Hintikka Multiplicities in Matrix Decision Methods for Some Propositional Modal Logics

Serenella Cerrito[1] and Marta Cialdea Mayer[2⋆]

[1] Université de Paris-Sud, LRI, Bât 490,
F-91405 Orsay Cedex, France.
e-mail: serena@lri.fr
[2] Dipartimento di Informatica e Automazione, Università di Roma Tre,
via Vasca Navale 84, 00146 Roma, Italia.
e-mail: cialdea@inf.uniroma3.it

Abstract. This work is a study of the inter-translatability of two closely related proof methods, i.e. tableau (or sequent) and connection based, in the case of the propositional modal logics *K*, *K4*, *T*, *S4*, paying particular attention to the relation between matrix *multiplicity* and multiple use of ν_0-formulae (contractions) in tableaux/sequent proofs.
The motivation of the work is the following. Since the role of a multiplicity in matrix methods is the encoding of the number of copies of a given formula that are needed in order to prove a valid formula, it is important to find upper bounds for multiplicities in order to reduce as much as possible the search space for proofs. Moreover, it is obviously a crucial issue if the matrix method is to be used as a decision method. We exploit previous results establishing upper bounds on the number of contractions in tableau/sequent proofs [4], in order to establish upper bounds for multiplicities in matrix systems.
We obtain two kinds of upper bounds: in function of the size of the formula to be proved and in function of the number of the atomic paths through the unindexed formula-tree. Such bounds may be non-optimal. However, the method used to establish them may be useful for obtaining finer upper bounds.

1 Introduction

Among the different proof methods that have been defined for modal logics, two classes of methods are clearly strongly related: sequent/tableau-based [5, 13] and connection-based methods [17, 18]. The efficiency of proof search strongly depends on features allowing the calculus to reduce the size of the search space. Recently, many improvements of tableau systems for modal logics have been proposed (see for example [8, 9, 12]). However, it has been argued that tableau and sequent-based proof search is intrinsically inadequate to deal with several

⋆ This work has been partially supported by the Italian MURST project "Rappresentazione della Conoscenza e Meccanismi di Ragionamento" and by ASI – Agenzia Spaziale Italiana.

kinds of redundancies in the search space. By contrast, the *matrix proof method*, based on Bibel *connection method* [2], exploits the possibility of searching proofs of formulae by working on their parsing trees. It has been advocated as a very good approach to minimise redundancies, while, admittedly, sacrificing human readability (the problem of translating matrix proofs into "readable proofs" is addressed, for example, in [1, 3, 14, 15]).

From a proof-theoretical point of view, a main interesting aspect of matrix methods is that a matrix can be seen as uniquely encoding a set of tableau proofs which differ only with respect to "irrelevant" permutations of rule applications (pretty much as proof-nets give a unique representation of several "equivalent" sequent proofs in linear logic [7]).

In the matrix method, validity within a logic is characterised by the existence of a set of *connections* (i.e. pairs of complementary literals) within the formula. If every *atomic path* through the formula contains a connection from the set, such a set is said to *span* the formula. However, in modal propositional logics, like in classical first order logic, it may be necessary to use several copies of some subformulae in order to simultaneously "close" all of the atomic paths. This holds not only for logics whose tableau (or sequent) inference rules explicitly allow a reuse of the same formula (i.e. contain implicit or explicit contractions), such as, for example, T and the transitive logics, but also for the simplest normal modal logic K (see Example 1 below).

Given a modal propositional logic $\mathcal{L}$, an upper bound on the needed *multiplicity*, i.e. the number of copies of a given subformula that are needed in order to span a valid formula, is called an *$\mathcal{L}$-Hintikka multiplicity* in [18]. To the best of our knowledge, a Hintikka multiplicity has been so far explicitly established only in the case of *S5* [18]. However, Wallen observes:

> We stress: defining *an* $\mathcal{L}$-Hintikka multiplicity for a given formula for the other logics is straightforward, since the number of subformulae of the formula is finite. This, essentially, is the approach taken in tableau methods, such as Fitting's [5] and Hughes and Cresswell's diagrammatic methods [10], to show that various modal logics are decidable in the first place. For the purposes of automated deduction, since the $\mathcal{L}$-Hintikka multiplicity defines the space that must be searched exhaustively to determine the non-validity of the formula, we are motivated by the desire to determine least such multiplicities. ([18], p.177).

Now, note that if *only* the fact that the number k of subformulae of a formula A is finite is used to determine $\mathcal{L}$-Hintikka multiplicities, without any further analysis, one gets, for instance, an *S4*-Hintikka multiplicity of the order of 2^{2^k} (2^{2^k} being the maximal number of leaves in a *S4* loop-free tableau for A, such as can be determined by using only the subformula property). The aim of this work is to push the analysis further, so as to progress towards the definition of minimal $\mathcal{L}$-Hintikka multiplicities. Such a problem is obviously crucial if the matrix method has to be used as a *decision method.*

Wallen's presentation of modal matrix methods emphasises their relationship with tableau/sequent style systems. In particular, its definition of path mirrors the definition of expansion rules in tableau systems and the use of prefixes closely resembles the role of prefixes in prefixed tableau systems. Moreover, multiplicity corresponds to the possible multiple use of a ν_0-formula in a tableau proof. In effect, the possibility of translating a modal matrix proof into a sequent/tableau proof has been shown in [15], while [16] uses the inverse translation, in order to show how to define a matrix method starting from tableau systems.

In this work, we refine the analysis of the inter-translatability of tableau and matrix systems, paying particular attention to the relation between multiplicity and multiple use of ν_0-formulae. As far as transitive logics are concerned, the multiplicity of a ν_0-formula reflects also the number of times a contraction affects such a formula (or better, its ν-parent) in a tableau proof. Now, such a number can be bounded polynomially in the length of the initial formula when single branches of the tableau are considered [4, 8, 9]. The possibility of exploiting this result in order to establish an upper bound for the multiplicity in modal matrix methods is the original inspiration of this work.

