tificial Intelligence, pages 723–737. © Springer-Verlag, 1994.

Strongly Analytic Tableaux for Normal Modal Logics

Fabio Massacci

Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
via Salaria 113, I-00198 Roma, Italy
E-mail: massacci@assi.dis.uniroma1.it

Abstract. A strong analytic tableau calculus is presentend for the most common normal modal logics. The method combines the advantages of both sequent-like tableaux and prefixed tableaux. Proper rules are used, instead of complex closure operations for the accessibility relation, while non determinism and cut rules, used by sequent-like tableaux, are totally eliminated. A strong completeness theorem without cut is also given for symmetric and euclidean logics. The system gains the same modularity of Hilbert-style formulations, where the addition or deletion of rules is the way to change logic. Since each rule has to consider only adjacent possible worlds, the calculus also gains efficiency. Moreover, the rules satisfy the strong Church Rosser property and can thus be fully parallelized. Termination properties and a general algorithm are devised. The propositional modal logics thus treated are K, D, T, KB, K4, K5, K45, KDB, D4, KD5, KD45, B, S4, S5, OM, OB, OK4, OS4, OM^+, OB^+, $OK4^+$, $OS4^+$. Other logics can be constructed with different combinations of the proposed rules, but are not presented here.

1 Introduction

The use of modal logic spreads across artificial intelligence and computer science, as a way of modelling knowledge and belief [10] or as a formalism to specify the behaviour of distributed systems [9]. Nonmonotonic versions are taking grounds too [14,16]. Thus, devising efficient and human-oriented proof procedures becomes a compelling task in automated deduction.

The main problem, even in the case of propositional modal logics, is that traditional proof systems, such as tableaux or sequent calculus [2,4,5,8,18], are not very efficient, whereas other proposals like resolution [17] and matrix proof methods [19] seem to need a "user-friendly" interface for the presentation of proofs and proof search. The prefixed tableaux of Fitting [3,4] appear to stand in between, yet they are more oriented towards model checking.

Beside that, a common limitation affects all these proposals: the unsatisfactory treatment of strong deducibility for euclidean (and sometimes also symmetric logics), which are now playing a major role in nonmonotonic reasoning [13,14,16]. At this point, sequent-like tableaux or sequent themselves treat these logics in an entirely non-deterministic way, since cut has been eliminated for K45 [8,18] but only for weak deducibility (i.e. without premises). Other methods do not cover them at all. Apart from Hilbert-style proof procedures, the only way to cope with them seemed to be the following: revert from automated deduction to model checking and try to construct explicitly a (counter) model; a construction that requires to calculate many times the euclidean, symmetric etc. closure of the accessibility relation [1,10].

We propose a new tableau calculus that combines the advantages of sequent-like and prefixed tableaux and allows an analytic, deterministic, effective, uniform, strongly confluent and "user friendly" approach to most normal propositional modal logics, including symmetric and euclidean ones: K, T, D, KB, K4, K5, K45, KDB, D4, KD5, KD45, KD45, T, B, S4, S5, OM, OB, OK4, OS4, OM^+, OB^+, $OK4^+$, $OS4^+$ and many others. For any of these logics, we provide a strong completeness theorem without cut, a necessary property if global assumptions are going to play the role of a knowledge base and local assumptions that of information provided by the user.

Intuitively, a tableau is a refutation, a failed attempt to build a countermodel for a given formula. The different behaviour of prefixed and sequent-like tableaux is due to the way they code information. A sequent-like tableau uses formulae, and thus "travels" across the supposed countermodel, moving formulae from one world to the next. Choosing a different accessibility relation means that different formulae may be taken along. Such a tableau considers only one world at a time and cannot look back. Hence, beside the usual nondeterminism brought by disjunction, further levels of nondeterminism are necessary to cope with alternative possible worlds or with symmetric and euclidean logics. In contrast, a prefixed tableau uses prefixes (sequence of integers) to "name" the possible worlds where a particular formula holds. It can move back but is forced to behave as a model checker and to devise a relation over prefixes that is a "carbon copy" of the accessibility relation over possible worlds. Hence a prefix must possibly access other prefixes "very far away" and the efficiency of the systems is limited.

Our proof system — we called it Single Step Tableaux (SST for short) — also uses prefixes to keep track of the worlds where formulae hold. The key difference is that the "accessibility relation" between prefixes is not used in the proof procedure, but only in the proof of correctness and completeness theorems (an idea also used for matrix proof methods, see [19]). Instead, we define rules to specify which formulae may be moved forward to the next close possible world, as sequent-like tableaux do, or backward, as they cannot do. The name "single step" has been used since the reduction of a formula, with a given prefix, makes only use of immediately accessibile prefixes, without caring of those accessible via the transitive, symmetric etc. closure.

The intuition is very close to the "visa rules", proposed for first order modal logic in Labelled Deductive Systems [6]: each logic grants a different "visa" to constants for moving from one world to another. Why can different logics not grant different visas to formulae?! In this spirit, our SST may be viewed as a particular labelled deductive system.

