MATRIX PROOF METHODS FOR MODAL LOGICS

Lincoln A. Wallen

University of Edinburgh, Scotland

Abstract

We present matrix proof systems for both constant- and varying-domain versions of the first-order modal logics K, K4, D, D4, T, 84 and 86 based on modal versions of Herbrand's Theorem specifically formulated to support efficient automated proof search. The systems treat the mil modal language (no normal-forming) and admit straightforward structure sharing implementations. A key fsature of our approach is the use of a specialised unification algorithm to reflect the conditions on the accessibility relation for a given logic. The matrix system for one logic differs from the matrix eystem for another only in the nature of this unification algorithm. In addition, proof search may be interpreted as constructing generalised proof trees in an appropriate tableau- or sequent-based proof system. This facilitates the use of the matrix systems within interactive environments.

1 Introduction.

Modal logics are widely used in various branches of artificial intelligence and computer science as logics of knowledge and belief *(eg.,* [Moo80,HM85,Kon84]), logics of programs *(eg.,* [Pne77]), and for specifying distributed and concurrent systems *(eg.,* (HM84,Sti85b). As a consequence, the need arises for proof systems for these logics which facilitate efficient automated proof search.

Traditional proof systems for modal logics, such as tableau- or sequent-based systems are readily available *(eg.,* [Kan57,Nis83,Fit83]). While these systems are to some extent human-oriented, the proof rules form an inadequate basis for automated proof search since they generate search spaces that contain considerable redundancies. The redundancies arise mainly from the characteristic emphasis on connectives and the proof rules for modal operators and quantifiers.

The matrix methods for first-order classical logic, pioneered by Prawits [Pra60], and further developed by Andrews [And81j and Bibel [Bib8l], have been demonstrated to be less redundant than the most efficient of the resolution based methods for that logic [Bib82b|. The methods combine an emphasis on *connections* (drawn from the resolution methods) with an intensions! notion of a *path.*

In this paper we present matrix proof systems for the modal logics K, K4, D, D4, T, 84 and 85, based on modal versions of Bibel's "computationally improved* Herbrand Theorem for first-order classical logic [Bib82cj. We consider both constant- and varying-domain versions of the first-order modal logics.

The major features of our approach may be summarised as follows. Validity within a logic is characterised by the existence of a set of connections (pairs of atomic formula occurrences: one positive, one negative) within the formula, with the property that every so-called atomic path through the formula contains (as a subpath) a connection from the set (§ 2.4). Such a set of connections is said to span the formula. For classical propositions! logic this condition suffices [And81,Bib8l|. For first-order logic a substitution (of parameters or terms for variables) must be found under which the (then propositional) connections in the spanning set are simultaneously complementary. Conditions are placed on the substitution that ensure amongst other things that a proof within a particular tableau- or sequent-based proof system is constructable from the connections and the substitution [Bib82c, Wal86). This basically amounts to ensuring that the restrictions found on the traditional quantifier rules can be met.

This work was supported in pert by 8ERC grant GR/D/44874

For the propositions! modal logics we keep the basic matrix framework but define a notion of complementarity for atomic formulae that ensures the existence of a proof in one of Fitting's *prefixed* tableau systems [Fit72,Fit83|. This amounts to ensuring that, semantically: the two atomic formulae of a connection can be interpreted as inhabiting the same "possible world," and proof-theoretically: that they can be given the same prefix (| 2.5.1). The key observation is that this can be established by noting the position of the atoms relative to the modal operators in the original formula and utilising a specialised unification algorithm operating over representations of these positions. Clearly, this notion of complementarity is logic-dependent, a dependence which is reflected in the choice of unification algorithm. Lifting these results to first-order constant-domain modal logics is simply a matter of combining this modal notion of complementarity with the first-order notion (§ 2.5.2).

For the varying-domain versions we index individual variables with the prefix of their quantifier. The substitution of one variable for another is permitted provided their prefixes can be unified ($ 2.5.2).

Checking a formula for validity within a modal logic is therefore reduced to a process of path checking and complementarity tests performed by a specialised unification algorithm (5 3). During this process extra copies may need to be considered of universally quantified formulae and/or formulae dominated by a modal operator of "necessary" (Q) force. The duplication in both cases is managed by an extension of Bibel's indexing technique or *multiplicity* [Bib82a] which supports the implementation of the matrix systems using structure-sharing techniques [BM72]. The notions of multiplicity, substitution and spanning sets of connections form the basis of the relationship with Herbrand's Theorem.

A number of authors have attempted to adapt computationally oriented proof systems for first-order logic to the modal logics considered here *(eg.,* [Far83,AM66a,Kon86]). We compare our approach favourably to theirs in Section 4.

2 The modal matrix system*.

2.1 Preliminaries.

We assume familiarity with the usual definition of the modal language and formulae. We let A, B range over formulae and P, Q range over atomic formulae.

A pair $\langle G, R\rangle$, comprising a non-empty set G and a binary relation R on G is called a *frame*. Let D be some non-empty set. A *first-order frame over D* is a triple $\langle G, R, P\rangle$ where $\langle G, R\rangle$ is a frame and P is a mapping from G to non-empty subsets of D. $P(w)$ can be interpreted as the set of individuals that "exist" in the world w.

