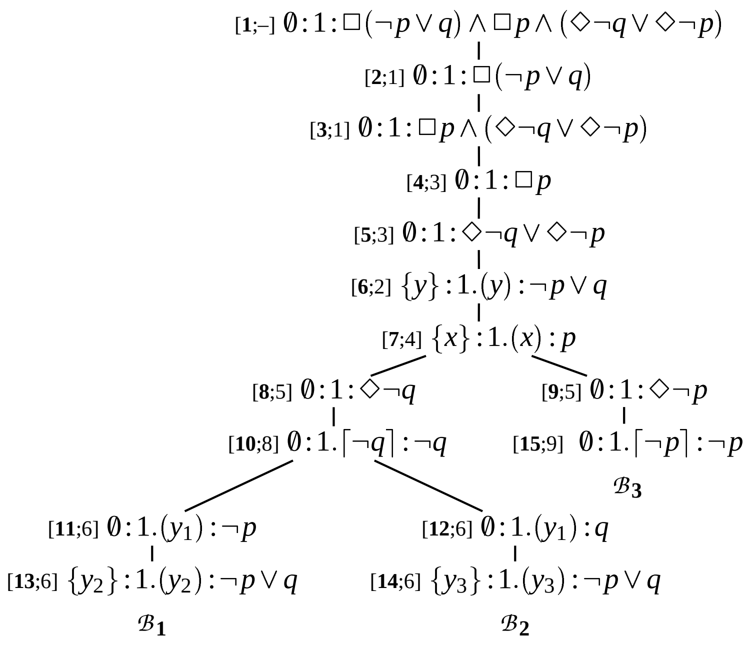
[1;–] $\emptyset : 1 : \Box(\neg p \vee q) \wedge \Box p \wedge (\Diamond\neg q \vee \Diamond\neg p)$
[2;1] $\emptyset : 1 : \Box(\neg p \vee q)$
[3;1] $\emptyset : 1 : \Box p \wedge (\Diamond\neg q \vee \Diamond\neg p)$
[4;3] $\emptyset : 1 : \Box p$
[5;3] $\emptyset : 1 : \Diamond\neg q \vee \Diamond\neg p$
[6;2] $\{y\} : 1.(y) : \neg p \vee q$
[7;4] $\{x\} : 1.(x) : p$
[8;5] $\emptyset : 1 : \Diamond\neg q$
[9;5] $\emptyset : 1 : \Diamond\neg p$
[10;8] $\emptyset : 1.\lceil\neg q\rceil : \neg q$
[15;9] $\emptyset : 1.\lceil\neg p\rceil : \neg p$
$\mathcal{B}_3$
[11;6] $\emptyset : 1.(y_1) : \neg p$
[12;6] $\emptyset : 1.(y_1) : q$
[13;6] $\{y_2\} : 1.(y_2) : \neg p \vee q$
[14;6] $\{y_3\} : 1.(y_3) : \neg p \vee q$
$\mathcal{B}_1$
$\mathcal{B}_2$

Figure 2. The tableau $\mathcal{T}$ from Example 4.10.

the universal-variable substitution $\lambda_3 = \{x/\lceil\neg p\rceil\}$ as $\mathcal{B}_3\lambda_3$ then contains the pair $\{\lceil\neg p\rceil\} : 1.(\lceil\neg p\rceil) : p$ and $\{\lceil\neg p\rceil\} : 1.\lceil\neg p\rceil : \neg p$ of complementary literals in nodes 7 resp. 15. The label $1.(\lceil\neg p\rceil)$ of node 7 is justified on $\mathcal{B}_3$ by formula 15.

The universal-variable substitution $\lambda_1 = \lambda_2 = \{x/\lceil\neg q\rceil\}$ that closes $\mathcal{B}_1$ and $\mathcal{B}_2$ is incompatible with the substitution $\lambda_3 = \{x/\lceil\neg p\rceil$ that closes $\mathcal{B}_3$. Therefore, if the variable x were not universal in formula 7, the tableau could not be closed; a second instance of formula 7 would have to be added.

In the above example, the only reason for instantiating free variables is to make the labels of closing literals identical. There are situations however, where a free variable has to be instantiated solely to make sure that the labels of the closing literals are justified (and not to make them identical).[5]

EXAMPLE 4.11. This example demonstrates why the second part Δ of tableau formulae is needed. Consider the formula $A = \Box\Box p \to p$, which is a theorem of the serial logic **KDB** but not of the non-serial logic **KB**. The calculi for both logics have the same tableau expansion rules (they only differ in the closure rule); the tableau $\mathcal{T}$ for the NNF $\Box\Box p \wedge \neg p$ of $\neg A$ shown in Figure 3 has been constructed using these rules.

The (single) branch of $\mathcal{T}$ contains the complementary literals $\phi_1 = \emptyset : \emptyset : 1 : \neg p$ and $\phi_2 = \{x\} : \{1.(x)\} : 1 : p$ in nodes 3 and 6. Using the closure rule of the calculus

[5] This happens, for example, during the construction of a tableau proof for the unsatisfiability of the formula $(\Box q) \wedge (\Diamond\Box\Diamond r) \wedge \Box((\Diamond\Box(p \wedge \neg p)) \vee \neg q)$.

$$\begin{array}{c}
[\mathbf{1};-]\ \emptyset : \emptyset : 1 : \Box\Box p \wedge \neg p \\
| \\
[\mathbf{2};1]\ \emptyset : \emptyset : 1 : \Box\Box p \\
| \\
[\mathbf{3};1]\ \emptyset : \emptyset : 1 : \neg p \\
| \\
[\mathbf{4};2]\ \{x\} : \emptyset : 1.(x) : \Box p \\
| \\
[\mathbf{5};4]\ \{x,y\} : \emptyset : 1.(x).(y) : p \\
| \\
[\mathbf{6};4]\ \{x\} : \{1.(x)\} : 1 : p
\end{array}$$

Figure 3. The tableau $\mathcal{T}$ from Example 4.11.

for **KDB**, that does not require a justification of labels, the branch can be closed with ϕ_1 and ϕ_2. Thus, a tableau proof for the **KDB**-unsatisfiability of $\neg A$ can be constructed from $\mathcal{T}$. Using the closure rule of the calculus for the logic **KB**, however, the branch cannot be closed. That closure rule requires the labels of the two complementary literals to be justified on the branch, including all labels in the sets $\Delta_1 = \emptyset$ and $\Delta_2 = \{1.(x)\}$—but neither the label $1.(x) \in \Delta_2$ nor any of its instances $1.(n)$ is justified. No tableau proof for the **KB**-unsatisfiability of A can be constructed, which is correct as $\neg A$ is, in fact, **KB**-satisfiable.

5. Soundness Proof

The following two lemmata, which will be used in the soundness proof, follow immediately from the definitions. The first one states that a tableau formula ψ and a renaming ψ' of ψ are equivalent. The second lemma states that if a label σ is *justified* on a tableau branch $\mathcal{B}$ satisfied by an interpretation $\langle \mathbf{M}, \mathbf{I} \rangle$, then $\mathbf{I}(\sigma)$ has to be a world in $\mathbf{M}$ (even if σ is conditional).

LEMMA 5.1. *Let $\langle \mathbf{M}, \mathbf{I} \rangle$ be an $\mathbf{L}$-interpretation, μ a grounding substitution, ψ a formula in a tableau $\mathcal{T}$, and ψ' a $\mathcal{T}$-renaming of ψ. Then $\langle \mathbf{M}, \mathbf{I}, \mu \rangle \models \psi$ if and only if $\langle \mathbf{M}, \mathbf{I}, \mu \rangle \models \psi'$.*

LEMMA 5.2. *Let $\langle \mathbf{M}, \mathbf{I} \rangle$ be an $\mathbf{L}$-interpretation, where $\mathbf{M} = \langle W, R, V \rangle$, let $\mathcal{B}$ be a tableau branch, and let σ be a ground label. If $\langle \mathbf{M}, \mathbf{I}, \mu \rangle$ satisfies $\mathcal{B}$, and the label σ is justified on $\mathcal{B}$, then $\mathbf{I}(\sigma) \in W$.*

The crux of the proof is to show that **L**-satisfiability is preserved by the expansion rule (Lemma 5.5), the substitution rule (Lemma 5.6), and the closure rule (Lemma 5.7). Consequently, if the initial tableau consisting of the input formula A is **L**-satisfiable, then all tableaux for A are **L**-satisfiable. None can be closed as closed tableaux are not **L**-satisfiable. Hence, the existence of a closed tableau for A implies the **L**-unsatisfiability of A.

For the soundness proof we restrict all considerations to *standard* interpretations, where (ground) labels are interpreted "in the right way," such that the diamond-rule preserves **L**-satisfiability (in standard models). Moreover, for serial logics, all ground labels are assigned a world by the interpretation function, which helps the closure rule to preserve **L**-satisfiability.

DEFINITION 5.3. An **L**-interpretation $\langle \mathbf{M}, \mathbf{I} \rangle$, where $\mathbf{M} = \langle W, R, V \rangle$, is a *standard* interpretation provided that:

1. For all ground labels $\tau.[n]$:

$$\text{if } \mathbf{I}(\tau) \in W \text{ and } \mathbf{I}(\tau) \models \Diamond A_n, \text{ then } \mathbf{I}(\tau.[n]) \in W \text{ and } \mathbf{I}(\tau.[n]) \models A_n,$$

   where A_n is the formula for which $n = \lceil A_n \rceil$ ($\lceil \cdot \rceil$ is the bijection from the set of formulae to the set of natural numbers used for the diamond-rule).