Moreover, going "step by step" does not reveal to be an obstacle: a "hidden" property of prefixed tableaux (a kind of proof theoretical counterpart of Generation Theorem of Segerberg [12]) is used to get completeness. Thus the method gains a lot of efficiency, since only a small part of the worlds' graph must be taken into account for each reduction. Yet, it is human oriented enough for doing proof by hand.

The SST-calculus provides other advantages: it offers the same flexibility of Hilbert-style proof methods (to change logic we just have to add or take away a rule), and may be used to obtain countermodels for non-valid formulae i.e. prefixes form a minimal spanning tree of the underlaying frame. The transitive, reflexive etc. closure of the accessibility relation may be done afterwards, if needed. Moreover, as a tableaux system, it can be used for establishing both satisfiability and validity.

Other properties of our formal system are also of great interest from the point of view of automated deduction and practical implementation. Our rules satisfy the strong Church-Rosser property and thus the calculus is strongly confluent. Hence, at any stage of the computation, the execution sequence of applicable rules may be changed in arbitrary fashion. Therefore SST can be a good basis for a flexible, i.e. compatible with many search heuristic, and highly parallelizable implementation. In practice, any formula could be assigned to a different processor, provided the processors exchange information to keep prefixes naming consistent.

The SST proof method allows us to separate the rules which treat positive and negative information and to prove some interesting properties about termination and the underlying prefixes graph structure. These properties, along with the Church-Rosser theorem, are used to devise a general and terminating algorithm (obviously in the case where global and local assumptions are finite).

The results we present may be easily used to extend other proof systems to cover the same logics. Indeed Fitting's prefixed tableaux are directly extended using the conditions over prefixes described in Table 6, in Sect. 3. Matrix proof methods and other systems may be enhanced in a similar way.

In Sect. 2 we present the SST-calculus, whereas correctness and completeness proofs are sketched in Sect. 3. Church Rosser and other termination properties are presented in Sect. 4. A general algorithm is sketched in Sect. 5. Related works and conclusions are discussed in Sect. 6.

2 The Single Step Tableaux calculus

2.1 Preliminaries

Familiarity with the usual definitions of modal language and formulae is assumed (see [12] for an introduction). In particular we use the signed version of modal formulae. Thus an *unsigned formula* is constructed from propositional letters by the usual connectives $\wedge, \vee, \rightarrow, \neg, \Box, \Diamond$ and a *signed formula* is an unsigned one prefixed by the operator T or F. Informally, signs can be interpreted as the qualifiers "is true" or "is false". In the sequel X and Y will range over unsigned formulae, Z and Q over signed ones and p over propositional letters.

Signed formulae allow us to use the uniform notation of Smullyan and Fitting that classifies signed formulae according to their signs and principal connectives, as shown in Table 1. Informally speaking, formulae with a similar semantic behaviour (see below) are classified accordingly.

Table 1. Classification of Signed Formulae

α	α_1	α_2	β	β_1	β_2	π	π_0	ν	ν_0
$T.X \wedge Y$	$T.X$	$T.Y$	$F.X \wedge Y$	$F.X$	$F.Y$	$F.\Box X$	$F.X$	$T.\Box X$	$T.X$
$F.X \vee Y$	$F.X$	$F.Y$	$T.X \vee Y$	$T.X$	$T.Y$	$T.\Diamond X$	$T.X$	$F.\Diamond X$	$F.X$
$F.X \rightarrow Y$	$T.X$	$F.Y$	$T.X \rightarrow Y$	$F.X$	$T.Y$				
$F.\neg X$	$T.X$	$T.X$	$T.\neg X$	$F.X$	$F.X$				

The semantics of modal logic is the usual one, based on *frames* i.e. pairs $\langle W, \mathcal{R} \rangle$ where W is a non empty set and $\mathcal{R}$ is a relation over W. Different modal logics may be obtained by varying $\mathcal{R}$, as shown in Table 2.

Once again, the purpose of this work is not to choose among these logics, but to develop a proof system for each alternative. Thus, to keep exposition as compact and parametric as possible, we refer to L-frames, L-models or L-tableaux etc. when referring to frames, models or tableaux etc. for a particular logic L, viz. one among those listed in Table 2.

Table 2. Conditions on Accessibility Relation

Logic	Accessibility Relation $\mathcal{R}$ on frames
K	any relation
KB	symmetric
K4	transitive
K5	euclidean
K45	transitive and euclidean
D	serial
KDB	serial and symmetric
D4	serial and transitive
KD5	serial and euclidean
KD45	serial, transitive and euclidean
T	reflexive
B	reflexive and symmetric
S4	reflexive and transitive
S5	equivalence (reflexive, symmetric and transitive)
OM	almost reflexive
OB	almost reflexive and almost symmetric
OK4	almost transitive
OS4	almost reflexive and transitive
OM^+	serial and almost reflexive
OB^+	serial, almost reflexive and almost symmetric
$OS4^+$	serial, almost reflexive and transitive

Note 1. * $\mathcal{R}$ is serial iff $\forall w \in W\, \exists w^* \in W$ s. t. $w\mathcal{R}w^*$. Almost reflexive iff $\forall w_0, w \in W$, $w_0\mathcal{R}w$ implies $w\mathcal{R}w$ and almost symmetric iff $\forall w_0, w, w^* \in W$, $w_0\mathcal{R}w$ implies $w\mathcal{R}w^* \Rightarrow w^*\mathcal{R}w$. A similar condition holds for almost transitivity.