We can obtain different versions of the first-order logics by restricting the way in which P varies from world to world. For example, we could require the *constant-domain* condition: for $w, w' \in G$, $P(w) = P(w')$. Axiomatically, constant-domain modal logics are obtained by including the so-called Barcan formula $\forall x \Box Ax \Rightarrow \Box \forall x Ax$

$\mathcal{L}$	Condition on R
K	no conditions
K4	transitive
D	idealisation
D4	idealisation, transitive
T	reflexive
S4	reflexive, transitive
S5	equivalence

Table 1: Conditions on accessibility relations.

α	α_1	α_2
$\langle A \wedge B, 1\rangle$	$\langle A, 1\rangle$	$\langle B, 1\rangle$
$\langle A \vee B, 0\rangle$	$\langle A, 0\rangle$	$\langle B, 0\rangle$
$\langle A \Rightarrow B, 0\rangle$	$\langle A, 1\rangle$	$\langle B, 0\rangle$
$\langle \neg A, 1\rangle$	$\langle A, 0\rangle$	$\langle A, 0\rangle$
$\langle \neg A, 0\rangle$	$\langle A, 1\rangle$	$\langle A, 1\rangle$

ν	ν_0
$\langle \Box A, 1\rangle$	$\langle A, 1\rangle$
$\langle \Diamond A, 0\rangle$	$\langle A, 0\rangle$

γ	γ_0
$\langle \forall x A, 1\rangle$	$\langle A, 1\rangle$
$\langle \exists x A, 0\rangle$	$\langle A, 0\rangle$

β	β_1	β_2
$\langle A \wedge B, 0\rangle$	$\langle A, 0\rangle$	$\langle B, 0\rangle$
$\langle A \vee B, 1\rangle$	$\langle A, 1\rangle$	$\langle B, 1\rangle$
$\langle A \Rightarrow B, 1\rangle$	$\langle A, 0\rangle$	$\langle B, 1\rangle$

π	π_0
$\langle \Box A, 0\rangle$	$\langle A, 0\rangle$
$\langle \Diamond A, 1\rangle$	$\langle A, 1\rangle$

δ	δ_0
$\langle \forall x A, 0\rangle$	$\langle A, 0\rangle$
$\langle \exists x A, 1\rangle$	$\langle A, 1\rangle$

Table 2: Classification of signed formulae.

as an additional axiom. Our purpose here is not to choose between these possibilities but to develop matrix proof systems for each of the variants.

If we restrict R to satisfy the conditions outlined in Table 1, we say that (G, *R, P)* is an £-frame over D, where *L* is the logic associated with the conditions. The "idealisation" condition is that for every element $w \in G$ there is some element $w' \in G$ such that $w\ R\ w'$. Once again our purpose is not to choose between these logics but to develop matrix proof systems for each.

An L-model over *D* is a pair $\langle\langle G, R, P\rangle, \Vdash\rangle$ where $\langle G, R, P\rangle$ is an L -frame over *D* and |f- is a relation between elements of *G* and sentences such that: for all $w \in G$

1. $w \Vdash A \wedge B$ iff $w \Vdash A$ and $w \Vdash B$.
2. $w \Vdash A \vee B$ iff either $w \Vdash A$ or $w \Vdash B$.
3. $w \Vdash A \Rightarrow B$ iff either $w \nVdash A$ or $w \Vdash B$.
4. $w \Vdash \neg A$ iff $w \nVdash A$.
5. $w \Vdash \Box A$ iff for all $v \in G$ with $w\ R\ v$, $v \Vdash A$.
6. $w \Vdash \Diamond A$ iff for some $v \in G$ with $w\ R\ v$, $v \Vdash A$.
7. $w \Vdash \forall x A$ iff for all $d \in P(w)$, $w \Vdash A[d/x]$.
8. $w \Vdash \exists x A$ iff for some $d \in P(w)$, $w \Vdash A[d/x]$.

Satisfaction in a model and validity are defined as usual.

A *stgned* formula is a pair $(A_t n)$ where *A* is a formula and $n \in \{0,1\}$. We let AT, *Y* range over signed formulae. Informally, the signs "1" and "0" should be interpreted as the qualifiers "is true" and "is false" respectively. For ease of exposition we use a uniform notation due to Smullyan and Fitting that classifies signed formulae according to their sign and major connective/operator as shown in Table 2.

2.2 Formula occurrences.

A *formula tree* for a signed formula is a variant of its formation tree containing additional information as to the polarity of its subformula occurrences *(i.e.,* whether an occurrence of a subformula is negative or positive within the formula). It is best explained by example. A formula tree *for* the signed formula

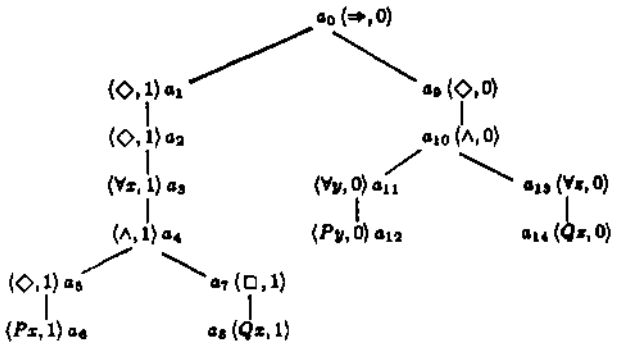

Figure 1: A formula tree.

$$\langle \Diamond\Diamond \forall x (\Diamond Px \wedge \Box Qx) \Rightarrow \Diamond(\forall y Py \wedge \forall z Qz), 0\rangle$$