However, the paths through a matrix do not preserve the tree structure of a tableau. Since a matrix is made of pointers to subformulae, multiple copies are needed also when an occurrence of a formula is used in different branches of the corresponding tableau. Intuitively, in a matrix proving A, the value $\mu(a_j)$, where μ is the multiplicity function and a_j the position of a ν_0-subformula B of $\langle A, 0\rangle$, gives a measure of the number n of distinct occurrences of B needed in a tableau $\mathcal{T}$ in order to refute FA. The number n is a function of two quantities:

1. the maximal number of copies of such a B needed in a branch of $\mathcal{T}$;
2. the number of branches of $\mathcal{T}$.

This remark is better illustrated by some examples (see Section 2 for a brief survey of the modal tableau and matrix methods).

Example 1. Consider the K-valid formula A_1: $\Diamond(\neg p \vee \neg q) \vee (\Box p \wedge \Box q)$. Any K-tableau proof of A_1 contains two branches. Each branch contains an application of the π_K-rule, "producing" two copies of the ν_0-formula $F(\neg p \vee \neg q)$: one in the first branch and the other in the second one. Let a_j be the unique position whose label is $\neg p \vee \neg q$ (with polarity 0) in the formula-tree for $\langle A_1, 0\rangle$. It is easy to see that no K-admissible substitution σ and no set of σ-complementary connections can span $\langle A_1, 0\rangle^\mu$ if $\mu(a_j)$ is strictly smaller than 2. In fact, the (variable) position a_j must be simultaneously unified with the (constant) positions associated to p (with polarity 0, sub-formula of $\Box p$) and q (with polarity 0, sub-formula of $\Box q$). Intuitively, the position corresponding to $\neg p \vee \neg q$ must be used twice in the matrix, in order to "match" p on a branch and q on the other.

Example 2. Consider the T-valid formula A_2: $\Diamond(q \to \Box q)$. Any T-tableau proof of A_2 has just one branch but needs at least two copies of the signed formula $F(q \to \Box q)$: the first one, generated by a ν_T-rule, yields $F(q \to \Box q)$, thus $F\Box q$, holding in the actual world. The second one is produced (as a "side effect") when the π_K-rule is applied to $F\Box q$. Let a_j be the (unique) position in the formula tree for $\langle A_2, 0\rangle$ whose label is $q \to \Box q$ (and polarity 0). It is easy to see that no T-admissible substitution σ and no set of σ-complementary connections can span $\langle A_2, 0\rangle^\mu$ if $\mu(a_j)$ is strictly smaller than 2.

Example 3. Consider again the formula A_2 of the previous example, but in the context of *S4*. Again, any *S4*-tableau $\mathcal{T}$ proving A_2 contains exactly one branch. A first application of the ν_T-rule generates a first copy of the ν_0-formula $F(q \to \Box q)$. If $\mathcal{T}$ is a proof of A_2, later on in the branch there is an application of the π_{S4} rule on $F\Box q$; this inference generates the π_0-formula Fq but does not produce any ν_0-formula. Differently from the case of T, a new application of the ν_T rule to $F\Diamond(q \to \Box q)$ is needed in order to close the branch (note the use of a contraction, in sequent terms). This application is responsible of the generation of a second copy of the ν_0-formula $F(q \to \Box q)$. Again, if a_j is the unique position whose label is $q \to \Box q$ (with polarity 0) in the formula-tree for $\langle A_2, 0\rangle$, no multiplicity μ such that $\mu(a_j)$ is strictly smaller than 2 will do.

In this work we provide upper bounds for the multiplicity needed to span a valid formula in the logics *K*, *T*, *K4* and *S4*, to be used when working with the matrix method as a *decision* method. The results are obtained via a translation of tableau proofs into matrices and the exploitation of upper bounds on the number of copies of a single formula needed in a tableau branch in order to obtain a closed tableau. Since the number of branches in tableaux must also be taken into account and such a number is, in general, exponential in the size of the initial formula, the bounds that we obtain in this way are exponential too. It seems plausible that much better cannot be done: in general, a given ν-formula may be used in all the branches of a tableau.

However, since the number of branches in a tableau proof bears some relation with the number a of atomic paths in the unindexed formula tree, it may be interesting in practical cases to limit the multiplicity in terms of a, that may be known in advance and is often far lower than its general upper bound. We have explored also this possibility and found that, while in the case of the logic K the value a itself is a Hintikka multiplicity, in the case of *K4*, *T* and *S4* the Hintikka multiplicity is still an exponential function of a. In many cases however, it is a better bound to consider than the Hintikka multiplicity expressed in terms of the size of the formula. The possibility of finding a polynomial bound in terms of a remains an open issue.

This work is organised as follows. Next section is a brief overview of tableau and matrix proof systems for the modal logics we are considering. Section 3 contains the main results concerning tableau systems (3.1) and illustrates how to shift them in the context of matrix systems in order to obtain $\mathcal{L}$-Hintikka multiplicities (3.2). The proof passes through the provably correct construction of a matrix proof from a tableau proof of a given formula.

2 Tableau and Matrix Proof Methods for Modal Logic

The propositional modal language extends the classical language by means of the modal operators $\Box$ and $\Diamond$. In this work we consider the modal systems *K*, *K4*, *T* and *S4*. The system *K* is axiomatically characterisable by addition of the following schema and inference rule to classical propositional axioms and rules (in the axiomatic characterisation of modal systems, the modal operator $\Diamond$ is usually a defined symbol: $\Diamond A =_{Def} \neg\Box\neg A$):

$$\text{(K)}\quad \Box(A \to B) \to (\Box A \to \Box B) \qquad\qquad \text{(Nec)}\quad \frac{A}{\Box A}$$

The systems *T*, *K4* and *S4* are the extensions of *K* obtained by addition, respectively, of the schema (T), the schema (4), and both schemata (T) and (4):

$$\text{(T)}\quad \Box A \to A \qquad\qquad \text{(4)}\quad \Box A \to \Box\Box A$$

In the following subsections we outline the tableau and connection proof methods for these logics.