A L-model is thus a triple $\langle W, \mathcal{R}, \Vdash\rangle$where $\langle W, \mathcal{R}\rangle$is a L-frame and $\Vdash$ is a relation between possible worlds and signed formulae such that,

$$
\begin{array}{lll}
w \Vdash T.\top & \text{and} & w \Vdash F.- \ . \\
w \Vdash \alpha & \text{iff} & w \Vdash \alpha_1 \text{ and } w \Vdash \alpha_2 \ . \\
w \Vdash \beta & \text{iff} & w \Vdash \beta_1 \text{ or } w \Vdash \beta_2 \ . \\
w \Vdash \nu & \text{iff} & \forall w^* \in W : w\mathcal{R}w^* \Rightarrow w^* \Vdash \nu_0 \ . \\
w \Vdash \pi & \text{iff} & \exists w^* \in W : w\mathcal{R}w^* \text{ and } w^* \Vdash \pi_0 \ .
\end{array}
$$

The conjugate $\overline{Q}$ of a formula Q is obtained by exchanging F with T. Clearly, for any L-model $\langle W, \mathcal{R}, \Vdash\rangle$and any $w \in W$, $w \Vdash Q$ iff $w \not\Vdash \overline{Q}$.

Validity and satisfiability of a formula in a L-model are given in the usual way, using local and global assumptions (set of signed formulae):

Definition 2. A signed formula Q is L-valid with local assumptions U and global assumptions G i.e. $G \models_{\mathrm{L}} U \Rightarrow Q$ if and only if, in every L-model $\langle W, \mathcal{R}, \Vdash\rangle$, such that for every $v \in W$ it is $v \Vdash G$, if $w \in W$ and $w \Vdash U$ then $w \Vdash Q$.

For short we write $w \Vdash S$ instead of $\forall Z \in S : w \Vdash Z$. The unsigned version may be reduced to the signed one, just by tagging all formulae in G, U, or Q itself, with the sign T.

2.2 Tableaux and Prefixes

SST use *prefixed formulae* i.e. pairs $\langle \sigma : Z\rangle$ where σ is a non empty sequence of integers called *prefix* and Z is a signed formula. Intuitively σ "names" the possible world where Z holds.

In the sequel, σ or $\langle n_1 n_2 \ldots n_p\rangle$, with $p \geq 1$, will be prefixes, ε the empty sequence, $\sigma_0 \cdot \sigma_1$ the concatenation of the sequence σ_0 with the sequence σ_1 and σn a short cut for $\sigma \cdot \langle n\rangle$. The relation $\sqsubset$ (intuitively $\sigma_0 \sqsubset \sigma$ holds if and only if σ_0 is an initial part of σ) and the binary operator $\sqcap$ ($\sigma_1 \sqcap \sigma_2$ is, in practice, the maximal initial part common to both prefixes) are also used.

The definitions of branch and tableau will be the same (but the rules!) of the prefixed tableau proposed by Fitting [3,4].

Remark 3. Differently from sequent-like tableaux, where many tableaux may be needed to prove a formula satisfiable, here there is *exactly one* tableau for both satisfiability and validity.

The definition of closure and termination is equally simple: a branch is *closed* if there are contradictory prefixed formulae on that branch (i.e. there is a σ such that, for some X, both $\langle \sigma : T.X\rangle$ and $\langle \sigma : F.X\rangle$ are present on the branch or either $\langle \sigma : T.-\rangle$ or $\langle \sigma : F.\top\rangle$ are present); a tableau is closed if every branch is closed. *Termination* occurs when no operation is possible. A branch is *open* if it is terminated and not closed and a tableau is open if at least one branch is such.

Definition 4 (Tableau Proof). Given a set of global assumptions G and local assumptions U in the modal logic L, a *proof* for the signed formula Q, i.e. $G \vdash_{\mathrm{L}} U \Rightarrow Q$, is the tableau starting with $\langle 1 : \overline{Q}\rangle$ and closed by using only SST-rules for the logic L with global assumptions G and local assumptions U.

Intuitively a tableau proof of Q is a failed attempt to prove that $\overline{Q}$ is satisfiable. When the tableau closes, e.g. because $\langle \sigma : T.-\rangle$ is present on the branch, this means that the assumption of $\overline{Q}$ leads to a contradiction in some possible world (the one "tagged" by σ) and thus Q must be valid. To check that Q is satisfiable, one just needs to start with $\langle 1 : Q\rangle$ and end with an open tableau.

2.3 SST-Rules

Table 3 shows the rules which are common to all logics viz. those which add local or global assumptions and those which reduce α, β and π formulae. In the case of π-rule some more terminology is needed: a prefix is *unrestricted* on a branch ϑ, if it is not the initial part of any other prefix already present on the branch.