2. If the logic **L** is serial, then $\mathbf{I}(\sigma) \in W$ for all ground labels σ.

The restriction to standard interpretations makes sense: every **L**-model **M** that satisfies a formula A can be combined with a label interpretation **I**, such that $\langle \mathbf{M}, \mathbf{I} \rangle$ is a standard interpretation and satisfies the initial tableau $\emptyset : \emptyset : 1 : A$.

LEMMA 5.4. *Given a formula A in NNF and an* **L**-*model* **M** *that satisfies A, there is a standard* **L**-*interpretation $\langle \mathbf{M}, \mathbf{I} \rangle$ that satisfies the tableau consisting of the singleton tableau formula $\emptyset : \emptyset : 1 : A$.*

PROOF. As $\mathbf{M} = \langle W, R, V \rangle$ satisfies A, we know that there is some world $w_1 \in W$ such that $w_1 \models A$. Now, for $n \geq 1$, let A_n be the formula for which $n = \lceil A_n \rceil$ (where $\lceil \cdot \rceil$ is the bijection from the set of formulae to the set of natural numbers used for the diamond-rule) and create **I** as follows. Put $\mathbf{I}(1) = w_1$, and for every ground label of the form $\tau.n$: (a) if there is a world $w \in W$ such that $\mathbf{I}(\tau) R w$ and $w \models A_n$ then put $\mathbf{I}(\tau.n) = \mathbf{I}(\tau.(n)) = w$; (b) else, if there is no such world w, but there is a world w' that is reachable from $\mathbf{I}(\tau)$, then put $\mathbf{I}(\tau.n) = \mathbf{I}(\tau.(n)) = w'$; (c) else, if there is no world reachable from $\mathbf{I}(\tau)$, put $\mathbf{I}(\tau.n) = \mathbf{I}(\tau.(n)) = \perp$.

The **L**-interpretation $\langle \mathbf{M}, \mathbf{I} \rangle$ is a standard interpretation by way of its definition, and in addition satisfies the tableau consisting of $\phi_0 = \emptyset : \emptyset : 1 : A$. To prove this, we have to show that for all grounding substitutions μ there is a branch $\mathcal{B}$ in this tableau such that for all substitutions λ the ground formula $\phi_0 \lambda_{|\emptyset} \mu$ is satisfied by $\langle \mathbf{M}, \mathbf{I} \rangle$. Since $\phi_0 \lambda_{|\emptyset} \mu = \phi_0$ this reduces to showing that (a) $\mathbf{I}(1) = \perp$ or $\mathbf{I}(1) \models A$, and (b) $\langle \mathbf{M}, \mathbf{I} \rangle$ satisfies the label 1. Condition (a) is satisfied since $\mathbf{I}(1) \models A$ by choice, and Condition (b) holds because there are no labels $\tau.n$ in $ipr(1) \cup \{1\}$. $\square$

Next we prove that satisfiability by standard interpretations is preserved by the tableau expansion rules, the substitution, and the closure rule.

In the proofs of Lemmata 5.5–5.7 we make use of the fact that, by definition, a tableau formula $X:\Delta:\sigma:F$ is satisfied by $\langle \mathbf{M},\mathbf{I},\mu\rangle$ iff for all grounding substitutions λ:

$$\mathbf{I}(\widetilde{\sigma}) = \bot, \text{ or } \mathbf{I}(\xi) = \bot \text{ for some } \xi \in \widetilde{\Delta}, \text{ or } \mathbf{I}(\widetilde{\sigma}) \models F; \qquad (*)$$

and

$$\text{if } \mathbf{I}(\xi) \neq \bot \text{ for all } \xi \in \widetilde{\Delta}, \text{ then } \langle \mathbf{M},\mathbf{I}\rangle \text{ satisfies } \widetilde{\sigma}; \qquad (**)$$

where $\widetilde{\sigma} = \sigma\lambda_{|X}\mu$ and $\widetilde{\Delta} = \Delta\lambda_{|X}\mu$. We can also formulate $(*)$ as:

$$\text{If } \mathbf{I}(\widetilde{\sigma}) \neq \bot \text{ and } \mathbf{I}(\xi) \neq \bot \text{ for all } \xi \in \widetilde{\Delta}, \text{ then } \mathbf{I}(\widetilde{\sigma}) \models F.$$

LEMMA 5.5. *If the tableau $\mathcal{T}$ is satisfied by the standard* **L**-*interpretation* $\langle \mathbf{M},\mathbf{I}\rangle$, *and $\mathcal{T}'$ is constructed from $\mathcal{T}$ by applying an* **L**-*expansion rule, then* $\langle \mathbf{M},\mathbf{I}\rangle$ *satisfies $\mathcal{T}'$ as well.*

PROOF. We show that for each grounding substitution μ, there is a branch $\mathcal{B}'$ in $\mathcal{T}'$ that is satisfied by $\langle \mathbf{M},\mathbf{I}\rangle$.

By assumption, $\langle \mathbf{M},\mathbf{I},\mu\rangle$ satisfies some branch $\mathcal{B}$ of $\mathcal{T}$. If $\mathcal{T}'$ is constructed from $\mathcal{T}$ by expanding a branch other than $\mathcal{B}$, then $\mathcal{B}$ is a branch of $\mathcal{T}'$ as well, and we are through. For the case that $\mathcal{T}'$ is constructed from $\mathcal{T}$ by expanding the branch $\mathcal{B}$, we show that $\langle \mathbf{M},\mathbf{I},\mu\rangle$ satisfies one of the branches of $\mathcal{T}'$ by cases according to the expansion rule applied.

Conjunctive rule. Let $\phi = X:\Delta:\sigma:(A \wedge B)$ be the formula on $\mathcal{B}$, such that $\mathcal{T}'$ is constructed from $\mathcal{T}$ by adding $\phi_1 = X:\Delta:\sigma:A$ and $\phi'_2 = X':\Delta':\sigma':B$ to $\mathcal{B}$. Because $\langle \mathbf{M},\mathbf{I},\mu\rangle \models \phi$, $(*)$ and $(**)$ hold for all grounding substitutions λ where $\widetilde{\sigma} = \zeta = \sigma\lambda_{|X}\mu$, $\widetilde{\Delta} = \Delta\lambda_{|X}\mu$, and $F = A \wedge B$. As $\mathbf{I}(\zeta) \models A \wedge B$ implies $\mathbf{I}(\zeta) \models A$ and $\mathbf{I}(\zeta) \models B$, the same is true for $F = A$ and for $F = B$. Therefore, $\langle \mathbf{M},\mathbf{I},\mu\rangle$ satisfies ϕ_1 and ϕ_2 and, by Lemma 5.1, the renaming ϕ'_2; thus, $\langle \mathbf{M},\mathbf{I},\mu\rangle$ satisfies $\mathcal{B} \cup \{\phi_1, \phi'_2\}$, which is a branch in $\mathcal{T}'$.

Disjunctive rule. Let $\phi = X:\Delta:\sigma:(A \vee B)$ be the formula in $\mathcal{B}$, such that $\mathcal{T}'$ is constructed from $\mathcal{T}$ by adding $\phi'_1 = \emptyset:\Delta_1:\sigma_1:A$ and $\phi' = X_2:\sigma_2:\Delta_2:A \vee B$ to $\mathcal{B}$ obtaining $\mathcal{B}'_1$ and adding $\phi'_2 = \emptyset:\Delta_1:\sigma_1:B$ and $\phi'' = X_3:\Delta_3:\sigma_3:A \vee B$ to $\mathcal{B}$ obtaining $\mathcal{B}'_2$. $\langle \mathbf{M},\mathbf{I},\mu\rangle \models \phi$ implies that both ϕ' and ϕ'' are satisfied by $\langle \mathbf{M},\mathbf{I},\mu\rangle$ (using Lemma 5.1), and it implies that $(*)$ and $(**)$ hold for all grounding substitutions λ' where $\widetilde{\sigma} = \zeta = \sigma\lambda'_{|X}\mu$, $\widetilde{\Delta} = \Delta\lambda'_{|X}\mu$, and $F = A \vee B$. Since $\mathbf{I}(\zeta) \models A \vee B$ implies that $\mathbf{I}(\zeta) \models A$ or $\mathbf{I}(\zeta) \models B$, the same is true for $F = A$ or for $F = B$. In particular, if we chose λ' equal to μ, then $(*)$ and $(**)$ hold for $\widetilde{\sigma} = \sigma\mu_{|X}\mu = \sigma\mu = \sigma\lambda_{|\emptyset}\mu$, $\widetilde{\Delta} = \Delta\lambda'_{|X}\mu = \Delta\lambda_{|\emptyset}\mu$, and $F = A$ or $F = B$, where λ is an *arbitrary* grounding substitution. This implies that $\langle \mathbf{M},\mathbf{I},\mu\rangle$ satisfies ϕ_1 or ϕ_2 and thus, by Lemma 5.1, (at least) one of the renamings ϕ'_1 and ϕ'_2. Therefore, $\langle \mathbf{M},\mathbf{I},\mu\rangle$ satisfies one of the branches $\mathcal{B}'_1$ and $\mathcal{B}'_2$ in $\mathcal{T}'$.

Table IV. Formulae added by the box-rule.