2.1 Signed Tableaux for Modal Logics

Tableau systems are definable for modal logics in different formats: either signed or unsigned tableaux, and either prefixed or unprefixed ones. In this work we refer to unprefixed signed tableaux [5]. Although the prefixed version is closer to the matrix method, the use of unprefixed tableaux is preferable in this setting. In fact it restricts the application of rules expanding formulae of possible force (when a rule is applied to one of such formulae, the others are "lost"), therefore allowing for a better analysis of the structure of a tableau proof.

A signed formula is either TX or FX, where X is a formula. A compact formulation of the modal expansion rules can be obtained by defining ν-formulae (formulae of necessary force) and π-formula (formulae of possible force), with their ν_0 and π_0, as follows.

ν	ν_0
$T\Box X$	TX
$F\Diamond X$	FX

π	π_0
$T\Diamond X$	TX
$F\Box X$	FX

The tableau systems for the logics *S4*, *K4*, *T*, *K* are obtained by the addition of specific modal rules to the classical propositional ones, as the following table shows.

K		$(\pi_K)\ \dfrac{\pi, S}{\pi_0, S\#}$	where $S\# = \{\nu_0 \mid \nu \in S\}$
K4		$(\pi_4)\ \dfrac{\pi, S}{\pi_0, S\#}$	where $S\# = \{\nu, \nu_0 \mid \nu \in S\}$
T	$(\nu_T)\ \dfrac{\nu, S}{\nu_0, \nu, S}$	$(\pi_K)\ \dfrac{\pi, S}{\pi_0, S\#}$	where $S\# = \{\nu_0 \mid \nu \in S\}$
S4	$(\nu_T)\ \dfrac{\nu, S}{\nu_0, \nu, S}$	$(\pi_{S4})\ \dfrac{\pi, S}{\pi_0, S\#}$	where $S\# = \{\nu \mid \nu \in S\}$

When tableau rules are compared with sequent calculi rules [5], it is apparent that a contraction is implicit in the rule ν_T, a sequence of weakenings is implicit in the rules π_K and π_{S4}, and both contractions and weakenings are implicit in π_4. Note that the π_4 rule is a derived expansion rule in *S4*, thanks to the presence of the ν_T rule. We prefer, however, not to have implicit applications of the latter rule, when possible.

Signed subformulae of a signed formula are defined as follows: if B occurs positively in A, then TB is a subformula of TA and FB is a subformula of FA; if B occurs negatively in A, then FB is a subformula of TA and TB is a subformula of FA. The calculi enjoy the *subformula property*: if $\mathcal{T}$ is a tableau for the set of modal formulae S, then every formula in a node in $\mathcal{T}$ is a (signed) subformula of a formula in S. As a consequence, every node in a tableau for S is a subset of the set of subformulae in S; hence, any refutable set S of formulae has a closed tableau whose depth does not exceed 2^n, where n is the number of subformulae in S.

We shall use the expressions $\mathcal{L}$-tableau, $\mathcal{L}$-contradictory, etc., where $\mathcal{L}$ is a specific modal system, in the obvious sense.

2.2 A Short Survey of the Matrix Method

We very briefly recall here the basic definitions of the matrix proof method, in the restricted context of the propositional modal logics *K, K4 T, S4*. The reader should consult [17] and [18] for more information.

The matrix method works on signed formulae, using the notation $\langle A, i\rangle$, where A is a formula and i, its *polarity*, is an element of $\{0, 1\}$, instead of FA and TA. Below we often write just X to mean a pair $\langle A, i\rangle$.

A *formula tree* for a (signed) formula X – also called *unindexed formula tree* for X – is the representation of X as a tree. Sometimes, if A is a formula, we shall use the expression *unindexed formula tree for* A to mean "unindexed formula tree for $\langle A, 0\rangle$". Each node of the tree is said to be a *position* and its *label* is a (signed) subformula Y of X. Hence, different occurrences of Y in X correspond to different positions, i.e. distinct nodes, in the formula tree for X. The *tree ordering* $<$ on the positions is the partial ordering induced by the formula tree (a node is seen as "smaller" than any of its sons). Positions are classified as $\alpha, \beta, \nu, \nu_0, \pi, \pi_0$ according to the corresponding classifications of their labels. Let $\mathcal{N}_0$ be the set of ν_0 position of a formula tree for X. A *modal multiplicity* μ is a function from $\mathcal{N}_0$ to the natural integers; as Wallen says, " it serves to encode the number of instances of subformulae of X in the scope of a modal operator of necessary force considered within a putative proof" [18].

Given a multiplicity μ for a (signed) formula X, X^μ is said to be an *indexed formula*. The notion of formula tree for X is then extended in a natural way so as to get the notion of *indexed formula tree* for the indexed formula X^μ; such a tree is a tree of indexed positions. Roughly, an indexed formula tree for X^μ may be seen as obtained from the unindexed formula tree by replacing each subtree T_a rooted in a positions a by n copies of T_a if $\mu(a) = n$; the roots of these n subtrees may be denoted by $a^1, \cdots, a^n$.

Paths through X^μ are sets of indexed positions; the rigorous definition is a recursive one. Here, we give just the "flavour" of how paths are generated by traversing the indexed formula tree for X^μ. The singleton containing the root position is the initial path. New paths may be obtained by replacing positions in the tree by their sons, with a *caveat*: at a β position we split into two paths, one containing β_1, the other β_2. In the case of a ν position, such a position is preserved, but any of its ν_0 sons in the tree for X^μ can be added to get a new path. An *atomic path* is a path that cannot generate any new path.