Table 4 shows the different rules which may be used to reduce a ν-formula. Each rule may be added or deleted from the system, with the same flexibility of Hilbert-style axiomatization. Thus, each collection of SST rules gives us an SST calculus for a different logic, as shown in Table 5.

Table 3. SST-rule Common to All Logic

$$\alpha : \quad \frac{\sigma : \alpha}{\begin{array}{c}\sigma : \alpha_1 \\ \sigma : \alpha_2\end{array}} \qquad \beta : \quad \frac{\sigma : \beta}{\sigma : \beta_1 \quad | \quad \sigma : \beta_2}$$

$$Loc : \quad \frac{\vdots}{1 : Z} \qquad \text{if} \quad Z \in U$$

$$Glob : \quad \frac{\vdots}{\sigma : Z} \qquad \text{if} \quad \sigma \text{ is present on the branch and } Z \in G$$

$$\pi : \quad \frac{\sigma : \pi}{\sigma n : \pi_0} \qquad \text{with} \quad \sigma n \text{ unrestricted on the branch}$$

Table 4. SST-rules for ν Formulae

$$K : \frac{\sigma : \nu}{\sigma n : \nu_0} \qquad D : \frac{\sigma : \nu}{\sigma : \pi} \qquad T : \frac{\sigma : \nu}{\sigma : \nu_0}$$

$$B : \frac{\sigma n : \nu}{\sigma : \nu_0} \qquad 4 : \frac{\sigma : \nu}{\sigma n : \nu} \qquad 4^R : \frac{\sigma n : \nu}{\sigma : \nu}$$

$$T^D : \frac{\sigma n : \nu}{\sigma n : \nu_0} \qquad B^D : \frac{\sigma nm : \nu}{\sigma n : \nu_0} \qquad 4^D : \frac{\sigma n : \nu}{\sigma nm : \nu}$$

Note 5. * σ, σn and σmn must all be already present on the branch, i.e. some π-rules must have introduced them already, and

$$\frac{\sigma : \nu}{\sigma : \pi} \quad \text{equals to} \quad \frac{\sigma : T.\Box X}{\sigma : T.\Diamond X} \quad \text{or} \quad \frac{\sigma : F.\Diamond X}{\sigma : F.\Box X} \ .$$

Table 5. Modal Logics and SST-rules

Logics	ν-SST-rules	Logics	ν-SST-rules	Logics	ν-SST-rules
K	K	T	K + T	OM	K + T^D
KB	K + B	B	K + T + B	OB	K + T^D + B^D
K4	K + 4	S4	T + 4	OK4	K + 4^D
K5	K + 4^D + 4^R	S5	T + 4 + 4^R	OS4	K + T^D + 4^D
K45	K + 4 + 4^R	KD*	K*-rules + D	O^{*+}	O*-rules + D

Remark 6. Each rule does not depend on a particular logic nor on some accessibility relation over prefixes: it just states which formulae may be "passed" to some close neighbours.

2.4 Examples

The following example, for the logic K4, should better clarify the key difference between SST and prefixed tableaux (as devised in [4]).

prefixed K4 K4-SST

$$\frac{\sigma : \nu}{\sigma nmk : \nu_0} \quad \Longleftrightarrow \quad \cfrac{\cfrac{\cfrac{\sigma : \nu}{\sigma n : \nu}}{\sigma nm : \nu}}{\sigma nmk : \nu_0}$$

Note 7. For K4 prefixed tableaux, one has to prove that $\sigma \rhd_{\mathrm{K4}} \sigma nmk$ and that σmnk already occurs on the branch, whereas, for K4-SST, one has to prove only that σn, σnm and σnmk are already present on the branch.

Remark 8. The key feature is that SST-rules access only immediate neighbours of the current prefix, thus boosting efficiency without affecting completeness.

Therefore, when a formula of the type $\langle \sigma : \pi \rangle$ is reduced, and a new prefix is introduced, the ν-formulae, which need to be reduced again, are *only those with the same prefix* σ. "Far away formulae may well be left sleeping". At the same time a Generation Lemma (see Sec. 3) ensures us that, if a "far" prefix occurs on a branch, all intermediate prefixes must occur too (e.g. if σnmk occurs then σ, σn, σnm also do).

A simple proof for the logic K5 is given in Fig. 1.

(1)	$1 : F.\Diamond X \to \Box\Diamond X$	
$(2a)$	$1 : T.\Diamond X$	α − rule at (1)
$(2b)$	$1 : F.\Box\Diamond X$	
(3)	$11 : T.X$	π − rule at $(2a)$
(4)	$12 : F.\Diamond X$	π − rule at $(2b)$
(5)	$1 : F.\Diamond X$	4^R − rule at (4) backward from $\langle 12 \rangle$ to $\langle 1 \rangle$
(6)	$11 : F.X$	K − rule at (5) forward from $\langle 1 \rangle$ to $\langle 11 \rangle$
	Contradiction in $\langle 11 \rangle$	tableau closes

Fig. 1. Proof of the euclidity axiom $\Diamond X \to \Box\Diamond X$ in K5

3 Correctness and Completeness

Only at this stage the relation $\triangleright$ over prefixes need to be introduced. $\triangleright$ will be the "carbon copy" of the accessibility relation $\mathcal{R}$ over possible worlds and will allow us to prove correctness and completeness with a suitable extension of Fitting prefixed tableaux. Obviously, conditions imposed on $\triangleright$ depend on the logic and are specified in Table 6. In the following, we just sketch the main ideas; details can be found in [15].