ψ	$\widetilde{\sigma}$	$\widetilde{\Delta}$	F	Condition
$\phi_{(\mathrm{K})} = X \cup \{x\} : \Delta : \sigma.(x) : A$	$\zeta.(n)$	$\Delta\lambda_{\vert X}\mu$	A	—
$\phi_{(4)} = X \cup \{x\} : \Delta : \sigma.(x) : \Box A$	$\zeta.(n)$	$\Delta\lambda_{\vert X}\mu$	$\Box A$	**L** transitive, or **L** euclidean and $\vert\sigma\vert \geq 2$
$\phi_{(4^r)} = X : \Delta \cup \{\sigma\} : \tau : \Box A$	ζ'	$(\Delta \cup \{\sigma\})\lambda_{\vert X}\mu$	$\Box A$	**L** euclidean
$\phi_{(5)} = X \cup \{x\} : \Delta \cup \{\sigma\} : 1.(x) : \Box A$	$1.(n)$	$(\Delta \cup \{\sigma\})\lambda_{\vert X}\mu$	$\Box A$	**L** euclidean and $\vert\sigma\vert = 2$
$\phi_{(\mathrm{T})} = X : \Delta : \sigma : A$	ζ	$\Delta\lambda_{\vert X}\mu$	A	**L** reflexive
$\phi_{(\mathrm{B})} = X : \Delta \cup \{\sigma\} : \tau : A$	ζ'	$(\Delta \cup \{\sigma\})\lambda_{\vert X}\mu$	A	**L** symmetric

where $n = \mu(x)$, $\sigma = \tau.l$, $\zeta = \sigma\lambda_{|X}\mu$, $\zeta' = \tau\lambda_{|X}\mu$.

Diamond-rule. Let $\phi = X : \Delta : \sigma : \Diamond A$ be the formula on $\mathcal{B}$, such that $\mathcal{T}'$ is constructed from $\mathcal{T}$ by adding $\phi_1 = X : \Delta : \sigma.\lceil A \rceil : A$ to $\mathcal{B}$. $\langle \mathbf{M}, \mathbf{I}, \mu \rangle \models \phi$ implies that $(*)$ and $(**)$ hold for all grounding substitutions λ where $\widetilde{\sigma} = \zeta = \sigma\lambda_{|X}\mu$, $\widetilde{\Delta} = \Delta\lambda_{|X}\mu$, and $F = \Diamond A$. Considering $(*)$, $\mathbf{I}(\zeta) = \bot$ implies $\mathbf{I}(\zeta.\lceil A \rceil) = \bot$; and $\mathbf{I}(\zeta) \models \Diamond A$ implies that $w \models A$ for some $w \in W$ reachable from $\mathbf{I}(\zeta)$. Because $\langle \mathbf{M}, \mathbf{I} \rangle$ is a standard interpretation, $\mathbf{I}(\zeta.\lceil A \rceil) \in W$ and $\mathbf{I}(\zeta.\lceil A \rceil) \models A$. Considering $(**)$, if $\mathbf{I}(\xi) \neq \bot$ for all $\xi \in \Delta\lambda_{|X}\mu$, then $\langle \mathbf{M}, \mathbf{I} \rangle$ satisfies ζ; in that case, (a) if $\mathbf{I}(\zeta) = \bot$, then $\langle \mathbf{M}, \mathbf{I} \rangle$ satisfies $\zeta.\lceil A \rceil$ as well, because ζ has to be conditional, (b) if $\mathbf{I}(\zeta) \neq \bot$, then $\mathbf{I}(\zeta) \models \Diamond A$, which implies by definition of standard interpretations that $\mathbf{I}(\zeta.\lceil A \rceil) \in W$. Therefore, $(*)$ and $(**)$ hold with $\widetilde{\sigma} = \sigma.\lceil A \rceil\lambda_{|X}\mu$ and $F = A$ as well, which implies that $\langle \mathbf{M}, \mathbf{I}, \mu \rangle$ satisfies ϕ_1 and, therefore, satisfies the branch $\mathcal{B} \cup \{\phi_1\}$ in $\mathcal{T}'$.

Box-rule. Let $\phi = X : \Delta : \sigma : \Box A$ be the formula in $\mathcal{B}$, such that $\mathcal{T}'$ is constructed from $\mathcal{T}$ by adding to $\mathcal{B}$—according to Table III—renamings of one or more of the formulae ψ shown in the first column of Table IV. The respective formulae ψ are only used if the corresponding conditions in the last column of Table IV is fulfilled. We show that these conditions imply that $\langle \mathbf{M}, \mathbf{I}, \mu \rangle$ satisfies ψ and thus, by Lemma 5.1, the renaming of ψ that is added to the branch. Thus, $\langle \mathbf{M}, \mathbf{I}, \mu \rangle$ satisfies the resulting new branch in $\mathcal{T}'$.

$\langle \mathbf{M}, \mathbf{I}, \mu \rangle \models \phi$ implies that $(*)$ and $(**)$ hold for all grounding substitutions λ where $\widetilde{\sigma} = \zeta = \sigma\lambda_{|X}\mu$, $\widetilde{\Delta} = \Delta\lambda_{|X}\mu$, and $F = \Box A$. For each of the formulae ψ in Table IV we prove that $\langle \mathbf{M}, \mathbf{I}, \mu \rangle$ satisfies ψ by showing that $(*)$ and $(**)$ hold with the corresponding instances of $\widetilde{\sigma}$, $\widetilde{\Delta}$, and F, as shown in columns 2–4 in Table IV. Throughout the rest of this proof, we make use of the fact that the variable x does not occur in σ and Δ and, thus, $\sigma\lambda_{|X}\mu = \zeta = \sigma\lambda_{|X\cup\{x\}}\mu$ and $\Delta\lambda_{|X}\mu = \Delta\lambda_{|X\cup\{x\}}\mu$. Note also that $\sigma.(x)\lambda_{|X\cup\{x\}}\mu = \zeta.(n)$; and $\zeta = \zeta'.[m]$ where $[m] = \mu(l)$.

$\psi = \phi_{(\mathrm{K})}$: $(*)$ If $\mathbf{I}(\zeta.(n)) \neq \bot$ then $\mathbf{I}(\zeta) \neq \bot$ and $\mathbf{I}(\zeta) R \mathbf{I}(\zeta.(n))$. If, in addition, $\mathbf{I}(\xi) \neq \bot$ for all $\xi \in \Delta\lambda_{|X}\mu = \Delta\lambda_{|X\cup\{x\}}\mu$, then we can conclude that $\mathbf{I}(\zeta) \models \Box A$. By the definition of the box-operator, this implies $\mathbf{I}(\zeta.(n)) \models A$. $(**)$ If $\mathbf{I}(\xi) \neq \bot$ for all $\xi \in \Delta\lambda_{|X}\mu$, then $\langle \mathbf{M}, \mathbf{I} \rangle$ satisfies the label $\zeta.(n)$ because it satisfies ζ and the extension (n) is conditional.

$\psi = \phi_{(4)}$: $(*)$ If $\mathbf{I}(\zeta.(n)) = w' \neq \perp$ then $\mathbf{I}(\zeta) = w \neq \perp$ and wRw'. If, in addition, $\mathbf{I}(\xi) \neq \perp$ for all $\xi \in \Delta\lambda_{|X}\mu = \Delta\lambda_{|X\cup\{x\}}\mu$, then we can conclude that $w \models \Box A$. If w' has no successors, then $w' \models \Box A$ for any A vacuously. So let w'' be any world such that $w'Rw''$. (a) If R is transitive, this immediately implies wRw''. (b) If R is euclidean and $|\zeta| \geq 2$, then there is a world w_0 such that w_0Rw. We can derive wRw'' as follows: w_0Rw implies wRw, wRw' and wRw implies $w'Rw$, $w'Rw$ and $w'Rw''$ implies wRw''. Now, since $w \models \Box A$ we have $w'' \models A$, and since w'' is an arbitrary world reachable from w', $w' \models \Box A$. Condition $(**)$ can be proven in the same way as in the previous sub-case.

$\psi = \phi_{(4^r)}$: $(*)$ If $\mathbf{I}(\zeta) = \perp$ or there is some label ξ in $\Delta\lambda_{|X}\mu$ such that $\mathbf{I}(\xi) = \perp$, then there is a label ξ in $(\Delta \cup \{\sigma\})\lambda_{|X}\mu = \Delta\lambda_{|X}\mu \cup \{\zeta\}$ such that $\mathbf{I}(\xi) = \perp$. Otherwise, if $\mathbf{I}(\zeta) = w \models \Box A$, then $\mathbf{I}(\zeta') = w' \neq \perp$ and $w'Rw$. If w' has no successors, then $w' \models \Box A$ for any A vacuously. So let w'' be any world such that $w'Rw''$. Since $\mathbf{L}$ is euclidean, $w'Rw$ and $w'Rw''$ implies wRw'' and $w''Rw$. Thus $w \models \Box A$ implies $w'' \models A$. This holds for all w'' reachable from w', so $\mathbf{I}(\zeta') = w' \models \Box A$. $(**)$ If $\mathbf{I}(\xi) \neq \perp$ for all $\xi \in (\Delta \cup \{\sigma\})\lambda_{|X}\mu$, then $\langle \mathbf{M}, \mathbf{I} \rangle$ satisfies the label $\zeta = \sigma\lambda_{|X}\mu$ and, thus, the label $\zeta' \in ipr(\zeta)$.