$(a_0$–$a_{14})$

$\alpha, \beta, \nu, \ldots)$
$(eg., \alpha_1, \alpha_2, \beta_1, \ldots)$

μ_0

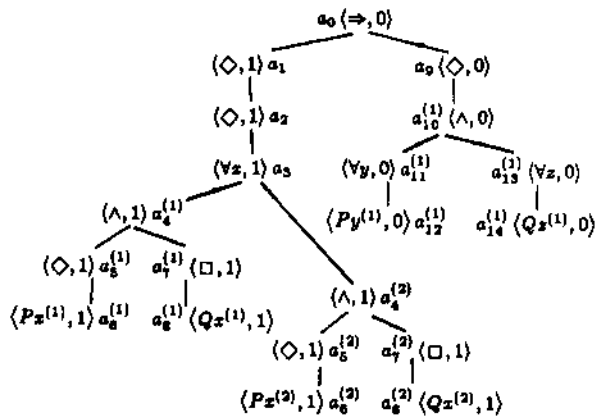

Figure 2: An Indexed formula tree.

A *multiplicity* μ for X is the combination of a modal and first-order multiplicity thus: for a position k of the formula tree

$$\mu(k) = \begin{cases} \mu_M(k), & k \in \mathcal{U}_0; \\ \mu_Q(k), & k \in \Gamma_0; \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

If μ is a multiplicity for X we define the (indexed) formula tree for the *indexed formula* X^μ as a tree of indexed positions of the form k^κ, where k is a position of the basic formula tree for X and κ is a sequence of positive integers defined as follows: if $k_1 < k_2 < \cdots < k_n \leq k$, $1 \leq n$, are those ν_0- and γ_0-type positions that dominate k in the basic formula tree for X, then

$$\kappa \in \{ (j_1\, j_2 \cdots j_n) \mid 1 \leq j_i \leq \mu(k_i),\ 1 < i \leq n \}.$$

The ordering in the indexed tree $<^\mu$ is defined in terms of the ordering on the underlying tree: for indexed positions k^κ and l^τ

$$k^\kappa <^\mu l^\tau \quad \text{iff} \quad k < l \quad \text{and} \quad \tau = \kappa\theta,$$

where θ is some sequence of positive integers. The polarity and label of an indexed position k^κ is taken to be the same as the polarity and label of its underlying position k except that, in the case of atomic formulae, individual variables are indexed with the index of the child of their quantifier position (*i.e.*, a γ_0 or δ_0 position) so as to distinguish the different instances. Consequently, indexed positions inherit the type of their underlying position also.

Figure 2 shows the indexed formula tree for the example formula of Figure 1 with a multiplicity of $\mu_Q(a_4) = 2$ and constant (*i.e.*, 1) otherwise. As a convention we omit indices consisting of the empty sequence.

We let u, v, possibly subscripted, range over indexed positions when we are not interested in the index, and drop the superscript on $<$. We abuse our notation and let $\mathcal{U}_0$, Π_0 *etc*, denote the sets of indexed positions of an indexed formula tree of the appropriate types. Henceforth we shall refer to indexed positions simply as positions.

Remark. Bibel's notion of a multiplicity [Bib82a] corresponds to our notion of a first-order multiplicity. We have altered his definition slightly to support the symmetry between the treatment of modal operators and quantifiers obtained above. Notice that, for an indexed formula, the set Γ_0 and the set of distinct universally quantified variables, and the set Δ_0 and the set of distinct existentially quantified variables in the formula are in 1-1 correspondence. We shall make use of this observation in the sequel. ■

2.4 Paths and connections.

Let X^μ be an indexed formula. A *path* through X^μ is a subset of the positions of its formula tree defined below. We shall use s, t, possibly subscripted, to denote paths, and adopt the notation $s[\alpha^\kappa]$ to denote a path s with an occurrence of a distinguished α-type position with index κ. Similarly for the other types. The set of paths through X^μ, is the smallest set such that:

1. $\left\{ k_0^{(1)} \right\}$ is a path, where $k_0{}^{(1)}$ is the root position of the formula tree for X^μ;
2. if $s[\alpha^\kappa]$ is a path, so is $(s - \{\alpha^\kappa\}) \cup \{\alpha_1{}^\kappa, \alpha_2{}^\kappa\}$;
3. if $s[\beta^\kappa]$ is a path, so are $(s - \{\beta^\kappa\}) \cup \{\beta_1{}^\kappa\}$ and $(s - \{\beta^\kappa\}) \cup \{\beta_2{}^\kappa\}$;
4. if $s[\nu^\kappa]$ is a path, so is $s \cup \{\nu_0{}^{\kappa j}\}$, $1 \leq j \leq \mu_M(\nu_0)$;
5. if $s[\pi^\kappa]$ is a path, so is $s \cup \{\pi_0{}^\kappa\}$.
6. if $s[\gamma^\kappa]$ is a path, so is $s \cup \{\gamma_0{}^{\kappa j}\}$, $1 \leq j \leq \mu_Q(\gamma_0)$;
7. if $s[\delta^\kappa]$ is a path, so is $s \cup \{\delta_0{}^\kappa\}$.

The path $(s - \{\alpha^\kappa\}) \cup \{\alpha_1{}^\kappa, \alpha_2{}^\kappa\}$ is said to have been *obtained by reduction on* α^κ from $s[\alpha^\kappa]$. Similarly in the other cases.