Table 6. Conditions on Prefixes

Logic	Conditions on the relation over prefixes $\sigma \triangleright \sigma^*$
K	$\sigma^* = \sigma n$
KB	$\sigma^* = \sigma n$ or $\sigma^* n = \sigma$
$K4$	$\sigma = \sigma_0 \cdot \sigma_1$ and $\sigma^* = \sigma_0 \cdot \sigma_2$ and $\sigma_0 \neq \varepsilon$ and $\sigma_1 = \varepsilon$ and $\sigma_2 \neq \varepsilon$
$K5$	$\sigma = \sigma_0 \cdot \sigma_1$ and $\sigma^* = \sigma_0 \cdot \sigma_2$ and $\sigma_0 \neq \varepsilon$ and or $\sigma_1 = \varepsilon$ and $\sigma_2 = n$ or $\sigma_1 \neq \varepsilon$ and $\sigma_2 \neq \varepsilon$ or $\lvert\sigma_0\rvert \geq 2$
$K45$	$\sigma = \sigma_0 \cdot \sigma_1$ and $\sigma^* = \sigma_0 \cdot \sigma_2$ and $\sigma_0 \neq \varepsilon$ and or $\sigma_2 \neq \varepsilon$ or $\lvert\sigma_0\rvert \geq 2$
$D - logics$	same conditions for K − logics
T	$\sigma^* = \sigma n$ or $\sigma^* = \sigma$
B	$\sigma^* = \sigma n$ or $\sigma^* n = \sigma$ or $\sigma^* = \sigma$
$S4$	$\sigma = \sigma_0 \cdot \sigma_1$ and $\sigma^* = \sigma_0 \cdot \sigma_2$ and $\sigma_0 \neq \varepsilon$ and $\sigma_1 = \varepsilon$
$S5$	$\sigma = \sigma_0 \cdot \sigma_1$ and $\sigma^* = \sigma \cdot \sigma_2$ and $\sigma_0 \neq \varepsilon$
$O - logics$	similar to $K - logics$ adding condition$\lvert\sigma_0\rvert \geq 2$

Note 9. * It is always $\sigma_0 \doteq \sigma \sqcap \sigma^*$.

To prove the correctness theorem it is necessary to merge the techniques used for sequent-like and prefixed tableaux:

1. show that properties of $\mathcal{R}$ induce the required behaviour of formulae in different worlds (e.g. when $\mathcal{R}$ is transitive if $w\mathcal{R}w^*$ and $w \Vdash \nu$ then $w^* \Vdash \nu$);
2. define a mapping to assign "names" (prefixes) to "things" (possible worlds) such that there is a suitable matching between $\triangleright$ and $\mathcal{R}$ (see [3,4]);
3. prove a safe step lemma such that, if a tableau is satisfiable then the tableau obtained by the application of any SST rule is still satisfiable;
4. prove the correctness theorem i.e. show that if the tableau closes then the formula is valid.

Theorem 10 (Correctness). *If Q has a proof with the* SST *for the logic* L *with global assumptions G and local assumptions U then Q is valid i.e. $G \models_{\mathrm{L}} U \Rightarrow Q$.*

A simple completeness proof for prefixed tableaux, yet restricted to non euclidean modal logics, has been given by Fitting [3,4]. The same ideas may be applied here, taking into account the fact we are dealing with SST:

1. apply a systematic procedure to the tableau;
2. if the algorithm terminates (no rule may be further applied) and the tableau does not close, then any open branch is saturated for the SST
3. if the algorithm never terminates then, by König Lemma, it is possible to prove that there is an infinite branch that will never close, even "ad infinitum" [4]. This branch can then be chosen as the saturated one;
4. prove that, given the conditions for $\triangleright$ as in Table 6, a branch saturated with the SST rules is also saturated with the prefixed rules of Fitting;
5. prove a strong model existence theorem, building a model with the saturated branch, using the property of L-Pref-SAT and a lemma that prove that the accessibility relation between prefixes has the same properties of the accessibility relation between possible worlds.
6. thus the proof of the completeness theorem is just round the corner.

The idea behind *prefixed downward saturation* for a logic L (L-Pref-SAT for short) is that all possible rules have been applied to a branch ϑ and yet no contradiction has been found. For example, to get consistency of ϑ, $\langle \sigma : Z \rangle$ and $\langle \sigma : \overline{Z} \rangle$ cannot be both in ϑ. To get saturation w. r. t. the ν-rule (ν-L-Pref-SAT), ϑ must be such that if $\langle \sigma : \nu \rangle$ is in ϑ then $\langle \sigma^* : \nu_0 \rangle$ must be in ϑ for every σ^* in ϑ such that $\sigma \triangleright \sigma^*$, according to the logic L (Table 6). Details may be found in [4,15].