$\psi = \phi_{(5)}$: $(*)$ If $\mathbf{I}(1.(n)) \neq \perp$ then $\mathbf{I}(1) \neq \perp$ and $\mathbf{I}(1)\,R\,\mathbf{I}(1.(n))$. If, in addition, $\mathbf{I}(\xi) \neq \perp$ for all ξ in $\widetilde{\Delta}$, then $\mathbf{I}(\sigma\lambda_{|X\cup\{x\}}\mu) = \mathbf{I}(\zeta) \neq \perp$ and $\mathbf{I}(\xi) \neq \perp$ for all ξ in $\Delta\lambda_{|X\cup\{x\}}\mu = \Delta\lambda_{|X}\mu$, which implies $\mathbf{I}(\zeta) \models \Box A$. Also, since $\mathbf{I}(\zeta) = \mathbf{I}(1.[m]) \neq \perp$, we have $\mathbf{I}(1)\,R\,\mathbf{I}(\zeta)$ and $\mathbf{I}(1.(n))\,R\,\mathbf{I}(\zeta)$ because the logic is euclidean. Now, for all worlds w such that $\mathbf{I}(1.(n))\,R\,w$, we have $\mathbf{I}(\zeta)\,R\,w$ (again because the logic is euclidean); and $w \models A$ since $\mathbf{I}(\zeta) \models \Box A$. This holds for all w reachable from $\mathbf{I}(1.(n))$; therefore $\mathbf{I}(1.(n)) \models \Box A$. $(**)$ If $\mathbf{I}(\xi) \neq \perp$ for all $\xi \in (\Delta \cup \{\sigma\})\lambda_{|X}\mu$, then $\langle \mathbf{M}, \mathbf{I} \rangle$ satisfies the label $\sigma\lambda_{|X}\mu = \zeta = 1.[m]$ and, thus, $1 \in ipr(1.[m])$. Because the extension (n) is conditional, $\langle \mathbf{M}, \mathbf{I} \rangle$ then also satisfies $1.(n)$.

$\psi = \phi_{(\mathrm{T})}$: $(*)$ $\mathbf{I}(\zeta) = \perp$, or $\mathbf{I}(\xi) = \perp$ for some $\xi \in \Delta\lambda_{|X}\mu$, or $\mathbf{I}(\zeta) \models \Box A$ which implies $\mathbf{I}(\zeta) \models A$ (by reflexivity). Condition $(**)$ is trivially satisfied in this sub-case.

$\psi = \phi_{(\mathrm{B})}$: $(*)$ If $\mathbf{I}(\zeta) = \perp$ or there is some label ξ in $\Delta\lambda_{|X}\mu$ such that $\mathbf{I}(\xi) = \perp$, then there is a label ξ in $(\Delta \cup \{\sigma\})\lambda_{|X}\mu = \Delta\lambda_{|X}\mu \cup \{\zeta\}$ such that $\mathbf{I}(\xi) = \perp$. Otherwise, if $\mathbf{I}(\zeta) = w \models \Box A$, then $\mathbf{I}(\zeta') = w' \neq \perp$ and $w'Rw$ (since $\zeta = \zeta'.[m]$). Because R is symmetric this implies wRw' and thus $\mathbf{I}(\zeta') = w' \models A$. Condition $(**)$ can be proven in the same way as in the sub-case $\psi = \phi_{(4^r)}$. □

LEMMA 5.6. *If the tableau $\mathcal{T}$ is satisfied by the standard* $\mathbf{L}$*-interpretation* $\langle \mathbf{M}, \mathbf{I} \rangle$*, and $\mathcal{T}'$ is constructed from $\mathcal{T}$ by applying the substitution rule, then $\langle \mathbf{M}, \mathbf{I} \rangle$ satisfies $\mathcal{T}'$ as well.*

PROOF. Let ν be the substitution applied to derive $\mathcal{T}'$ from $\mathcal{T}$. We must show that for each grounding substitution μ, there is a branch $\mathcal{B}'$ in $\mathcal{T}'$ that is satisfied by $\langle \mathbf{M}, \mathbf{I}, \mu \rangle$. Let μ be an arbitrary but fixed grounding substitution. By assumption, $\langle \mathbf{M}, \mathbf{I}, \mu \rangle$ satisfies some branch $\mathcal{B}$ of $\mathcal{T}$. Let $\phi = X{:}\Delta{:}\sigma{:}A$ be an arbitrary formula on $\mathcal{B}$. Let λ be an arbitrary grounding substitution, and define the substitution $\lambda' = \lambda \circ \nu$. Then the substitutions $\mu \circ \lambda_{|X} \circ \nu$ and $\mu \circ \lambda'_{|X}$ are identical, because ν does not instantiate variables in X, and thus $\lambda'_{|X} = (\lambda \circ \nu)_{|X} = \lambda_{|X} \circ \nu$. As $\langle \mathbf{M}, \mathbf{I}, \mu \rangle \models \phi$, $(*)$ and $(**)$ hold with $\widetilde{\sigma} = \sigma\lambda'_{|X}\mu$, $\widetilde{\Delta} = \Delta\lambda'_{|X}\mu$, and $F = A$. This implies, because $\mu \circ \lambda_{|X} \circ \nu = \mu \circ \lambda'_{|X}$, that $(*)$ and $(**)$ hold as well with $\widetilde{\sigma} = \sigma\nu\lambda_{|X}\mu$ and $\widetilde{\Delta} = \Delta\nu\lambda_{|X}\mu$. Thus, $\langle \mathbf{M}, \mathbf{I}, \mu \rangle$ satisfies $\phi\nu$. □

LEMMA 5.7. *If the tableau $\mathcal{T}$ is satisfied by the standard* $\mathbf{L}$*-interpretation* $\langle \mathbf{M}, \mathbf{I} \rangle$*, and $\mathcal{T}'$ is constructed from $\mathcal{T}$ by applying the* $\mathbf{L}$*-closure rule, then* $\langle \mathbf{M}, \mathbf{I} \rangle$ *satisfies $\mathcal{T}'$ as well.*

PROOF. $\mathcal{T}'$ is obtained from $\mathcal{T}$ by marking a branch $\mathcal{B}$ in $\mathcal{T}$ as closed, because it contains formulae $\phi_1 = X_1{:}\Delta_1{:}\sigma_1{:}p$ and $\phi_2 = X_2{:}\Delta_2{:}\sigma_2{:}\neg p$, and there is a substitution λ of the universal variables in $\mathcal{T}$ such that $\sigma_1\lambda_{|X_1} = \sigma_2\lambda_{|X_2} = \xi$ and (a) the logic $\mathbf{L}$ is serial, or (b) all labels ζ in $\{\xi\} \cup \Delta_1\lambda_{|X_1} \cup \Delta_2\lambda_{|X_2}$ are ground and justified on $\mathcal{B}$. Suppose the branch $\mathcal{B}$ were satisfied by $\langle \mathbf{M}, \mathbf{I}, \mu \rangle$ for some grounding substitution μ. Then $\mathbf{I}(\zeta\mu) \in W$, because (1) if the logic $\mathbf{L}$ is serial, then $\mathbf{I}(\zeta\mu) \in W$ since $\langle \mathbf{M}, \mathbf{I}, \mu \rangle$ is a standard interpretation; (2) otherwise, ζ is ground and justified, and thus $\zeta\mu = \zeta$ and $\mathbf{I}(\zeta) \in W$ according to Lemma 5.2. Now we have a contradiction, because $\langle \mathbf{M}, \mathbf{I}, \mu \rangle \models \phi_1$ implies $\mathbf{I}(\xi\mu) = \mathbf{I}(\sigma_1\lambda_{|X_1}\mu) \models p$, and $\langle \mathbf{M}, \mathbf{I}, \mu \rangle \models \phi_2$ implies $\mathbf{I}(\xi\mu) = \mathbf{I}(\sigma_2\lambda_{|X_2}\mu) \models \neg p$.

Thus, our assumption is wrong, and $\mathcal{B}$ is not satisfied by $\langle \mathbf{M}, \mathbf{I}, \mu \rangle$ for any μ. But then there has to be a different branch $\mathcal{B}_0$ in $\mathcal{T}$ for all μ, that occurs in $\mathcal{T}'$ as well and is not affected by marking the branch $\mathcal{B}$ as closed. □

Now we have everything needed to prove soundness of our calculus.

THEOREM 5.8. *Let A be a formula in NNF. If there is an* $\mathbf{L}$*-proof* $\mathcal{T}^0, \ldots, \mathcal{T}^r$ *for the* $\mathbf{L}$*-unsatisfiability of A (Def 4.6), then A is* $\mathbf{L}$*-unsatisfiable.*

PROOF. For a contradiction, suppose there is an $\mathbf{L}$-proof $\mathcal{T}^0, \ldots, \mathcal{T}^r$ for the $\mathbf{L}$-unsatisfiability of A, but that A is $\mathbf{L}$-satisfiable. Then there is an $\mathbf{L}$-model $\mathbf{M}$ of A and by Lemma 5.4 there is a standard $\mathbf{L}$-interpretation of $\mathcal{T}^0$. Lemmata 5.5, 5.6 and 5.7 imply that satisfiability by standard $\mathbf{L}$-interpretations is preserved in tableau proofs. Hence the tableau $\mathcal{T}^r$ is satisfied by a standard $\mathbf{L}$-interpretation as well. But by definition of a tableau proof, all branches in $\mathcal{T}^r$ are marked as closed, thus the tableau $\mathcal{T}^r$ cannot possibly be $\mathbf{L}$-satisfiable. □

Note that if a tableau is $\mathbf{L}$-satisfiable then it is satisfied by a *standard* $\mathbf{L}$-interpretation. Thus, $\mathbf{L}$-satisfiability is preserved in general; it is, however, quite difficult (if not impossible) to prove this directly without using the notion of standard interpretations.

6. Completeness Proof

We now turn to the completeness of our calculus. The completeness theorem can be stated in two contraposing ways (let A be a formula in NNF): "If A is **L**-unsatisfiable, then there is a tableau proof $\mathcal{T}^0, \ldots, \mathcal{T}^r$ for $\emptyset : \emptyset : 1 : A$." or equivalently "If there is no tableau proof for $\emptyset : \emptyset : 1 : A$ then A is **L**-satisfiable."