Each path s through X determines a set (*branch* or *sequent*) of positions as follows

$$S(s) = \{ x \mid x \leq y \quad \text{for some} \quad y \in s \}.$$

A path, s, through X^μ is an *atomic path* iff for $k^\kappa \in s$, either

(a) k is labelled by an atomic formula; or

(b) $k \in \mathcal{U}$, and for all j, $1 \leq j \leq \mu_M(\nu_0)$, $\nu_0{}^{\kappa j} \in S(s)$; or

(c) $k \in \Gamma$, and for all j, $1 \leq j \leq \mu_Q(\gamma_0)$, $\gamma_0{}^{\kappa j} \in S(s)$.

Remark. Our definition of path differs from Andrews' [And81] and Bibel's [Bib81] definition so as to demonstrate the relationship between the matrix methods and tableau/sequent methods. Each clause in the definition, when interpreted as operating on the branch associated with the path, corresponds to an analytic tableau rule [Smu68,Fit83]. A path is a representation of the *unused* formulae on a branch. Furthermore, for a given multiplicity, the branch associated with an atomic path is *complete*. These relationships are discussed in more detail in [Wal86]. ■

Consider our example (signed) formula indexed as in Figure 2. If we distinguish its α-type subformulae from its β-type subformulae by placing the components of the former side-by-side and the components of the latter one above the other, we obtain a nested *matrix* thus:

$$\Diamond\Diamond\forall x \left(\begin{pmatrix} \Diamond\left(Px^{(1)}\right) \wedge \Box\left(Qx^{(1)}\right) \end{pmatrix} \quad \begin{pmatrix} \Diamond\left(Px^{(2)}\right) \wedge \Box\left(Qx^{(2)}\right) \end{pmatrix} \right) \Rightarrow \quad \Diamond \begin{pmatrix} \forall y\left(Py^{(1)}\right) \\ \wedge \\ \forall x\left(Qx^{(1)}\right) \end{pmatrix}.$$

Notice that the two instances of the subformula $Px \wedge Qx$ are considered to be the components of an implicit α-type formula. This follows from the γ clause (6) of the definition of paths above. If we omit the connectives and operators we are left with the skeleton matrix:

$$\begin{pmatrix} Px^{(1)} & Qx^{(1)} \end{pmatrix} \quad \begin{pmatrix} Px^{(2)} & Qx^{(2)} \end{pmatrix} \quad \begin{pmatrix} Py^{(1)} \\ Qx^{(1)} \end{pmatrix}$$

which corresponds in part to the so-called "deep formula" in the expansion tree approach of Miller [Mil84].

The atomic elements of an atomic path are simply the horizontal matrix paths through such a matrix. In this case there are two atomic paths through the formula, one with atomic elements $\{Px^{(1)}, Qx^{(1)}, Px^{(2)}, Qx^{(2)}, Py^{(1)}\}$ and one whose atomic elements are $\{Px^{(1)}, Qx^{(1)}, Px^{(2)}, Qx^{(2)}, Qx^{(1)}\}$. More precisely, we should express these sets as positions thus: $\{a_0^{(1)}, a_2^{(1)}, a_6^{(2)}, a_8^{(2)}, a_{12}^{(1)}\}$ and $\{a_0^{(1)}, a_2^{(1)}, a_6^{(2)}, a_8^{(2)}, a_{14}^{(1)}\}$.

A *connection* in an (indexed) formula is a subpath of a path through the formula consisting of two positions labelled by an atomic formula

with the same predicate symbol but of different polarities. A set of connections is said to *span* the formula just when every atomic path through it contains a connection from the set.

For example, the two connections $\{a_8^{(1)}, a_{12}^{(1)}\}$ and $\{a_8^{(1)}, a_{14}^{(1)}\}$ span the indexed formula displayed above. So does the connection pair $\{a_8^{(1)}, a_{12}^{(1)}\}$ and $\{a_8^{(2)}, a_{14}^{(1)}\}$.

2.5 Complementarity.

As remarked above, for a given multiplicity, the atomic paths through an indexed formula serve to represent the branches of a complete analytic tableau with the main formula at its root. In the same spirit, we wish to interpret connections as the two formula occurrences that atomically close the branches on which they occur.

For propositional logic, connections are complementary *by definition*. Since there is no need for multiplicities (no modal operators or quantifiers) this observation leads to a simple characterisation of validity.

Theorem 2.5.1 (Andrews,Bibel) *A propositional formula A is valid iff there exists a set of connections that spans $\langle A, 0\rangle$.*

This theorem is the matrix counterpart to the following theorem for analytic tableaux:

Theorem 2.5.2 (Smullyan) *A propositional formula A is valid iff there exists an atomically closed analytic tableau for $(A, 0)$.*

The matrix theorem is more appropriate as a basis for automated proof search because there is no need to actually construct a tableau. The spanning condition simply ensures that a tableau of the appropriate form can be constructed; we search for a spanning set of connections directly rather than via the connective oriented tableau rules [Wal86].

In the presence of modal operators and quantifiers we must be more careful. We deal with modal operators first.

2.5.1 Propositional modal systems.

Informally we must ensure that the two atomic formulae represented by the positions of a connection can be considered to inhabit the *same* possible world. In terms of tableaux, this involves synchronising the choices of possible worlds made during the reduction of the modal (sub)formulae that contain these atomic formulae as subformulae; or, in terms of positions, the reduction of the ν- and π-type positions that dominate the positions of the connection in the formula tree.

The following definitions are introduced for a given (indexed) formula tree for a given (indexed) formula X^a.