However L-Pref-SAT is a powerful propriety, and thus too "expensive": some sets are not the result of any tableau reduction. For instance the set $\{\langle 1 : T.\Box p \rangle, \langle 1111 : T.p \rangle\}$ is K4-Pref-SAT with $U = G = \emptyset$, but there is no K4-tableau corresponding to this set.

The following property better characterizes prefixed tableaux and SST.

Definition 11 (Generated Submodel Property). A set of prefixed formulae ϑ satisfies the generated submodel property iff

1. for any prefix $\langle n_0 n_1 n_2 \dots n_k \rangle$ present in ϑ all "ancestors" $\langle n_0 n_1 \dots n_i \rangle$ must be in ϑ with $i = 0, 1 \dots k$
2. there is exactly one prefix $\langle n_0 \rangle$ in ϑ such that, for every prefix σ in ϑ, it is $n_0 \sqsubseteq \sigma$

Lemma 12 (Generation). *If ϑ is a branch of a tableau then it satisfies the generated submodel property.*

Remark 13. This may be seen as the proof theoretical counterpart of the Generation Theorem by Segerberg (see [12]): checking validity in a model reduces to checking validity in the generated submodels.

An analogous definition of L-SST-SAT can be given: change the ν-condition for Pref-SAT into a similar one for SST-rules. e.g. for K45:

K-SST-SAT if $\langle \sigma : \nu \rangle \in \vartheta$ then $\langle \sigma n : \nu_0 \rangle \in \vartheta$ for every σn present in ϑ.
4-SST-SAT if $\langle \sigma : \nu \rangle \in \vartheta$ then $\langle \sigma n : \nu \rangle \in \vartheta$ for every σn present in ϑ.
4^R**-SST-SAT** if $\langle \sigma n : \nu \rangle \in \vartheta$ then $\langle \sigma : \nu \rangle \in \vartheta$ for every[1] σ in ϑ.

From these and the Generation Lemma the following key lemma results:

Lemma 14 (Prefix Closure). *If a set of prefixed formulae ϑ is* L-SST-SAT *and satisfies the generated submodel property then it is also* L-Pref-SAT.

The main idea behind the proof (see [15]) is to "let a formula travel throughout prefixes until it reaches the right point", e.g. first example of Sect. 2.

One more lemma — indeed a lemma for any logic L — needs to be proved:

[1] Indeed just one.

Lemma 15 (Matching). *Given the conditions on the relation over prefixes $\rhd$ for the logic* L *(in Table 6), $\rhd$ has the same properties of the accessibility relation over possible worlds $\mathcal{R}$ of* L*-frames (in Table 2).*

Now we have all the necessary machinery to state the following

Theorem 16 (Strong Model Existence Theorem). *If ϑ is a set of prefixed formulae that is* L-SST-SAT *with the local assumptions U and global assumptions G then there is an* L*-model where ϑ is satisfiable, with local assumptions U and global assumptions G and a suitable L-interpretation.*

Proof. We reduce SST-SAT to Pref-SAT thanks to Prefix Closure and Generation Lemma and then construct the model using prefixes as possible worlds and $\rhd$ as the accessibility relation i.e.

$$\begin{array}{rcl} W & \doteq & \{\sigma \in \mathbb{N}^+ \ : \ \sigma \text{ is present in } \vartheta\} \\ \sigma\mathcal{R}\sigma^* & \text{iff} & \sigma \rhd \sigma^* \\ v(\sigma, p) & \doteq & \begin{cases} \text{true} & \text{if } \langle \sigma : T.p\rangle \in \vartheta \\ \text{false} & \text{otherwise} \end{cases} \end{array}$$

Then it is possible to prove that

1. $\forall Z \in U \ : \ 1 \Vdash Z$,
2. $\forall Z \in G : \forall \sigma \in W \ : \ \sigma \Vdash Z$,
3. $\forall \langle \sigma : Z\rangle \in \vartheta \ : \ \sigma \Vdash Z$,
4. $\langle W, \mathcal{R}\rangle$is an L-frame.

The first two items are a consequence of the properties of L-Pref-SAT and the third must be proved by induction on the complexity of the formula Z, by L-Pref-SAT too. The fourth is a consequence of the Matching Lemma. □

And strong completeness follows:

Theorem 17 (Strong Completeness). *If Q is* L*-valid with G as global assumptions and U as local assumptions, i.e. $G \models_{\mathrm{L}} U \Rightarrow Q$, then Q has a* L-SST *proof.*

4 Some Properties of SST

Theorem 18 (Church-Rosser). *The* SST *calculus satisfies the strong Church-Rosser Property i.e. it is strongly confluent up to isomorphic renaming of prefixes.*

Proof. SST-rules may be seen as particular term rewriting ones in which ν formulae are not deleted (and thus can be used more than once on the same branch). It's clear that a formula of a given type may be reduced only by the corresponding rule and that one cannot reduce a subformula before reducing its parent. Hence no superposition exists between different rules and it is possible only between different ν-rules. But ν-formulae are not deleted, and thus every critical pair is locally confluent. Moreover, SST rules are left linear (just look at Tables 3 and Table 4). Hence, by Neuman's theorem, the result follows [11]. □

To prove termination properties we devised the founding system:

dim(ϑ) = the sum of sizes of formulae not yet processed or potentially reusable (the latter are only ν-formulae);

scope(ϑ) = the number of prefixes where a ν-formula is active (i.e. not yet processed) or absent (thus potentially introducible by some suitable ν-rule).