We prove the completeness theorem as stated in the paper viz: Let Ψ be a fair tableau procedure, and let A be an **L**-unsatisfiable formula in NNF. Then there is a (finite) tableau proof $\mathcal{T}^0, \ldots, \mathcal{T}^r$ for the **L**-unsatisfiability of A, where $\mathcal{T}^i$ is constructed from $\mathcal{T}^{i-1}$ $(1 \leq i \leq r)$ by

– applying the appropriate **L**-expansion rule to the branch $\mathcal{B}$ and the formula ψ on $\mathcal{B}$ chosen by Ψ from $\mathcal{T}^{i-1}$; or

– applying a most general substitution such that the **L**-closure rule can be applied to a previously open branch in $\mathcal{T}^{i-1}$.

So suppose we are given a fair tableau procedure Ψ and an initial tableau $\emptyset : \emptyset : 1 : A$. We prove the theorem in a rather roundabout way following the method of Beckert and Posegga [7]:

Step 1: We define the notion of an **L**-Hintikka set of ground tableau formulae (Def. 6.1) and show that every Hintikka set is satisfied by some **L**-interpretation (Lemma 6.2).

Step 2: Consider the sequence $(\mathcal{T}_n)_{n \geq 0}$ deterministically constructed by the fair tableau procedure Ψ without closing branches or applying substitutions, and define the infinite tableau $\mathcal{T}_\infty$ to be the limit of the $\mathcal{T}_n$.

Assuming that *no* substitution of the variables in $\mathcal{T}_\infty$ gives a closed instance of $\mathcal{T}_\infty$, we define a particular substitution θ_∞ and show that $\mathcal{T}_\infty\theta_\infty$ contains at least one branch that forms a Hintikka set (Lemma 6.3). Step 1 then gives an **L**-satisfiable set, and in particular an **L**-interpretation satisfying the formula A in root $\emptyset : \emptyset : 1 : A$ of $\mathcal{T}_\infty\theta_\infty$. Consequently, if there is no substitution that closes $\mathcal{T}_\infty$ then A is **L**-satisfiable (Lemma 6.4).

Step 3: Contraposing Lemma 6.4 gives: If A is **L**-unsatisfiable, there is some substitution θ that, when applied, allows to close all branches in $\mathcal{T}_\infty$. Thus, for some $n \in \mathbb{N}$, all branches in the finite tableau $\mathcal{T}_n\theta$ can be closed.

Step 4: To conclude the completeness proof, we show that if $\mathcal{T}_n\theta$ can be closed, then the substitution θ can be decomposed so that: $\theta = \theta' \circ \xi_r \circ \xi_{r-1} \circ \ldots \circ \xi_1$ where ξ_i is a most general closing substitution for the instantiation $(\mathcal{B}_i)\xi_1\xi_2 \ldots \xi_{i-1}$ of the i-th branch $\mathcal{B}_i$ in $\mathcal{T}_n$. (And θ' is the part of θ that is not actually needed to close $\mathcal{T}'$.) Thus the tableau $\mathcal{T}_n$ constructed using the fair procedure $\mathcal{R}$ can be closed by r applications of the substitution and the closure rule.

6.1. STEP 1: HINTIKKA SETS

DEFINITION 6.1. A set $\mathcal{X}$ of ground tableau formulae is an **L**-Hintikka set, if it satisfies the following conditions:

1. $lab(\mathcal{X})$ is a strongly generated set of labels with root 1.
2. There is no primitive proposition p such that (a) $X:\Delta_1:\sigma:p$ and $Y:\Delta_2:\sigma:\neg p$ are in $\mathcal{X}$, and (b) the logic **L** is serial or all labels in $\{\sigma\} \cup \Delta_1 \cup \Delta_2$ are justified in $\mathcal{X}$.
3. If $X:\Delta:\sigma:A \wedge B \in \mathcal{X}$ then $X:\Delta:\sigma:A \in \mathcal{X}$ and $X:\Delta:\sigma:B \in \mathcal{X}$.
4. If $X:\Delta:\sigma:A \vee B \in \mathcal{X}$ then $X:\Delta:\sigma:A \in \mathcal{X}$ or $X:\Delta:\sigma:B \in \mathcal{X}$.
5. If $X:\Delta:\sigma:\Box A \in \mathcal{X}$, then the following conditions have to be satisfied for the logic **L** as determined by Table III:

   (K) condition: $X \cup \{n\}:\Delta:\sigma.(n):A \in \mathcal{X}$ for every $n \in \mathbb{N}$;

   (4) condition: $X \cup \{n\}:\Delta:\sigma.(n):\Box A \in \mathcal{X}$ for every $n \in \mathbb{N}$ (for **K5** and **KD5** only if $\sigma = \tau.l$);

   (4^r) condition: if $\sigma = \tau.l$ then $X:\Delta \cup \{\sigma\}:\tau:\Box A \in \mathcal{X}$;

   (T) condition: $X:\Delta:\sigma:A \in \mathcal{X}$;

   (B) condition: if $\sigma = \tau.l$ then $X:\Delta \cup \{\sigma\}:\tau:A \in \mathcal{X}$;

   (5) condition: if $\sigma = 1.l$ then $X \cup \{n\}:\Delta \cup \{\sigma\}:1.(n):\Box A \in \mathcal{X}$.

6. If $X:\Delta:\sigma:\Diamond A \in \mathcal{X}$ then $X:\Delta:\sigma.n:A \in \mathcal{X}$ for some $n \in \mathbb{N}$.

LEMMA 6.2. *Every* **L**-*Hintikka set* $\mathcal{X}$ *is satisfied by some* **L**-*interpretation* $\langle \mathbf{M}, \mathbf{I} \rangle$.

PROOF. We define the **L**-model $\mathbf{M} = \langle W, R, V \rangle$ as follows. Put $W = \{[\sigma] \mid \sigma \in \mathcal{L}\}$ if **L** is serial and put $W = \{[\sigma] \mid \sigma \in lab(\mathcal{X}), \sigma \text{ is justified in } \mathcal{X}\}$ if **L** is not serial, where $[\sigma]$ is the equivalence class of all labels that are identical to σ up to (conditional) parentheses. For all $[\sigma], [\tau] \in W$, let $[\sigma]R[\tau]$ iff $\sigma \rhd \tau$: that is, iff τ is **L**-accessible from σ (see Table II). For each primitive proposition p let $V(p)$ be defined by: If **L** is serial, then $V(p) = \{[\sigma] \mid X:\Delta:\sigma:p \in \mathcal{X}\}$. Otherwise, if **L** is not serial, then $V(p) = \{[\sigma] \mid X:\Delta:\sigma:p \in \mathcal{X}, \mathbf{I}(\sigma) \in W, \mathbf{I}(\xi) \in W \text{ for all } \xi \in \Delta\}$. The (identity) interpretation **I** is defined by: $\mathbf{I}(\sigma) = [\sigma]$ if $[\sigma] \in W$ and $\mathbf{I}(\sigma) = \perp$ otherwise.

Lemma 3.5 then implies that **M** is an **L**-model. According to its construction $\langle \mathbf{M}, \mathbf{I} \rangle$ is, therefore, an **L**-interpretation.

We show by induction on the degree of tableau formulae $\phi = X:\Delta:\sigma:A$ that if $\phi \in \mathcal{X}$ then (a) $[\sigma] \notin W$, or (b) $[\xi] \notin W$ for some $\xi \in \Delta$, or (c) $[\sigma] \models A$. This induction hypothesis implies that $\langle \mathbf{M}, \mathbf{I} \rangle$ satisfies ϕ. Thus $\mathcal{X}$ itself is satisfied by the $\mathbf{L}$-interpretation $\langle \mathbf{M}, \mathbf{I} \rangle$.

Let the degree of a formula A be defined syntactically (as usual). The degree *deg* of a tableau formula is then defined by: $deg(X:\Delta:\sigma:A) < deg(X':\Delta':\sigma':A')$ iff (a) $deg(A) < deg(A')$ or (b) $deg(A) = deg(A')$ and $|\sigma| < |\sigma'|$.

Base case: Thus the traditional part A of ϕ is a literal. In case $A = p$, $[\sigma] \in W$, and $[\xi] \in W$ for all $\xi \in \Delta$, we have $[\sigma] \models p$ by the definition of V. In case $A = \neg p$, we must show that if $[\sigma] \in W$ and $[\xi] \in W$ for all $\xi \in \Delta$ then $[\sigma] \models \neg p$. That is, $[\sigma] \notin V(p)$. For a contradiction suppose that $[\sigma] \in V(p)$. By the definition of V there has to be a tableau formula $X':\Delta':\sigma':p \in \mathcal{X}$ and, in addition, if $\mathbf{L}$ is not serial then $[\sigma'] \in W$ and $[\xi'] \in W$ for all $\xi' \in \Delta'$. By definition, if $\mathbf{L}$ is not serial, $[\sigma'] \in W$ iff σ' is justified in $\mathcal{X}$. Thus we have two complementary and "completely justified" atomic formulae in $\mathcal{X}$; contradicting condition 2 in the definition of Hintikka sets.

The induction step depends on the form of the formula A in ϕ.