Let T_M denote the union of $\mathcal{U}_0$ and Π_0. We associate a sequence of positions called a *prefix*, denoted $\mathrm{pre}(u)$, with each position u of the formula tree as follows: if $u_1 < u_2 < \cdots < u_n \leq u$, $1 \leq n$, are those T_M-elements that dominate u in the formula tree, then

$$\mathrm{pre}(u) = \begin{cases} (u_1\, u_2 \cdots u_n), & \text{K, K4, D, D4, T, S4;} \\ (u_n), & \text{S5.} \end{cases}$$

The prefix of a position encodes its modal context within the formula tree. We shall use p, q to denote prefixes.

For example, the prefix of $a_8^{(1)}$ is $\left(a_0 a_3 a_9 a_8^{(1)}\right)$ while the prefix of $a_{12}^{(1)}$ is $\left(a_0 a_{10}^{(1)}\right)$.

We can place various conditions on a binary relation $R_0 \subseteq T_M^* \times T_M^*$ as shown in Table 3. Such a relation is an *$\mathcal{L}$-accessibility relation* provided it satisfies the properties associated with $\mathcal{L}$ in Table 4.

Remark. These definitions are adapted from Fitting [Fit72,Fit83]. Each prefix "names" a possible world. Since the positions of the (indexed) formula tree correspond to signed subformulae of X^a, a position taken together with its prefix corresponds to his notion of a *prefixed* signed (sub)formula. The prefix identifies the world in which the subformula is taken to be true or false depending on its sign. Binary relations on prefixes are thus used to represent the properties of the accessibility relation for a given logic. ■

Property	Condition: For $p, q \in T_M^*$
general	$p\, R_0\, pq$, $\lvert q\rvert = 1$
reflexive	$p\, R_0\, p$
transitive	$p\, R_0\, pq$, $\lvert q\rvert \geq 1$

Table 3: Prefix conditions.

$\mathcal{L}$	Properties of R_0
K, D	general
T	general, reflexive
K4, D4	general, transitive
S4	general, reflexive, transitive
S5	every prefix accessible from every other prefix

Table 4: Accessibility on prefixes.

We have indicated that the two positions that constitute a connection must be interpreted as inhabiting the same possible world; *i.e.*, have the same prefix. We ensure this by building a *modal substitution* σ_M under which the prefixes of the positions are identical. The discussion below motivates the ensuing definitions.

Consider a ν-type position u with prefix p. The semantic clause for the subformula rooted at u allows us to conclude that the subformula rooted at the child of u, say v, has the same truth value (sign) in *any* world accessible from the world denoted by p. By definition, the prefix of v is (pv) since v is of ν_0-type. Tables 3 and 4 give us the conditions under which a prefix can be considered to be accessible from p.

Take D4 for example. Any prefix of which p is a proper initial subsequence will be accessible from p. Consequently, if we consider v to be a "variable" and allow it to be instantiated under some mapping $\sigma_M : \mathcal{U}_0 \to T_M^*$ to any non-empty sequence we can guarantee that the image of (pv) under (the homomorphic extension of) σ_M will be accessible from the image of p under (the homomorphic extension of) σ_M. In the case of S4, we allow v to be instantiated with any sequence including the empty sequence to reflect the reflexivity of the S4 accessibility relation. For S5, since our notion of prefix is different, we need only consider unit sequences as possible instantiations for such "variables."

Now consider a π-type position u with prefix p. The semantic clause for the subformula rooted at u allows us to conclude that the subformula rooted at the child of u, say v, has the same truth value (sign) as u in *some* world accessible from the world denoted by p. Again, by definition, the prefix of v is (pv) since v is of a π_0-type. From the tables we can see that (pv) itself is accessible from p by virtue of the fact that accessibility relations on prefixes for all of the logics satisfy the general condition. Consequently we consider π_0-type positions as "constants" under the mappings σ_M introduced above. In the context of a tableau proof, the choice of this possible world must be arbitrary; *i.e.*, the prefix (pv) must be new to the tableau. The "constant" v can only be introduced in a prefix by the reduction of v's parent u, or by the reduction of a ν-type position introducing a "variable" (a ν_0-type position) whose image under σ_M contains v. To preserve soundness therefore, we must ensure that the former can occur before the latter. •

A modal substitution $\sigma_M : \mathcal{U}_0 \to T_M^*$ induces an equivalance relation $\sim_M$ and a relation $\sqsubset_M$ on $T_M \times T_M$ as follows:

1. If $\sigma_M(u) = v$ for some v of ν_0-type, then $u \sim_M v$.
2. If $\sigma_M(u) = p$ and p is not a unit sequence consisting of a ν_0-type position, then for all $v \preceq p$, $v \sqsubset_M u$; where $\preceq$ is the subsequence relation on T_M^*.
3. If $v \sqsubset_M u$ and $u \sim_M u'$, then $v \sqsubset_M u'$.

A modal substitution σ_M is *$\mathcal{L}$-admissible* provided

1. σ_M respects $\mathcal{L}$-accessibility relations R_0, i.e., for all $p, q \in T_M^*$,

$$p\, R_0\, q \quad \text{implies} \quad \sigma_M^\#(p)\, R_0\, \sigma_M^\#(q)$$

where $\sigma_M^*: T_M^* \to T_M^*$ is the homomorphic extension of σ_M to T_M^*.

2. (K-logics only) $u \sim_M u'$ implies $v \sqsubset_M u$ (and hence $v \sqsubset_M u'$) for some position v.

3. $\lhd = (< \cup \sqsubset_M)^+$ is irreflexive, where $\sqsubset_M$ is the relation induced by σ_M described above.