Proposition 19. *Given a finite* SST*, the iterative application of α, β, π-rules and the introduction of* local *assumptions terminates for any logic* L*.*

Proposition 20. *Given a finite* SST*, the iterative application of α, β, ν-rules and the introduction of* local and global *assumptions terminates for any logic* L*.*

More results may be proved about the prefixes' graph structure.

Lemma 21. *On a tableaux branch, the formulae with a given prefix are bounded by the subformulae of $G \cup U \cup \{\overline{Q}\}$.*

Lemma 22. *On a tableau branch, the number of prefixes of a given length n is finite and bounded by the number of π-subformulae of $G \cup U \cup \{\overline{Q}\}$ raised to the $n-1$ (π and ν subformulae for D and O^+ logics).*

Proof. By induction on n. If $n = 1$ then the only prefix is $\langle 1 \rangle$. For the inductive step just note that the only way to introduce a new prefix is to reduce a π-formula and those are bounded by Lemma 21. Then a multiplication suffices. □

Theorem 23. *If a tableau branch is infinite, it must contain a "growing chain" of prefixes and, after a finite number of steps, the chain becomes "periodical" i.e. formulae will just be repeated with a longer prefix.*

Proof. By Lemma 21 and 22 the number of formulae with prefixes of a given length is finite and thus a growing chain of prefixes must exist. By Lemma 21 a prefix can have only formulae contained in $G \cup U \cup \{\overline{Q}\}$ and thus the chain must be periodical. □

Thus, even if the underlying countermodel is potentially infinite, an algorithm may check for this periodicity and thus terminate in any case. Clearly the number of steps to realize a branch cannot be closed are bounded by the size of the powerset of subformulae of $G \cup U \cup \{\overline{Q}\}$ but, from a practical point of view, a better condition has been devised for the algorithm of Sect. 5.

5 A General Algoritm

To cope with the reuse of ν-formulae, each formula must be set to one among the following states:

Active: the formula must be processed;
Done: the formula has been processed once and needs not be processed anymore (typically α, β, π-formulae);
Quiet: the formula has been already processed (and thus it isn't anymore active) but may be awaken at any time. Typically it's a ν-formula that becomes active as soon as a π-rule creates a new neighbour prefix.

The Church-Rosser property allows us to choose a suitable ordering of rule application in order to exploit the termination properties of Sec. 4. We just sketch the main procedure and the intuition that lies behind. Therefore, to check if $\overline{Q}$ is satisfiable with the local assumptions U and global assumptions G:

1. set $\langle 1 : \overline{Q} \rangle$ and $\langle 1 : Z \rangle$ for all $Z \in U$ as the starting branch of the tableaux and set them as **Active**;
2. repeat
   (a) choose an open branch;
   (b) iteratively apply all π-rules (thus spannig the tree of possible worlds);

(c) apply, if necessary, Glob-rule to every prefix (thus both new and old worlds will have the same global assumptions);
(d) iteratively apply all ν, α, β-rules (thus moving information from a prefix (a world) to another);
(e) check if any periodicity in the prefix structure has been reached and then suspend the formulae with the periodical prefix (or awake them if the structure is no longer periodical);

3. until a branch is open (with all formulae either **Quiet** or **Done**) and thus $\overline{Q}$ is SATISFIABLE and hence Q NON VALID;
4. or until all branches close, $\overline{Q}$ is NON-SAT and hence Q is VALID.

Note 24. When a ν-formula is reduced, all availables ν-rules are applied and only afterwards the ν-formula is set to **Quiet**. When, later on, a π-rule is applied, and a new prefix σn is created, the *only* ν-formulae that must be "awaken", and set back to **Active**, are those with the prefix σ!

Note 25. To suspend the search on a prefix (thus setting its π-formulae from **Active** to **Quiet**), it is enough to find a shorter prefix such that: the same formulae occur in both, the same rules may be applied to both and the same rule may introduce formulae in both. Later on, the proof search may show that prefixes were not really periodical and then π-formulae may be reset **Active** to continue the computation further.

In any case, only a limited number of prefixes must be considered at any time, and the rest may well be left in secondary memory to be retrieved only when the proof search approaches them.

Obviously, if U and G are huge and largely irrelevant, this algorithm may be inefficient but, thanks to the Church-Rosser property, we can also proceed in an incremental way etc. Anyhow, this and other considerations (e.g. discarding closed branches, pushing β-rule down as much as possible etc.) are at stake during the implementation of the system and will not be considered here.

Remark 26. We stress the fact that the system does not depend on the particular strategy. For example depth first strategies may well be devised or a particular ordering on formula reduction may be fixed etc.