A *connection* is a subset of an atomic path consisting of two positions of different polarities but labelled by the same (unsigned) atomic formula.

A *modal position* is either a ν_0 position or a π_0 position. The *modal prefix* $pre(a)$ of a position a in an indexed formula tree is the string of modal positions dominating a in the ordering $<$, to which a itself is appended, if it is a modal position. Those elements of $pre(a)$ which are ν_0 positions are seen as variables, the others as constants.

A *modal substitution* σ is essentially a function from the set of ν_0 positions (in the indexed formula tree for X^μ) to strings of modal positions. Given a modal logic $\mathcal{L}$, specific conditions, depending on $\mathcal{L}$, are imposed on σ to ensure its *$\mathcal{L}$-correctness* (our terminology). These conditions reflect the properties of the accessibility relation between words typical of each logic.

A modal substitution σ induces a relation $<<$ on positions such that, if $\sigma(a) = string$ and *string* is not itself a position of ν_0 type, then, for any position a' occurring in *string*, $a' << a$. The transitive closure of the union of the partial ordering $<$ and $<<$ defines a binary relation $\triangleleft$ on positions called *reduction ordering*. Given a modal logic $\mathcal{L}$, a modal substitution σ is said to be *$\mathcal{L}$-admissible* if it is $\mathcal{L}$-correct and the induced reduction ordering is irreflexive.

If σ is a modal substitution for an indexed formula X^μ, a connection $\{a, a'\}$ is said to be *σ-complementary* iff $\sigma(pre(a)) = \sigma(pre(a'))$ (the substitution σ being extended to strings in the natural way). A set of σ-complementary connections *spans* X^μ when every atomic path through X^μ contains a σ-complementary connection.

The following theorem is the base of the matrix method.

Theorem 1 (Wallen [17]). *A propositional modal formula A is $\mathcal{L}$-valid iff there is a modal multiplicity μ, an $\mathcal{L}$-admissible modal substitution and a set of σ-complementary connections that spans the indexed formula $\langle A, 0\rangle^\mu$.*

Example 4. As a short example, let us consider the K-valid formula $A_4 : \Box\neg p \vee \Diamond p$ and the constant multiplicity μ whose value is always 1. An unindexed formula tree for $\langle A_4, 0\rangle$ is:

$$\begin{tikzcd} & a_0 \arrow[dl] \arrow[dr] & \\ a_1 \arrow[d] & & a_2 \arrow[d] \\ a_3 \arrow[d] & & a_4 \\ a_5 & & \end{tikzcd}$$

where $label(a_0) = \langle A_4, 0\rangle$, $label(a_1) = \langle \Box\neg p, 0\rangle$, $label(a_2) = \langle \Diamond p, 0\rangle$, $label(a_3) = \langle \neg p, 0\rangle$, $label(a_4) = \langle p, 0\rangle$ and $label(a_5) = \langle p, 1\rangle$. There exists a single atomic path through this tree, namely $\{a_2, a_4, a_5\}$. The prefix of a_5 is $a_0 a_3$, where both

a_0 and a_3 are constant, because their labels are π_0-formulae (the root of the tree is always treated as a π_0 position). The prefix of a_4 is a_0a_4, where a_4 is a variable, because $label(a_4)$ is a ν_0-formula. Such an atomic path is σ-complementary for the K-admissible substitution σ such that $\sigma(a_3) = a_4$.

3 Hintikka Multiplicities in Matrix Proof Search

3.1 Some Useful Results about Tableau Systems

We noticed in Section 1 that the multiplicity $\mu(a_j)$ – where a_j is a position with $label(a_j) = X$ – needed to span a valid formula A is a function of both the maximal number of inferences "producing" X in a branch and the maximal number of branches in a tableau proof of A. This fact is formally stated and proved in next subsection. Here, we establish upper bounds to the number of inferences "producing" X in a branch and the number of branches in a tableau proof of A. The main results stated in this section are reformulations of results in [4], where they are used in order to give a polynomial translation of *S4* into *T* in the style of [6] and to obtain a bound on the maximal number of ν-inferences needed in a branch of an *S4* tableau proof of A. The latter result yields a bound to the number of inference steps per branch equal to the cube of the length of A and an upper bound to the number of contractions needed in *S4* sequent style proofs.

Some definitions are used in the sequel:

Definition 2. 1. Let $\mathcal{T}$ be a tableau proof of A and X a (signed) subformula of FA. An inference $\mathcal{J}$ in $\mathcal{T}$ is said to *produce* X if X does not occur in the premise of $\mathcal{J}$ and it occurs in the conclusion. In particular, an application of the ν_T-rule $\dfrac{\nu, S}{\nu_0, \nu, S}$ produces ν_0 (if $\nu_0 \notin S$), and applications of the rules π_K or π_4, $\dfrac{\pi, S}{\pi_0, S\#}$ produce, besides the π_0 specific of the inference, also all the ν_0-formulae such that $\nu \in S$.

2. If $\mathcal{T}$ is a tableau, then $br(\mathcal{T})$ is the number of branches in $\mathcal{T}$ and, for any signed formula X, $k(\mathcal{T}, X)$ is the maximal number of rules producing X in any branch of $\mathcal{T}$.
3. If A is a signed formula and S a set of formulae, then:
   (a) If X a subformula of A, $occ(X, A)$ is the number of occurrences of X in A.
   (b) $\beta(A)$ denotes the number of *occurrences* of β-subformulae in A.
   (c) $\nu(A)$ is the number of ν-subformulae of A.
   (d) $\pi(A)$ is the number of π-subformulae of A.
   (e) If X a subformula of A, $deg(X, A)$ is the modal degree of X in A (i.e. the number of modal operators governing X in A) and $deg(A)$ is the maximal modal degree of a subformula in A.
   (f) $deg(S)$ is the maximal modal degree $deg(A)$ for $A \in S$.