The appropriate notion of complementarity for the propositional modal logics under consideration is as follows: (for an indexed formula X^μ) if σ_M is an $\mathcal{L}$-admissible modal substitution for X^μ a connection $\{u, v\}$ in X^μ is said to be σ_M-*complementary* iff

1. $\sigma_M^*(\text{pre}(u)) = \sigma_M^*(\text{pre}(v))$.

Remarks. The relation $v \sqsubset_M u$ should be interpreted as a prescription that "position v should be reduced before position u," in the sense of tableaux. The relation $\lhd$ is called the *reduction ordering*. Its irreflexivity ensures that we could construct an analytic tableau with X as root using the generic prefixes instantiated by σ_M, so that all of the restrictions on prefixes mentioned above are met. This method of representing the restrictions on traditional modal tableau rules is an adaptation of the method used by Bibel [Bib82a] for the classical quantifier rules.

Suitable mappings can be computed using variants on a string-unification algorithm. In all cases the set of most general unifiers is finite but not necessarily a singleton [Sie84j. For S5 the standard unification algorithm suffices. The admissibility check is an check for acyclicity if <q is interpreted as a directed graph.

The extra condition for the K-logics is a translation into the current setting of Fitting's notion of a *used* prefix. Basically, since these logics are not id e aliz able we must ensure that each prefix (under cr_M) of a vo-type position (formula) has been introduced by the reduction of a 7r-type position (formula) beforehand, pj

We have proved the following theorem:

Theorem 2.5.8 *A propositional modal formula A is C-valid iff there is a modal multiplicity pM,* an L-*admissible modal substitution om and a set of a M -complementary connections that spans the indexed formula* $(A, 0)^{\mu_M}$

The proof involves showing that starting from a tableau with $\{A_j\ 0)$ at its root we can construct an atomic ally closed prefixed tableau by following the reduction ordering induced by the substitution, and prefixing each subformula with the image under the substitution of the prefix of its root position. The multiplicity indicates the number of times a given z/-type formula is reduced to form the tableau. Completeness involves showing that a suitable modal multiplicity *pu* can be constructed to form a modal Hintikka set from the set associated with any non-complementary atomic path *(i.e.,* unclosed branch) through $(A, 0)^{\mu_M}$

Although we have used tableau systems to motivate the definition of the matrix systems, no tableau construction is actually performed in the use of such methods. The theorem above is utilised directly. (See Section 3.)

2.5.2 Fir it-order modal systems.

Extending the propositional matrix systems presented above to first-order modal logics is straightforward. We consider both constant- and varying-domain versions.

For constant-domains, a pair of atomic formulae labelling the positions of a connection can be interpreted as complementary if we can find a *first-order* substitution *OQ* of parameters for individual variables that render the two atoms identical.

For varying-domains, the modalities and quantifiers interact. Universally quantified variables only range over those individuals that 'exist" in the world denoted by the prefix of their quantifiers. Existential quantifiers express the existence of individuals only in the world denoted by their prefixes. Consequently, our first-order substitution *OQ* must respect the modal substitution o*M.*

Instead of introducing an explicit set of parameters we note that there is a 1-1 correspondence between Γ_0 and the set of distinct universally bound variables, and Δ_0 and the set of distinct existentially bound variables within the indexed formula. Consequently first-order substitutions are considered over these positions rather than individual variables. Notice that the position corresponding to the individual variable x quantified at a position u is the child of u in the formula tree.

More formally, let T_Q denote the set $\Gamma_0 \cup \Delta_0$. A first-order substitution is a mapping $\sigma_Q: \Gamma_0 \to T_Q$. For soundness, we must place restrictions on first-order substitutions to ensure that the positions representing parameters introduced for existentially bound variables (Δ_0) are indeed arbitrary. In terms of tableaux, we must ensure that such positions are introduced (by the reduction of their parent) before the introduction of any position representing a universally bound variable which receives the same parameter under the substitution σ_Q. The similarity between these restrictions on quantifier reductions and the restrictions on modal operator reductions is not accidental [Smu70].

A first-order substitution $\sigma_Q: \Gamma_0 \to T_Q$ induces an equivalence relation $\sim_Q$ and a relation $\sqsubset_Q$ on $T_Q \times T_Q$ as follows:

1. If $\sigma_Q(u) = v$ for some v of γ_0-type, then $u \sim_Q v$.

2. If $\sigma_Q(u) = v$ for some v of δ_0-type, then $v \sqsubset_Q u$.

3. If $v \sqsubset_Q u$ and $u \sim_Q u'$, then $v \sqsubset_Q u'$.

A *combined* substitution is a pair consisting of a modal substitution and a first-order substitution. A combined substitution $\langle \sigma_M, \sigma_Q \rangle$ is $\mathcal{L}$-admissible provided

1. σ_M respects $\mathcal{L}$-accessibility relations, as before.

2. (K-logics only) $u \sim_M u'$ implies $v \sqsubset_M u$ (and hence $v \sqsubset_M u'$) for some position v.

3. $\lhd = (< \cup \sqsubset_M \cup \sqsubset_Q)^+$ is irreflexive, where $\sqsubset_M$ and $\sqsubset_Q$ are the relations induced by σ_M and σ_Q respectively as described above and in § 2.5.1.