6 Related Works and Conclusion

Sequent-like (or Fitch-style) tableaux are one of the traditional tools for deduction in modal logics and may be found in [2,4,5,8], whereas full model checkers are the tableaux for K45 proposed in [10] or the TABLEAUX system presented in [1]. The latter has been implemented and of treats many logics. A dual approach may be found in [7], where the accessibility relation is made explicit by using restricted quantification, and deduction is performed in first order logic.

Other methods are, more or less, linked by the same idea: use labels (called prefixes, indexes, worlds' paths, etc.) to name the worlds where formulae are supposed to hold.

An intermediate approach is that of prefixed tableaux, proposed by Fitting [3,4], which are closer to model checking than others. This feature allows one to use this system as a precious tool to build completeness theorems for other proof methods (see [15,19] or here).

A comprehensive work about modal resolution is presented by Ohlbach [17] where world paths are used to identify worlds where a formula holds. Different unification procedures are used to cope with different accessibility relations. The system is also lifted to a full first order framework.

The application of matrix proof methods to modal logics is due to Wallen [19]. Prefixes and unification procedures, varying with the logic, are used to ensure connections are drawn between formulae belonging to the same world. The method is definitely effective and spans to first order modal logics but needs a human-oriented interface for the presentation of proofs and proof search (see [19] for further references).

A common limit is that systems not based on pure model checking do not handle euclidean or symmetric logic as well as our proposal. Moreover, passing from a logic to another is not simple, since information is coded in the unification procedure or in the accessibility relation.

We have proposed a new tableau calculus that brings together the advantages of sequent-like tableaux and prefixed tableaux and allows an analytic, deterministic, reasonably efficient, strongly confluent, uniform and "user friendly" approach to many normal propositional modal logics, including symmetric and euclidean ones: K,T,D, KB, K4, K5, K45, KDB, D4, KD5, KD45, KD45, T, B, S4, S5, OM, OB, OK4, OS4, OM^+, OB^+, $OK4^+$, $OS4^+$ and many others. Future work is in the direction of lifting this framework to first order logic.

Moreover, we think that other deduction methods may directly benefit from this work, especially prefixed tableaux and matrix proof methods: they just need to extend respectively $\triangleright$ and the unification procedure with the new conditions for euclidean and other logics contained in Table 6.

Acknowledgements

A warm thank goes to L. Carlucci Aiello and F. Pirri for reading drafts of this paper and for their many useful hints and suggestions. Comments from W. Bibel and anonymous referees helped to improve the presentation of this paper.

References

1. L. Catach. TABLEAUX, a general theorem prover for modal logics. *J. of Automated Reasoning*, 7, 1991.
2. F. Fitch. Tree proofs in modal logic. *J. of Symbolic Logic*, 31, 1966.
3. M. Fitting. Tableau methods of proof for modal logic. *Notre Dame J. of Formal Logic*, 13, 1972.
4. M. Fitting. *Proof Methods for Modal and Intuitionistic Logics.* Reidel, Dordrecht, 1983.
5. M. Fitting. First order modal tableaux. *J. of Automated Reasoning*, 4, 1991.
6. D. M. Gabbay. *Labelled Deductive Systems.* Oxford Univ. Press, Oxford, 1993.
7. I. Gent. Theory Matrices (for Modal Logics) Using Alphabetical Monotonicity. *Studia Logica*, 52, 1993.
8. R. Goré. Cut-free Sequent and Tableau Systems for Propositonal Normal Modal Logics. PhD thesis, Computer Laboratory, Univ. of Cambridge, 1992.
9. J. Y. Halpern and R. Fagin. Modelling knowledge and action in distributed systems. *Distributed computing*, 3(4), 1989.
10. J. Y. Halpern and Y. Moses. A guide to the modal logic of knowledge and belief. In *Proc. of IJCAI-85*, 1985.
11. G. Huet. Confluent reductions: Abstract properties and applications to term rewriting systems. *J. of the ACM*, 1980.
12. G. E. Hughes and M. J. Cresswell. *a Companion to Modal Logic.* Methuen, London, 1984.
13. K. Konolige. On relation between default and autoepistemic logic. *Artificial Intelligence*, 35(3), 1988.
14. W. Marek, G. F. Schwarz, and M. Truszczynski. Modal nonmonotonic logics: ranges, characterization, computation. In *Proc. of KR-91*, San Matteo, CA, 1991.

15. F. Massacci. Ragionamento autoepistemico automatico. Tesi di Laurea, Fac. Ingegneria, Univ. di Roma "La Sapienza", 1993. In Italian.
16. R. C. Moore. Possible world semantics for autoepistemic logic. In M. Ginsberg, editor, *Readings on Nonmonotonic Reasoning*. Morgan Kaufmann, 1990.
17. H. J. Ohlbach. A resolution calculus for modal logic. In *Proc. of CADE-88*, LNCS 310. Springer Verlag, 1988.
18. G. F. Schvarts. Gentzen style systems for K45 and K45D. In A. Meyer and M. Taitslin, editors, *Logic at Botik '89, Symposium on Logical Foundations of Computer Science*. LNCS 363. Springer-Verlag, 1989.
19. L. A. Wallen. Matrix proof methods for modal logics. In *Proc. of IJCAI-87*, 1987.