$A = B \wedge C$: According to condition 3 in the definition of Hintikka sets, there are formulae $X:\Delta:\sigma:B \in \mathcal{X}$ and $X:\Delta:\sigma:C \in \mathcal{X}$. The induction hypothesis applies to these formulae. Therefore, (a) $[\sigma] \notin W$, or (b) $[\xi] \notin W$ for some $\xi \in \Delta$, or $[\sigma] \models B$ and $[\sigma] \models C$ which implies $[\sigma] \models B \wedge C$.

$A = B \vee C$: Similar to the case $A = B \wedge C$.

$A = \Box B$: Suppose (a) $[\sigma] \in W$, and (b) $[\xi] \in W$ for all $\xi \in \Delta$; we then have to prove that $[\tau] \models B$ for all $[\tau] \in W$ such that $\sigma \rhd \tau$. We first show that (certain combinations of) the sub-conditions laid out as part of condition 5 of the definition of Hintikka sets imply this property for certain $[\tau] \in W$:

(K) *condition:* for all τ of the form $\tau = \sigma.[n]$ where $n \in \mathbb{N}$. Proof: By definition of Hintikka sets, there is some $X':\Delta:\sigma.(n):B \in \mathcal{X}$ for every $n \in \mathbb{N}$, to which the induction hypothesis applies. Thus, if $[\tau] \in W$, then $[\tau] \models B$.

(K) *and* (4) *conditions:* for all τ of the form $\tau_k = \sigma.n_1 \ldots n_k$ where $k \geq 1$ and $n_1, \ldots, n_k \in \mathbb{N}$ (for logics **K5** and **KD5** only provided that $|\sigma| \geq 2$). Proof: We use an induction on k. We show, that for all $k \geq 0$ (with $\tau_0 = \sigma$) there is a formula $X_k:\Delta:\tau_k:\Box B \in \mathcal{X}$ (for some X_k). Using the same argument as above for the (K) condition, this implies $[\tau_k] \models B$ if $[\tau_k] \in W$ for all $k \geq 1$. For $k = 0$ we have $X_0:\Delta:\tau_0:\Box B = X:\Delta:\sigma:A \in \mathcal{X}$ by assumption. Induction step: $X_k:\Delta:\tau_k:\Box B \in \mathcal{X}$ implies $X_{k+1}:\Delta:\tau_{k+1}:\Box B \in \mathcal{X}$ (with $X_{k+1} = X_k \cup \{n_k\}$), by the (4) condition.

(T) *condition:* for $\tau = \sigma$. Proof: There has to be a formula $X:\Delta:\sigma:B \in \mathcal{X}$ that the induction hypothesis applies to. Thus, if $[\sigma] \in W$, then $[\sigma] \models B$.

(B) *condition:* for τ s.t. $\sigma = \tau.l$. Proof: There has to be a formula $X:\Delta \cup \{\sigma\}:\tau:B$ in $\mathcal{X}$ that the induction hypothesis applies to. Thus, if $[\tau] \in W$, then $[\tau] \models B$.

(K), (4), (4^r), (5) *conditions*: for all τ such that $|\tau| \geq 2$ in case $|\sigma| \geq 2$, and for all τ of the form $1.n$ where $n \in \mathbb{N}$ otherwise (i.e., in case $\sigma = 1$). Proof: If $|\sigma| \geq 2$, we prove by induction on the length of τ using the (4^r) condition that for all $\tau \in ipr(\sigma)$ there is a formula $X:\Delta:\tau:\Box B \in \mathcal{X}$. Using the same argument as above for the (K) condition this implies $[\tau] \models B$ for all τ such that $|\tau| = 2$ and $[\tau] \in W$. In addition, we have $X : \Delta : 1.m : \Box B \in \mathcal{X}$ for some $m \in \mathbb{N}$ (where $1.m \in ipr(\sigma)$); thus the (5) condition implies that $X \cup \{n\} : \Delta \cup \sigma : 1.(n) : \Box B \in \mathcal{X}$ for all $n \in \mathbb{N}$. Now, we can prove $[\tau] \models B$ for all τ such that $|\tau| \geq 3$ and $[\tau] \in W$ as in the case of the (K) and (4) conditions. If $\sigma = 1$, we can derive $[\sigma.n] \models B$ for all $n \in \mathbb{N}$ and $[\sigma.n] \in W$ using the (K) condition (see above).

(K), (4), (4^r) *conditions:* for all τ such that $|\tau| \geq 2$. Proof: We prove by induction on the length of τ using the (4^r) condition that for all $\tau \in ipr(\sigma)$ there is a formula $X : \Delta : \tau : \Box B \in \mathcal{X}$. In particular, we have $X : \Delta : 1 : \Box B \in \mathcal{X}$. Now, we can proceed to prove $[\tau] \models B$ for all τ such that $|\tau| \geq 2$ and $[\tau] \in W$ as in the case of the (K) and (4) conditions.

(K), (T), (4), (4^r) *conditions:* for all τ. Proof: Similar to the case of conditions (K), (4) and (4^r) we prove by induction that for all τ such that $[\tau] \in W$ there is a formula $X : \Delta : \tau : \Box B \in \mathcal{X}$, which then, using the same argument as above for the (T) condition implies $[\tau] \models B$.

By checking Table II, it is obvious that the sub-conditions for box-formulae that apply to an **L**-Hintikka set imply: if $[\tau] \in W$ and $\sigma \rhd \tau$, then $[\tau] \models B$.

$A = \Diamond B$: According to condition 6 in the definition of Hintikka sets, there is a formula $X : \Delta : \sigma.n : B \in \mathcal{X}$. The induction hypothesis applies to this formula. Therefore, (a) $[\sigma.n] \notin W$ or (b) $[\xi] \notin W$ for some $\xi \in \Delta$; or $[\sigma.n] \models B$. Now, since the label $\sigma.n$ itself occurs in $lab(\mathcal{X})$, $[\sigma.n] \notin W$ implies that $\sigma.n$ is not justified in $\mathcal{X}$. Since the last position of $\sigma.n$ is unconditional this implies that σ is not justified in $\mathcal{X}$. Hence $[\sigma] \notin W$. Furthermore, if $[\sigma.n] \models B$ then $[\sigma] \models \Diamond B$ since $\sigma \rhd \sigma.n$ for all **L**. Together, we have enough to prove that (a) $[\sigma] \notin W$ or (b) $[\xi] \notin W$ for some $\xi \in \Delta$ or (c) $[\sigma] \models \Diamond B$ as desired. □

6.2. Step 2: Considering an Infinite Tableaux

Let $\mathcal{T}_1$, $\mathcal{T}_2$, etc. be the sequence of tableaux deterministically constructed using $\mathbf{\Psi}$ without closing branches or applying a substitution. These tableaux approximate the infinite tree $\mathcal{T}_\infty$.

Now suppose that no substitution allows $\mathcal{T}_\infty$ to be closed. Thus we can choose any substitution θ, and we are guaranteed that $\mathcal{T}_\infty\theta$ will contain some open branch. The branch may differ according to the choice of θ.

We now define a particular substitution θ_∞ as follows: Let $\{\mathcal{B}_1, \mathcal{B}_2, \ldots\}$ be an enumeration of all the branches of $\mathcal{T}_\infty$. Let $\Phi = \{\phi_1, \phi_2, \ldots\}$ be an enumeration of the disjunctive formulae (formulae of the form $X_i : \Delta_i : \sigma_i : B_i \vee C_i$) in $\mathcal{T}$ *excluding renamings*. For every disjunctive formula, if ϕ_i occurs on $\mathcal{B}_k$ then let ϕ_{ijk} be the j-th renaming of ϕ_i on the k-th branch.

The label σ_i of ϕ_i will be of finite length. Thus, the set of all ground instances of σ_i is enumerable: let $\{\sigma_i^1, \sigma_i^2, \ldots\}$ be such an enumeration.

Let x be a variable in X_{ijk} and suppose it occurs in the p-th position of σ_{ijk}. Note that the sets X_{ijk} of universal variables are all pairwise disjoint (even if only one of the formulae in the numerator of the conjunctive rule is renamed). To ensure that the k-th branch $\mathcal{B}_k$ is a potential source of an **L**-model we ensure that the occurrences of ϕ_i on $\mathcal{B}_k$ "cover" all the instances of σ_i. So choose θ_∞ so that $\theta_\infty(x) = n$, where n is the value of the p-th position of σ_i^j. Thus if x is in the p-th position of σ_{i1k}, its value "covers" σ_i^1, if it is in the p-th position of σ_{i2k}, its value "covers" σ_i^2, and so on.

LEMMA 6.3. *If θ_∞ is defined as above, and $\mathcal{B}$ is an open branch of the tableau $\mathcal{T}_\infty$ that cannot be closed when θ_∞ is applied then*

$$\mathcal{X} = \{\phi\lambda_{|X}\theta_\infty \mid \phi = X:\Delta:\sigma:A \textit{ is a formula on } \mathcal{B}, \textit{ and } \\ \lambda \textit{ is a grounding substitution}\}$$

is an **L**-*Hintikka set.*

PROOF. We have to check each clause of Definition 6.1 for the set $\mathcal{X}$.

Condition 1. It is obvious that the root is 1, and fairly easy to see that we always produce a strongly generated set.