For constant-domains the appropriate notion of complementarity is as follows (for an indexed formula X^μ}: Let σ be an I!-admissible combined substitution for X". A connection $\{u, v\}$ is σ-*complementary* iff

1. $\sigma_M^*(\text{pre}(u)) = \sigma_M^*(\text{pre}(v))$.

way: (for an indexed formula X^μ) if σ is an $\mathcal{L}$-admissible combined substitution for X^μ, a connection $\{u, v\}$ in X^μ is σ-*complementary* iff

1. $\sigma_M^*(\text{pre}(u)) = \sigma_M^*(\text{pre}(v))$.

2. $\sigma_Q(\text{label}(u)) = \sigma_Q(\text{label}(v))$.

3. If $\sigma_Q(u') = v'$, then $\sigma_M^*(\text{pre}(u')) = \sigma_M^*(\text{pre}(v'))$.

Note the addition of the third clause by which the modal and first-order substitution interact.

Remark. We have blurred the distinction between an individual variable and the position that represents it in order to state the second condition,

Consequently we have:

Theorem 2.5.4 *A (first-order) modal formula A is* $\mathcal{L}$*-valid iff there is a multiplicity u, an L-admissible combined substitution a and a set of a-complementary connections that spans the indexed formula* $(A, 0)^\mu$.

Once again we utilise tableau techniques to prove this theorem.

3 Proof search in the matrix systems.

The matrix systems presented above reduce the task of checking a modal formula for validity to one of path checking and complementarity tests. The path checking is performed by adding connections

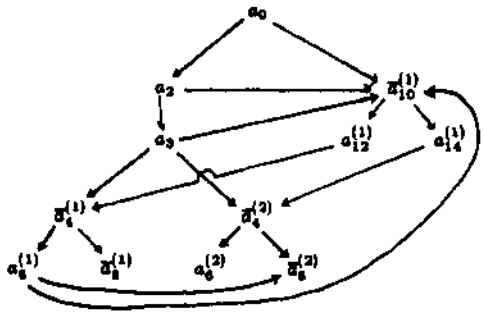

Figure 3: Reduction relation for connections 1 and 2'.

	1	2	2'
Conn.	$a_8^{(1)}, a_{12}^{(1)}$	$a_8^{(1)}, a_{14}^{(1)}$	$a_8^{(2)}, a_{14}^{(1)}$
M-pre.	$a_0a_2a_3a_8^{(1)}, a_0\overline{x}_{10}^{(1)}$	$a_0a_2a_3\overline{x}_8^{(1)}, a_0\overline{x}_{10}^{(1)}$	$a_0a_2a_3\overline{x}_8^{(2)}, a_0\overline{x}_{10}^{(1)}$
σ_M	$\overline{x}_{10}^{(1)} \rightarrow a_2a_3a_8^{(1)}$	$\overline{x}_8^{(1)} \rightarrow a_8^{(1)}$	$\overline{x}_8^{(2)} \rightarrow a_8^{(1)}$
σ_Q	$\overline{x}_4^{(1)} \rightarrow a_{12}^{(1)}$	$\overline{x}_4^{(1)} \rightarrow a_{14}^{(1)}$	$\overline{x}_4^{(2)} \rightarrow a_{14}^{(1)}$
Q-pre.	$a_0a_2a_3, a_0\overline{x}_{10}^{(1)}$	$a_0a_2a_3, a_0\overline{x}_{10}^{(1)}$	$a_0a_2a_3, a_0\overline{x}_{10}^{(1)}$

Table 5: Connections and unification problems.

to a set and eliminating all those atomic paths that contain the new connection. If all atomic paths can be eliminated in this manner, the formula is valid. Complementarity tests are performed as each connection is added. Bibel [Bib82b,Bib82a] shows how some of the standard resolution search strategies can be utilised for this process. His results carry over to our modal systems without change.

Consider the example (indexed) formula of Figure 2. Ignoring the first-order features for the moment, the connection $\left\{a_8^{(1)}, a_{12}^{(1)}\right\}$ gives rise to the problem of unifying the prefixes $\left(a_0a_2a_3a_8^{(1)}\right)$ and $\left(a_0\overline{x}_{10}^{(1)}\right)$, where we have overlined the ν_0-type ("variable") positions. We can immediately see that such a connection cannot be made (propositionally) complementary (condition 1) unless the accessibility relation of the logic is transitive. If this is the case, the (most general) unifier sends $\overline{x}_{10}^{(1)}$ to the sequence $a_2a_3a_8^{(1)}$.

The second connection $\left\{a_8^{(1)}, a_{14}^{(1)}\right\}$ gives rise to the problem of unifying $\left(a_0a_2a_3\overline{x}_8^{(1)}\right)$ and $\left(a_0\overline{x}_{10}^{(1)}\right)$. Since $\overline{x}_{10}^{(1)}$ has the value $a_2a_3a_8^{(1)}$ under the current modal substitution, we can make the two connections (propositionally) complementary if we send $\overline{x}_8^{(1)}$ to $a_8^{(1)}$.

Consider now the first-order features of our example formula. In addition to the modal substitution we must build a first-order substitution which unifies the labels of the connections.

For the first connection we must unify $Px^{(1)}$ with $Py^{(1)}$. This gives rise to the problem of unifying $\overline{x}_4^{(1)}$ with $a_{12}^{(1)}$. The most general unifier simply maps the former position to the latter. So far so good. Consider now the second connection. That gives rise to the problem of unifying $Qx^{(1)}$ with $Qx^{(1)}$, *i.e.*, $\overline{x}_4^{(1)}$ with $a_{14}^{(1)}$. Clearly we cannot build a consistent mapping for $\overline{x}_4^{(1)}$ which unifies both labels.