If $\mathcal{L}$ is any of the logics K, $K4$, T, $S4$, A is an $\mathcal{L}$-valid formula and X a (signed) subformula of FA, then (abusing the functional notation) we shall use $k(X)$ and $br(A)$ to denote upper bounds for $k(\mathcal{T}, X)$ and $br(\mathcal{T})$, respectively, where $\mathcal{T}$ is a tableau proof of A, i.e. integers such that there exists a tableau proof $\mathcal{T}$ of A such that $k(\mathcal{T}, X) \leq k(X)$ and $br(\mathcal{T}) \leq br(A)$.

The following lemma is a preliminary result that will be used in the sequel.

Lemma 3. *If $\mathcal{L}$ is $K4$, T or $S4$ and A is an $\mathcal{L}$-valid formula, then there exists an $\mathcal{L}$-tableau proof $\mathcal{T}$ of A such that the number of applications of the π-rule (either the π_4 or the π_K or the π_{S4}-rule) in any branch is bounded by:*

1. *$deg(A)$, if $\mathcal{L} = T$;*
2. *$(\nu(A) + 1) \times \pi(A)$, if $\mathcal{L} = K4$ or $\mathcal{L} = S4$;*

Proof. 1. Since the modal degree of the conclusion of a π_K-rule application is smaller than the modal degree of its premise, after the n-th application of the π_K rule in a branch of a T-tableau there may occur only formulae Y such that $deg(Y, A) \geq n$.

2. It can be proved that if A is a $K4$- or $S4$-valid formula, then there is a $K4$- (resp. $S4$)-tableau proof of A where every branch contains at most $\nu(A) + 1$ applications of the π_4 (resp. π_{S4})-rule to the same formula (see [4] or [8], making ideas in [11] explicit). □

The next two results establish upper bounds for $k(X)$ and $br(A)$, respectively.

Lemma 4. *If $\mathcal{L}$ is K, $K4$, T or $S4$ and A is an $\mathcal{L}$-valid formula, then there exists an $\mathcal{L}$-tableau proof $\mathcal{T}$ of A such that, for every ν_0-subformula X of FA there are at most $k(X)$ applications of rules producing X in any branch of $\mathcal{T}$, where:*

1. *if $\mathcal{L} = K$, $k(X) = occ(X, A)$;*
2. *if $\mathcal{L} = T$, $k(X) = deg(X, A) + 1$;*
3. *if $\mathcal{L} = K4$ or $\mathcal{L} = S4$, $k(X) = 1 + ((\nu(A) + 1) \times \pi(A))$;*

This is summarised by the following table:

	K	T	$K4$ *and* $S4$
$k(X)$	$occ(X, A)$	$deg(X, A) + 1$	$1 + ((\nu(A) + 1) \times \pi(A))$

Proof. 1. Every inference in K "consumes" its principal formula and a consumed occurrence never produces copies of itself.

2. The ν_T-rule contains an implicit contraction, therefore a ν-formula can produce copies of itself in the same branch. However, let $\mathcal{B}$ be a branch in a tableau proof of A and $\mathcal{J}_1$, $\mathcal{J}_2$ two applications of rules producing a ν_0 formula X in $\mathcal{B}$. It can be shown that, if between the conclusion of $\mathcal{J}_1$ and the conclusion of $\mathcal{J}_2$ there is no application of the π_K-rule (hence, in particular, $\mathcal{J}_2$ is not an application of the π_K-rule), then the inference $\mathcal{J}_2$ can be eliminated and the resulting tableau is still a proof of A.

   The same reasoning proving Item 1 in Lemma 3 shows that, if no wasteful application of the ν_T rule produces X, there may be at most $deg(X, A) + 1$ inferences producing X in a branch of a tableau proof of A (the "+1" is due to the fact that, possibly, a ν_T-rule produces X before any application of the π_K-rule).

3. Since in $K4$ the only rule producing ν_0-formulae is the π_4-rule, the result follows immediately from Lemma 3.
   Similarly to the case of T, any $S4$-tableau proof of A can be transformed into a tableau proof of A such that for every branch $\mathcal{B}$, if $\mathcal{J}_1$ and $\mathcal{J}_2$ are two applications of the ν_T rule to the same formula in $\mathcal{B}$, then at least one application of the π_{S4} rule is between them. This fact and Lemma 3 clearly imply the desired result (again, the "+1" is due to possibility of a ν_T-rule producing X before any application of the π_K-rule). □

Lemma 5. *If $\mathcal{L}$ is K, K4, T or S4 and A is an $\mathcal{L}$-valid formula, then there exists an $\mathcal{L}$-tableau proof $\mathcal{T}$ of A such that the number of branches of $\mathcal{T}$ is not greater than $br(A)$, where:*

1. if $\mathcal{L} = K$, $br(A) = 2^{\beta(A)}$;
2. if $\mathcal{L} = T$, $br(A) = 2^{\beta(A)\times(deg(A)+1)}$;
3. if $\mathcal{L} = K4$ or $\mathcal{L} = S4$, $br(A) = 2^{\beta(A)\times(1+((\nu(A)+1)\times\pi(A)))}$.

This is summarised by the following table:

	K	T	$K4$ and $S4$
$br(A)$	$2^{\beta(A)}$	$2^{\beta(A)\times(deg(A)+1)}$	$2^{\beta(A)\times(1+((\nu(A)+1)\times\pi(A)))}$

Proof. We preliminarily observe that if a tableau $\mathcal{T}$ contains no more than b applications of the β-rule in any branch, then $br(T) \leq 2^b$.