Condition 2. Suppose this condition is violated by $\mathcal{X}$. Then there have to be formulae $\phi_1 = X_1:\Delta_1:\sigma_1:p$, $\phi_2 = X_2:\Delta_2:\sigma_2:\neg p$ on $\mathcal{B}$ and grounding substitutions λ_1 and λ_2, such that $\sigma_1\lambda_{1|X_1}\theta_\infty = \sigma_2\lambda_{2|X_2}\theta_\infty = \zeta$, and (b) the logic **L** is serial or all labels in $\{\zeta\} \cup \Delta_1\lambda_{1|X_1}\theta_\infty \cup \Delta_2\lambda_{2|X_2}\theta_\infty$ are justified in $\mathcal{X}$. Our expansion rules always use $\mathcal{T}$-renamings of universal variables in their numerator, hence $X_1 \cap X_2 = \emptyset$. Therefore, there is a single grounding substitution λ of the universal variables in $\mathcal{T}_\infty$, such that $\sigma_1\lambda = \sigma_1\lambda_{1|X_1}$ and $\sigma_2\lambda = \sigma_2\lambda_{2|X_2}$. In addition, $\lambda \circ \theta_\infty = \theta_\infty \circ \lambda$, since θ_∞ only instantiated free variables in $\mathcal{T}_\infty$. Thus the branch $\mathcal{B}$ *can* be closed using the substitution λ of the universal variables in $\mathcal{T}_\infty$, which contradicts the choice of $\mathcal{B}$.

Condition 3. We must show that for all $\phi = X:\Delta:\sigma:A \wedge B \in \mathcal{B}$ and all grounding substitutions λ, and thus for all formulae of the form $\phi\lambda_{|X}\theta_\infty$, the formulae $\phi_1\lambda_{|X}\theta_\infty$ and $\phi_2\lambda_{|X}\theta_\infty$ are in $\mathcal{X}$, where $\phi_1 = \emptyset:\Delta:\sigma:A$ and $\phi_2 = \emptyset:\Delta:\sigma:B$. Since the appropriate rule has been applied to ϕ, the formulae $X:\Delta:\sigma:A$ and $X':\Delta':\sigma':B$ are both on $\mathcal{B}$. Thus, $\phi_1\lambda_{|X}\theta_\infty \in \mathcal{X}$. For ϕ_2, let μ be the substitution renaming the variables in X such that $X':\Delta':\sigma':B = (X:\Delta:\sigma:B)\mu$, and put $\lambda' = (\lambda \circ \mu)$, which implies $\lambda'_{|X'} = \lambda_{|X} \circ \mu$. The substitution λ' is grounding. Therefore, by definition the set $\mathcal{X}$ contains the formula $\phi_2'\lambda'_{|X'}\theta_\infty$, which is identical to $\phi_2\lambda_{|X}\theta_\infty$.

Condition 4. We have to show, that for all $\phi = X:\Delta:\sigma:A \vee B \in \mathcal{B}$ and all grounding substitutions λ, and thus for all formulae of the form $\phi\lambda_{|X}\theta_\infty$, one of the formulae $\phi_1\lambda_{|X}\theta_\infty$ and $\phi_2\lambda_{|X}\theta_\infty$ is in $\mathcal{X}$, where $\phi_1 = \emptyset:\Delta:\sigma:A$ and $\phi_2 = \emptyset:\Delta:\sigma:B$. If $X = \emptyset$ then this holds immediately by the special case of the disjunctive rule. Otherwise, according to the construction of $\mathcal{T}_\infty$ and θ_∞, there has to be a renaming

$\phi' = \phi\nu$ of ϕ on $\mathcal{B}$ (where ν is the renaming substitution) such that $\theta_{\infty|X'} \circ \nu = \lambda_{|X}$, i.e., $\phi'\theta_\infty = \phi\lambda_{|X}\theta_\infty$. Since the appropriate rule has been applied to ϕ', one of the formulae $\phi'_1 = \phi_1\nu$ and $\phi'_2 = \phi_2\nu$ is on $\mathcal{B}$ and thus $\phi'_1\lambda_{|\emptyset}\theta_\infty = \phi'_1\theta_\infty = \phi_1\lambda_{|X}\theta_\infty$ or $\phi'_2\lambda_{|\emptyset}\theta_\infty = \phi_2\lambda_{|X}\theta_\infty$ is in $\mathcal{X}$.

Condition 5 and 6. Since these conditions closely resemble the tableau expansion rules, the proof that they hold for $\mathcal{X}$ is similar to that for condition 3 (conjunctive formulae). □

LEMMA 6.4. *If there is no substitution* θ *such that all branches of* $\mathcal{T}_\infty\theta$ *can be closed, then the input formula for which the tableau sequence has been constructed is* **L**-*satisfiable.*

PROOF. Since $\mathcal{T}_\infty$ cannot be closed using any substitution, it cannot be closed by applying θ_∞ as defined above. Thus there is some open branch in $\mathcal{T}_\infty\theta_\infty$. By Lemma 6.3 this branch forms an **L**-Hintikka set. By Lemma 6.2 such a set gives an **L**-interpretation $\langle \mathbf{M}, \mathbf{I} \rangle$ that satisfies the root $\emptyset:\emptyset:1:A$ of $\mathcal{T}^0$. But this means that in the **L**-model **M** we must have $\mathbf{I}(1) \models A$. □

6.3. STEP 3: CONSTRUCTING A FINITE CLOSED TABLEAUX

Contraposing Lemma 6.4 gives: If A is **L**-unsatisfiable, then there is at least one substitution θ that, when applied, allows to close all branches in $\mathcal{T}_\infty$.

LEMMA 6.5. *If there is a substitution* θ *such that all branches of* $\mathcal{T}_\infty\theta$ *can be closed, then there is an* $n \in N$ *such that all branches in the finite tableau* $\mathcal{T}_n\theta$ *can be closed.*

PROOF. According to König's Lemma, the tableau $\mathcal{T}$ that results from removing from $\mathcal{T}_\infty\theta$ all formulae that are not needed to close one of the branches on which they occur (that includes justification) is *finite*. Thus, there is a finite $n \in N$ such that $\mathcal{T}$ is an initial subtree of $\mathcal{T}_n\theta$.

6.4. STEP 4: DECOMPOSITION OF THE CLOSING SUBSTITUTION

LEMMA 6.6. *If* $\mathcal{T}_n'\theta$ *is closed for some finite n, then the substitution* θ *can be decomposed so that:* $\theta = \theta' \circ \xi_r \circ \xi_{r-1} \circ \ldots \circ \xi_1$ *where* ξ_i *is a most general closing substitution for the instantiation* $(\mathcal{B}_i)\xi_1\xi_2\ldots\xi_{i-1}$ *of the i-th branch,* $\mathcal{B}_i$*, in* $\mathcal{T}_n'$*. And* θ' *is the part of* θ *that is not actually needed to close* $\mathcal{T}_n'$.

PROOF. We construct the ξ_i inductively as follows: Define $\xi'_0 = \theta$. For $1 \leq i \leq r$, let ξ_i be a most general substitution, such that (a) ξ'_{i-1} is a specialisation of ξ_i; that is, there is a substitution ξ'_i such that $\xi'_{i-1} = \xi'_i \circ \xi_i$; and (b) ξ_i is a closing substitution for $(\mathcal{B}_i)\xi_1\xi_2\ldots\xi_{i-1}$.

Then ξ_i is a most general *closing* substitution. For otherwise, there must be a closing substitution ξ_i'', that is more general than ξ_i. The is-more-general relation is transitive hence ξ_i'' is more general than ξ_{i-1}', which contradicts our choice of ξ_i as a most general substitution satisfying the two conditions.

Finally, define $\theta' = \xi_r'$. □

We can now conclude the proof of the completeness theorem (Theorem 4.9) as follows:

The formula A is **L**-unsatisfiable and $\mathcal{T}_\infty$ is constructed using a fair proof procedure so there is a substitution θ such that all branches of $\mathcal{T}_\infty\theta$ can be closed (Lemma 6.4). Then there is a finite tableau $\mathcal{T}_n$ such that all branches of $\mathcal{T}_n\theta$ can be closed (Lemma 6.5). The tableau $\mathcal{T}_n$ can be constructed with n expansion rule applications using the procedure $\mathcal{R}$. Since θ can be decomposed into r most general closing substitutions (Lemma 6.6), a closed tableau can be constructed from $\mathcal{T}_n$ by r applications of the substitution and closure rules satisfying the conditions of the completeness theorem.

7. Conclusion and Future Work

The advantages of free variables. We believe that labels with variables deliver the following advantages:

– The use of variables generates a smaller search space since a label can now stand in for all its ground instances. This is in stark contrast to the modular systems of [22, 16], where only ground labels are used.

– The use of a Gödelisation function in the diamond-rule leads to a smaller number of labels than in other labelled tableau methods since two different occurrences of the formula $X:\Delta:\sigma:\Diamond A$ lead to the same formula $X:\Delta:\sigma.\lceil A\rceil:A$. We therefore do not need to delete duplicate occurrences of a formula as is done in some tableau implementations for modal logics. This is particularly important since the world $\sigma.\lceil A\rceil$ may be the root of a large sub-model and duplicating it is likely to be extremely inefficient.

Experiences with a Lean Implementation. Tableau-based theorem provers developed during the last decade for first-order logic have been complex and highly sophisticated, typified by systems like Setheo [21] and $_3T^AP$ [6]. On the other hand, free-variable tableaux, and their extensions like universal-variable tableaux, have been used successfully for *lean* Prolog implementations, as typified by leanT^AP [7] and `ileanTAP` [25]. A “lean” implementation is an extremely compact program that exploits Prolog’s built-in clause indexing scheme and backtracking mechanisms instead of relying on elaborate heuristics. Such compact lean provers are much easier to understand than their more complex stablemates, and hence easier to adapt to special needs.

We have implemented our calculus as a "lean" theorem prover written in Prolog (the source code is available at `i12www.ira.uka.de/~beckert/modlean` on the *World Wide Web*). It makes extensive use of Prolog features like unification and backtracking.[6] The basic version for the logic **K** is called lean**K**, and consists of just eleven Prolog clauses and 45 lines of code. The version for the logic **KD** which does not demand justified labels, is even shorter: it consists of only 6 clauses and 27 lines of code.