1. Obvious, since any K-tableau has no more than one inference per formula occurrence in each branch.
2. We must set a limit to the number of times a β-formula can be produced, hence used for a β-inference, in a T-tableau. By Lemma 3, any β-subformula of A needs not be used more than $deg(A) + 1$ times in each branch (even if a β-formula occurs more than once in A, it needs not be expanded more than once in a branch segment between two consecutive applications of the π_K-rule).
3. Using Lemma 3 again and the fact that no formula need be used more then once in each branch segment between two applications of the π_4 (resp. π_{S4})-rule. □

3.2 Hintikka Multiplicities for K, K4, T and S4

All the results of this section descend from a basic result, expressing a relation between the multiplicity, in a matrix proving A, for a position a whose label is X and the number of copies of X produced in a tableau proof of A.

Following [18], we say that a multiplicity μ is an *$\mathcal{L}$-Hintikka multiplicity* for a propositional formula A whenever the absence of an $\mathcal{L}$-admissible modal substitution σ and of a set of σ-complementary connections spanning $\langle A, 0\rangle^\mu$ implies that A is not $\mathcal{L}$-valid. In other words, an $\mathcal{L}$-Hintikka multiplicity for a formula provides a bound on the increase of multiplicity needed to decide whether A is $\mathcal{L}$-valid or not. The result below provides a general method to compute $\mathcal{L}$-Hintikka multiplicities. Its proof constitutes also a syntactical completeness proof of the connection method.

Lemma 6. *Let FA be the root of a closed $\mathcal{L}$-tableau $\mathcal{T}$ whose number of branches is $br(\mathcal{T})$. For any occurrence of a ν_0-subformula X in FA, corresponding to position x in the unindexed formula tree for $\langle A, 0\rangle$, let $prod(x)$ be the maximal number of copies of this occurrence produced in any branch of $\mathcal{T}$. Let μ be the multiplicity defined by: for any ν_0 position x in the unindexed formula tree for $\langle A, 0\rangle$,*

$$\mu(x) = prod(x) \times br(\mathcal{T}).$$

The multiplicity μ is an $\mathcal{L}$-Hintikka multiplicity for A.

Proof. A proof of this result can be given either by a double step mapping (of signed tableau proofs into prefixed tableau proofs, preserving $k(\mathcal{T}, X)$ and $br(\mathcal{T})$, and then mapping the branches of a prefixed tableau proof to the atomic paths of a formula tree with given multiplicity), or else directly mapping signed tableau proofs to matrices. We give here a sketch of the latter proof, that does not require knowledge of prefixed tableau systems.

Let μ be the multiplicity defined above. Let $Tree_{\mathcal{T}}$ denote the indexed formula tree for A^{μ} (we use the indexed notation for such a tree in order to emphasise that it actually depends on $\mathcal{T}$, in that μ itself depends on $\mathcal{T}$).

Let $\mathcal{B}_1, ..., \mathcal{B}_n$ be the set of all the branches in $\mathcal{T}$. The multiplicity μ has been chosen "big enough" to allow the construction of a mapping φ from formula occurrences in any of the $\mathcal{B}_i$ to positions in $Tree_{\mathcal{T}}$ such that:

1. for every occurrence X_i of a signed formula X in $\mathcal{T}$, the label of $\varphi(X_i)$ is X;
2. given any ν_0-subformula X of FA:
   (a) if X_1 and X_2 are two occurrences of X produced by distinct inferences in $\mathcal{B}_i$, then $\varphi(X_1) \neq \varphi(X_2)$.
   (b) if two occurrences X_i and X_j of X are produced by two inferences occurring respectively in two branches $\mathcal{B}_i$ and $\mathcal{B}_j$ of $\mathcal{T}$, then $\varphi(X_i) \neq \varphi(X_j)$.
   (c) if X is produced by an inference $\mathcal{I}$ in the common part of $\mathcal{B}_i$ and $\mathcal{B}_j$, then two different occurrences of X are however considered: $X_i \in \mathcal{B}_i$ and $X_j \in \mathcal{B}_j$ and $\varphi(X_i) \neq \varphi(X_j)$. (This choice is perhaps superfluous but it does not affect the computation of the $\mathcal{L}$-Hintikka multiplicity and makes the proof easier).

The proof consists of the following steps.

1. Given any branch $\mathcal{B}_i$ of $\mathcal{T}$ whose leaf closes via a pair Tp, Fp of atoms, we show that there is a modal substitution $\sigma_{\mathcal{B}_i}$ which unifies the prefixes of the positions $\varphi(Tp)$ and $\varphi(Fp)$.
2. We define a substitution $\sigma_{\mathcal{T}}$ as the union of the $\sigma_{\mathcal{B}_i}$. (Note that such a substitution is well defined because of our choice 2c above).
3. We show that σ_T is $\mathcal{L}$-admissible.

4. We show that any atomic path through $Tree_{\mathcal{T}}$ contains a sub-path $\{a, a'\}$ such that the respective labels of a and a' are two signed atomic subformulae of FA which close some branch in $\mathcal{T}$ (i.e. the subformulae have the form Fp, Tp).
5. We conclude that μ is an $\mathcal{L}$-Hintikka multiplicity for A.

To prove step 1, we reason by induction on the length of the branch $\mathcal{B} = S_0, S_1 ..., S_n$ ending with a leaf closing via, say, Fp, Tp. Of course, the induction shows that the result holds for *sets* of formulae (the S_i's), so that actually formula trees for sets of formulae have to be considered (we do not give the tedious details here). Let p_i and p_j denote the occurrences of Fp, Tp in S_n. We show the existence of a substitution $\sigma_{\mathcal{B}}$ which unifies the prefixes of $\varphi(p_i)$ and $\varphi(p_j)$ (which are determined by the formula-tree $Tree_{\mathcal{T}}$, where $\mathcal{T}$ is a tableau for S_0).