Our initial results with this lean implementation are encouraging. However, as a comparison with state-of-the-art theorem provers for modal logics [1] has shown, modal logics differ from first-order logic: the free-variable technique is not enough. Because of the propositional flavour of modal logics, techniques such as simplification [23] are needed to implement an efficient prover.

We have developed the theory of how to obtain modular proof systems based upon free-variable tableaux for modal logics; but it is future work to validate the usefulness of the free-variable technique in the modal framework by combining it with other techniques.

Future work. Our method is really a very clever translation of propositional modal logics into first-order logic, and most of the complications arise because some worlds may have no successors. The new notion of conditional labels allows us to keep track of these complications, and thus handle the non-serial logics that frustrate other "general frameworks" [14, 19]. Nevertheless, our method can also handle *second-order* "provability" logics like **G** and **Grz**; see [16]. Furthermore, specialised versions of these tableau systems can match the theoretical lower bounds for particular logics like **K45**, **G** and **Grz** if we give up modularity; see [16, 22]. We intend to extend our initial implementation of lean**K** along these lines.

The 15 basic normal modal logics are decidable and techniques from [12, 16, 22, 18] can be used to extend our method into a decision procedure (we have reported first results in [3, 4]).

Fitting [11] shows how to view the original leanT^AP program for classical propositional logic as an unusual sequent calculus `dirseq`. He also shows how to extend `dirseq` to handle the modal logics **K**, **KT**, **K4**, and **S4**. As with traditional modal tableaux, however, `dirseq` does not handle the symmetric logics like **S5** and **B**. Our work can be extended to give a modular free variable version of `dirseq` that does handle these logics.

It is also possible to extend our method to deal with the notions of global and local logical consequence [12].

[6] When the prover is ported to other languages, the Prolog mechanisms may have to be implemented; but even then the advantage of lean provers remains, namely that (the main part of) of the prover is compact and thus easy to understand and to adapt to special needs.

An alternative approach [17] uses different unification algorithms to find complementary labelled literals for branch closure. The interactions between modalities, variable labels, and unification algorithms, however, is by no means easy to disentangle. Extending our method to utilise special unification algorithms is perfectly possible, now that correctness and completeness have been proved for the interactions between modalities and variable labels.

Acknowledgements

We thank the anonymous referees and Nicolette Bonnette for useful comments on earlier versions of this paper.

References

1. Peter Balsinger and Alan Heuerding. Comparison of theorem provers for modal logics: Introduction and summary. In *Proceedings, International Conference on Theorem Proving with Analytic Tableaux and Related Methods, Oisterwijk, The Netherlands*, LNCS 1397. Springer, 1998.
2. Bernhard Beckert and Rajeev Goré. Free variable tableaux for propositional modal logics. In *Proceedings, International Conference on Theorem Proving with Analytic Tableaux and Related Methods, Pont-à-Mousson, France*, LNCS 1227, pages 91–106. Springer, 1997.
3. Bernhard Beckert and Rajeev Goré. leanK 2.0: Description for the comparison of theorem provers for modal logics. In *Proceedings, International Conference on Theorem Proving with Analytic Tableaux and Related Methods, Oisterwijk, The Netherlands*, LNCS 1397, pages 33–34. Springer, 1998.
4. Bernhard Beckert and Rajeev Goré. System description: leanK 2.0. In *Proceedings, 15th International Conference on Automated Deduction (CADE), Lindau. Germany*, LNCS 1421, pages 51–55. Springer, 1998.
5. Bernhard Beckert and Reiner Hähnle. Analytic tableaux. In Wolfgang Bibel and Peter H. Schmitt, editors, *Automated Deduction — A Basis for Applications*, volume I: Foundations. Kluwer, Dordrecht, 1998.
6. Bernhard Beckert, Reiner Hähnle, Peter Oel, and Martin Sulzmann. The tableau-based theorem prover $_3T^AP$, version 4.0. In *Proceedings, 13th International Conference on Automated Deduction (CADE), New Brunswick, NJ, USA*, LNCS 1104, pages 303–307. Springer, 1996.
7. Bernhard Beckert and Joachim Posegga. leanT^AP: Lean tableau-based deduction. *Journal of Automated Reasoning*, 15(3):339–358, 1995.
8. E. Bencivenga. Free logic. In D. Gabbay and F. Günthner, editors, *Handbook of Philosophical Logic*, volume 3. Kluwer, Dordrecht, 1986.
9. Nicolette Bonnette. K_t^{FV}: A free variable sequent system for tense logic Kt. Australian Computational Logic Workshop, February 2000.
10. Marcello D'Agostino, Dov Gabbay, and Alessandra Russo. Grafting modalities onto substructural implication systems. *Studia Logica*, 59:65–102, 1997.
11. Melvin Fitting. leanTAP revisited. *Journal of Logic and Computation*, 8(1):33–47, 1998.
12. Melvin C. Fitting. *Proof Methods for Modal and Intuitionistic Logics*, volume 169 of *Synthese Library*. D. Reidel, Dordrecht, Holland, 1983.

13. Melvin C. Fitting. *First-Order Logic and Automated Theorem Proving*. Springer, second edition, 1996.
14. Alan Frisch and Richard Scherl. A general framework for modal deduction. In J. Allen, R. Fikes, and E. Sandewall, editors, *Proceedings, 2nd Conference on Principles of Knowledge Representation and Reasoning*. Morgan-Kaufmann, 1991.
15. Dov Gabbay. *Labelled Deductive Systems*. Oxford University Press, 1996.
16. Rajeev Goré. Tableau methods for modal and temporal logics. In Marcello D'Agostino, Dov Gabbay, Reiner Hähnle, and Joachim Posegga, editors, *Handbook of Tableau Methods*. Kluwer, Dordrecht, 1999.
17. Guido Governatori. A reduplication and loop checking free proof system for S4. In *Short Papers: TABLEAUX'96*, number 154-96 in RI-DSI, Via Comelico 39, 20135 Milan, Italy, 1996. Department of Computer Science, University of Milan.
18. Alain Heuerding, Michael Seyfried, and Heinrich Zimmermann. Efficient loop-check for backward proof search in some non-classical logics. In P. Miglioli, U. Moscato, D. Mundici, and M. Ornaghi, editors, *Proceedings, 5th Workshop on Theorem Proving with Analytic Tableaux and Related Methods, Terrasini, Palermo, Italy*, LNCS 1071, pages 210–225. Springer, 1996.
19. Peter Jackson and Hans Reichgelt. A general proof method for first-order modal logic. In *Proceedings, International Joint Conference on Artificial Intelligence (IJCAI)*, pages 942–944, 1987.
20. S. Kanger. *Provability in Logic*. Stockholm Studies in Philosophy, University of Stockholm. Almqvist and Wiksell, Sweden, 1957.
21. Reinhold Letz, Johann Schumann, Stephan Bayerl, and Wolfgang Bibel. SETHEO: A high-performance theorem prover. *Journal of Automated Reasoning*, 8(2):183–212, 1992.
22. Fabio Massacci. Strongly analytic tableaux for normal modal logics. In A. Bundy, editor, *Proceedings, 12th International Conference on Automated Deduction (CADE), Nancy, France*, LNCS 814, pages 723–737. Springer, 1994.
23. Fabio Massaci. Simplification: A general constraint propagation technique for propositional and modal tableaux. In *Proceedings, International Conference on Theorem Proving with Analytic Tableaux and Related Methods, Oisterwijk, The Netherlands*, LNCS 1397. Springer, 1998.
24. Grigori Mints. *A Short Introduction to Modal Logic*. CSLI, Stanford, 1992.
25. Jens Otten. `ileanTAP`: An intuitionistic theorem prover. In *Proceedings, International Conference on Theorem Proving with Analytic Tableaux and Related Methods, Pont-à-Mousson, France*, LNCS 1227. Springer, 1997.
26. Jeremy Pitt and Jim Cunningham. Distributed modal theorem proving with KE. In M. Miglioli, U. Moscato, D. Mundici, and M. Ornaghi, editors, *Proceedings, International Conference on Theorem Proving with Analytic Tableaux and Related Methods, Terrasini, Palermo, Italy*, LNCS 1071, pages 160–176. Springer, 1996.
27. Steve V. Reeves. Semantic tableaux as a framework for automated theorem-proving. In C. Mellish and J. Hallam, editors, *Advances in Artificial Intelligence (Proceedings of AISB-87)*, pages 125–139. Wiley, 1987.
28. Alessandra Russo. Generalising propositional modal logic using labelled deductive systems. In F. Baader and K. Schulz, editors, *Proceedings, Frontiers of Combining Systems (FroCoS), Munich, Germany*, volume 3 of *Applied Logic Series*. Kluwer, Dordrecht, 1996.
29. Lincoln A. Wallen. *Automated Deduction in Nonclassical Logics: Efficient Matrix Proof Methods for Modal and Intuitionistic Logics*. MIT Press, 1989.

Address for Offprints:
BERNHARD BECKERT, University of Karlsruhe, Inst. for Logic, Complexity and Deduction Systems, D-76128 Karlsruhe, Germany. Email: beckert@ira.uka.de, WWW: i12www.ira.uka.de/˜beckert.
RAJEEV GORÉ, Automated Reasoning Project, Australian National University, Canberra, ACT, 0200, Australia. Email: rpg@arp.anu.edu.au, WWW: arp.anu.edu.au/˜rpg.