Obviously, each specific logic $\mathcal{L}$ requires a different construction of $\sigma_{\mathcal{B}}$. *We outline here the construction for T.* The cases where the top inference in $\mathcal{T}$ is not a π_K-inference are quite easy and we do not spell them out here. When $\mathcal{B}$ begins with an application of the π_K-rule, such an inference has the form:

$$\frac{\pi, \nu^1, \cdots, \nu^m, E}{\pi_0, \nu^1{}_0, \cdots, \nu^m{}_0}$$

where $\nu^1, \cdots, \nu^m$ are ν-formulae and no formula in E is a ν-formula. Say that the remaining segment of $\mathcal{B}$ is $\mathcal{B}' = S_1 ..., S_n$. Let $\mathcal{T}'$ be the subtree of $\mathcal{T}$ with root node $S_1 = \{\pi_0, \nu^1{}_0, \cdots, \nu^m{}_0\}$, a the position of π_0 in $Tree_{\mathcal{T}}$ (as determined by the mapping φ) and, for i s.t. $1 \leq i \leq m$, let x_i be the (variable) position of ν_0^i (as determined by φ).

Let v_1 and v_2 be the prefixes of $\varphi(p_1)$ and $\varphi(p_2)$ in $Tree_{\mathcal{T}}$ and u_1 and u_2 the prefixes of $\varphi(p_1)$ and $\varphi(p_2)$ in $Tree_{\mathcal{T}'}$.

Since Fp and Tp both occur in S_n, thus also in the conclusion of the π_K-rule, they are subformulae of either π or some of the ν^i in the premise. Therefore, the prefixes v_1 and v_2 have the form, respectively, $a_i u_1$ and $a_j u_2$, where a_i (resp. a_j) is the position associated (via φ) either to the occurrence of π in the displayed rule or to that of the ν^i's where the atom occurs.

Thus, three cases have to be considered:

1. both Tp and Fp are subformulae of π; therefore $a_i = a_j$;
2. both Tp and Fp are subformulae of some of the ν^i; therefore a_i and a_j are both variable positions, let us say $a_i = x_i$ and $a_j = x_j$
3. Tp is a subformula of some of the ν^i's and Fp is a subformula of π (or *vice versa*); therefore a_i is a variable position, let us say $a_i = x_i$.

In the the first case, it suffices to set $\sigma_{\mathcal{B}}$ equal to $\sigma_{\mathcal{B}'}$. In the second and third case, it must be noted beforehand that $x_i \notin u_1$ and $x_i \notin u_2$; similarly for x_j. In the second case, it suffices to add the substitution x_i/x_j (or *vice versa*) to $\sigma_{\mathcal{B}'}$. In the third one, it suffices to add the substitution x_i/a to $\sigma_{\mathcal{B}'}$.

The heart of step 3 is the proof that the reduction ordering $\triangleleft$ induced by $\sigma_{\mathcal{T}}$ is irreflexive. We outline here the proof for the logic T. Suppose that we have a $\triangleleft$-cycle c: $a_1, ...a_i, ..., a_n$ (where $a_1 = a_n$). A $\Rightarrow$*-motif* in the cycle c is a *maximal* segment of c where any two consecutive positions, say a_j and a_{j+1} are such that $a_j << a_{j+1}$. By exploiting the definition of $\sigma_{\mathcal{T}}$, we show that:

1. if a putative cycle contains a number m of $\Rightarrow$-motifs, where $m \geq 2$, than it has a sub-cycle containing a number $m-1$ of $\Rightarrow$-motifs.
2. there exists no cycle containing either one $\Rightarrow$-motif or no $\Rightarrow$-motif at all.

Finally, for step 4, it suffices to observe that the atomic paths through $Tree_{\mathcal{T}}$ can be generated in the same order in which the inference rules are applied in $\mathcal{T}$. □

By the above result and the already established lemmas we get the following theorem.

Theorem 7. *Let A be an $\mathcal{L}$-valid formula, X a subformula of A and $k(X)$, $br(A)$ as in Lemmas 4 and 5, respectively, for the logic $\mathcal{L}$. Let μ be the multiplicity such that, for any ν_0 position a, labelled by X, in the formula tree for A*

$$\mu(a) = k(X) \times br(A).$$

*The multiplicity μ is an $\mathcal{L}$-*Hintikka multiplicity *for A.*

Therefore, if $label(a) = X$:

	K	T
$\mu(a)$	$2^{\beta(A)}$	$(deg(X,A)+1) \times 2^{\beta(A)\times(deg(A)+1)}$

	$K4$ and $S4$
$\mu(a)$	$(1+((\nu(A)+1)\times\pi(A))) \times 2^{\beta(A)\times((1+(\nu(A)+1)\times\pi(A))}$

The above result gives exponential bounds for multiplicities and the reason for this is that the maximal number of branches in a tableau has to be taken into account. However, in many practical cases the actual number of branches in a tableau proof is far from reaching its upper bound. Therefore it can be useful to characterise $br(A)$ as a function of the number of atomic paths in the unindexed formula tree for A. This can be done, thanks to the following lemma.

Lemma 8. *Let A be an $\mathcal{L}$-valid formula, and let $atp(A)$ be the number of atomic paths in the unindexed formula tree for $\langle A, 0\rangle$. Then $br(A)$ is not greater than $br^*(A)$, where:*

1. *$br^*(A) = atp(A)$, if $\mathcal{L} = K$;*
2. *$br^*(A) = atp(A)^{1+deg(A)}$, if $\mathcal{L} = T$;*
3. *$br^*(A) = atp(A)^{1+((\nu(A)+1)\times\pi(A))}$, if $\mathcal{L} = K4$ or $\mathcal{L} = S4$.*

Proof. The case of K is obvious. For the other logics, we note that a maybe wasteful but complete strategy for finding a tableau proof of A applies propositional expansion rules and ν-rules exactly once for each formula, then a π-rule, then again propositional expansion rules and ν-rules, etc. Each of such cycles produces a subtree with no more than $atp(A)$ branches. Therefore we can see the proof as a tree with branching factor equal to $atp(A)$ and whose depth is equal to $1+m$ where m is the maximal number of applications of the π-rule in a branch. The result then follows from Lemma 3. □

The above lemma can be combined with Lemma 6, obtaining the following:

Theorem 9. *Let A be an $\mathcal{L}$-valid formula, X a subformula of A and $k(X)$ as established in Lemma 4. Let μ be the multiplicity such that, for any ν_0 position a, labelled by X, in the formula tree for A*

$$\mu(a) = k(X) \times br^*(A).$$

*The multiplicity μ is an $\mathcal{L}$-*Hintikka multiplicity *for A.*

Therefore, if $label(a) = X$:

	K	T
$\mu(a)$	$atp(A)$	$(deg(X, A) + 1) \times atp(A)^{1+deg(A)}$

	$K4$ and $S4$
$\mu(a)$	$(1 + ((\nu(A) + 1) \times \pi(A))) \times atp(A)^{1+((\nu(A)+1)\times\pi(A))}$

4 Concluding Remarks

In this work we have addressed the problem of finding $\mathcal{L}$-Hintikka multiplicities for the modal logics K, $K4$, T and $S4$. The problem is crucial if the matrix proof methods for such logics are to be used as decision methods. Upper bounds for multiplicity, ensuring completeness, have been established both in terms of the size of the formula A to be proved and the number of atomic paths in the unindexed formula tree for A. These bounds are obtained by exploiting the close relation of matrices to tableaux and previous results setting a bound on the number of "copies" of the same formula needed in a tableau proof.

The problem of establishing good bounds to the proof length in tableau or sequent based proof search has been addressed by several authors. With respect to *time* complexity, all of them consider only the complexity of the construction of a single branch; the number of branches in the tree, responsible of the exponential upper bounds in connection based decision methods, is not taken into account. The main results established in [4] have been already outlined in Section 3.1. Similar ideas have been nicely used in [8] to define a sequent calculus for $S4$ embedding a search strategy which prevents the construction of any looping branch. In [9], a rather different approach is taken to bound the length of $S4$ sequent proofs, based on the extension of the language by means of a new modal connective (and a significant reformulation of the rules of the calculus); this allows the definition of a measure that strictly decreases at any application of an inference rule. It results that not only the length of each branch can be polynomially bounded in the length of the proved formula, but also the space complexity for the associated decision procedure is improved w.r.t. [11]. It would be of interest to analyse how the new calculus relates to the matrix method and find out whether better bounds can be obtained.

In fact, the main issue that still remains to be addressed is which lower bounds can be established for $\mathcal{L}$-Hintikka multiplicities and, consequently, whether the upper bounds can be significantly improved or not.

Acknowledgements. The authors wish to thank the anonymous referees of previous works for their suggestion to work in this direction and Alain Heuerding for useful discussions on this topic.

References

1. P. Andrews. Transforming matings into natural deduction proofs. In *Proc. of CADE-5*, pages 281–292, 1980.
2. W. Bibel. *Automated Theorem Proving*. Vieweg Verlag, 1987.
3. W. Bibel, D. Korn, C. Kreitz, and S. Schmitt. Problem-oriented applications of automated theorem proving. In *Proc. of DISCO-96*, pages 1–21, 1996.
4. S. Cerrito and M. Cialdea Mayer. A polynomial translation of S4 into T and contraction free tableaux for S4. *Logic Journal of the IGPL*, 5(2):287–300, 1997.
5. M. Fitting. *Proof Methods for Modal and Intuitionistic Logics*. Reidel Publishing Company, 1983.
6. M. Fitting. First-order modal tableaux. *Journal of Automated Reasoning*, 4:191–213, 1988.
7. J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50, 1987.
8. A. Heuerding, M. Seyfried, and H. Zimmermann. Efficient loop-check for backward proof search in some non-classical propositional logics. In *Proc. of the 5th Int. Workshop on Theorem Proving with Analytic Tableaux and Related Methods (Tableaux '96)*, 1996.
9. J. Hudelmaier. Improved decision procedures for the modal logics K, T and S4. In *Proc. of the 9th International Workshop, CSL'95, Annual Conference of the EACSL*, pages 320–333. Springer Verlag, LNCS 1092, 1995.
10. G. E. Hughes and M. J. Cresswell. *An Introduction to Modal Logic*. Methuen & Co., 1968.
11. R. E. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM Journal of Computation*, 6:467–480, 1977.
12. F. Massacci. Strongly analytic tableaux for normal modal logics. In *Proceedings of CADE-12*, pages 723–737, 1994.
13. M. Ohnishi and K. Matsumoto. Gentzen method in modal calculi. *Osaka Mathematical Journal*, 9:113–130, 1957.
14. S. Schmitt and C. Kreitz. On transforming intuitionistic matrix proofs into standard sequent proof. In *Proc. of the 4th Int. Workshop on Theorem Proving with Analytic Tableaux and Related Methods (Tableaux '95)*, pages 106–153, 1995.
15. S. Schmitt and C. Kreitz. Converting non-classical matrix proofs into sequent-style systems. In *Proc. of CADE-13*, 1996.
16. L. A. Wallen. Generating connection calculi from tableau- and sequent-based proof systems. In A. G. Cohn and J. R. Thomas, editors, *Artificial Intelligence and its Applications*, pages 35–50. John Wiley & Sons, 1986.
17. L. A. Wallen. Matrix proof methods for modal logic. In *Proc. of the 10th International Joint Conference on Artificial Intelligence (IJCAI-10)*, pages 917–923, 1987.
18. L. A. Wallen. *Automated Deduction in Nonclassical Logics: Efficient Matrix Proof Methods for Modal and Intuitionistic Logics*. MIT Press, 1990.