Due to the multiplicity of a_4 there is an alternative connection $\left\{a_8^{(2)}, a_{14}^{(1)}\right\}$ which together with the first also forms a spanning set (§ 2.4). Propositionally, this connection gives us the problem of unifying $\left(a_0a_2a_3\overline{x}_8^{(2)}\right)$ and $\left(a_0\overline{x}_{10}^{(1)}\right)$ which is easily accomplished by mapping $\overline{x}_8^{(2)}$ to $a_8^{(1)}$ (recall that $a_{10}^{(1)}$ is already mapped to $a_2a_3a_8^{(1)}$). At the first-order level we must unify $Qx^{(2)}$ and $Qx^{(1)}$, *i.e.*, $\overline{x}_4^{(2)}$ with $a_{14}^{(1)}$ which can now be accomplished.

The reduction ordering induced by these substitutions is shown as a graph in Figure 3. Notice that it is cyclic. It is easy to show that no increase in multiplicity can overcome this. Consequently we conclude that the formula is not valid in the first-order constant domain versions of the transitive logics.

We can also check the third condition to determine the status of the formula with respect to the varying-domain logics. Our first-order substitution mapped $\overline{x}_4^{(1)}$ to $a_{12}^{(1)}$ and $\overline{x}_4^{(2)}$ to $a_{14}^{(1)}$. The prefix of $\overline{x}_4^{(1)}$ is $(a_0a_2a_3)$ while the prefix of $a_{12}^{(1)}$ is $\left(a_0\overline{x}_{10}^{(1)}\right)$. Under the modal substitution this latter prefix becomes $\left(a_0a_2a_3a_8^{(1)}\right)$. Since these two prefixes cannot be unified the connections are not complementary in the varying-domain logics. (Notice that we do not even get as far as a cyclicity check in this case.) The prefixes and unifiers are summarised in Table 5.

The path checking process may be interpreted as constructing proof trees in a prefixed tableau/sequent based proof system where the prefixes contain "Skolem" variables and are interpreted as "Skolem* functions. The appropriate systems are similar in spirit to those of Jackson and Reichgelt [JR87]. This has been utilised in implementations to provide a human oriented interface to the search [WW87]. Note that we are concerned with an interface to the search itself rather than the presentation of an already constructed proof for which the techniques of [And80,Mil84] are applicable.

4 Related work

There are two main approaches for extending resolution techniques to modal logics. The first is to restrict the syntactic form of formulae, so that an appropriate modal clausal-form may be defined, and apply clausal resolution techniques *(eg.,* [Far83]). Bibel's comprehensive comparison of clausal resolution-based methods and his matrix method for first-order logic [Bib82b] suffices to demonstrate the advantages of proof search based on the matrix approach for modal logics presented above.

The second approach is to restrict the application of the resolution rule to modal contexts in which it is sound. In semantic terms this means utilising resolution within each possible world. Inference across possible worlds is performed by another mechanism. Abadi and Manna's systems [AM86a,AM66·bj, based on non-clausal resolution [MW80,Mur82|, form perhaps the most comprehensive extension of resolution techniques to modal logics along these lines. The mechanism they employ to manage modalities are Hilbert-style deduction rules which are used to conjoin new formulae. For example, the modal deduction rules for 85 are:

$$\text{M1}: \ \Box A, \Diamond B \mapsto \Diamond(\Box A \wedge B) \qquad \text{M3}: \ \Box A \mapsto A$$
$$\text{M2}: \ \Diamond A, \Diamond B \mapsto \Diamond(\Diamond A \wedge B) \qquad \text{M4}: \ A \mapsto \Diamond A.$$

While hand proofs using these systems can be short, the search spaces they generate are quite redundant due to the connective-based rules for manipulating modalities. Combinations of MS and M4 must be aplied to facilitate the application of MI and M2. Only when complementary subformulae are moved into the same modal context in this manner can the resolution rule be applied. Moreover, since the systems are generative, rules remain applicable to old formulae throughout the proof. This should be compared with our connection based approach and the *calculations* used to establish validity illustrated in the previous section. In the example there, the propositional structure of the formula defined the space to be searched (four possible connections). The modal operators were dealt with using a unification algorithm.

Konolige's systems |Kon86j are based on tableau systems (one tableau for each possible world). Ordinary resolution is utilised within each tableau and a version of Stickel's Theory-resolution [Sti85a] used 'to manipulate modalities by creating new tableaux. Search is complicated by the need to choose suitable sets of formulae to form these new tableaux. The use of theory resolution is not effective, in the sense that an arbitrary amount of search must be performed to determine that the generation of a given resolrent is indeed sound. Konolige proposes the use of multiple refutation procedures to overcome these problems.

References

[AM86a] M. Abadi and Z. Manna. Modal theorem proving. In J.H. Siekmann, editor, *8th International Conference on Automated Deduction,* pages 172-180, July 1986. Lecture Notes in Computer Science, Volume 230, Springer Verlag.

[AM86b] M. Abadi and Z. Manna. A timely resolution. In *Proceedings of Symposium on Logic in Computer Science,* pages 176-186, June 1986.

[And80] P.B. Andrews. Transforming matings into natural deduction proofs. In W. Bibel and R. Kowalski, editors, *5th International Conference on Automated Deduction,* pages 281-292, 1980. Lecture Notes in Computer Science, Volume 87, Springer Verlag.

|And8l] P.B. Andrews. Theorem-proving via general matings. *Journal of the Association for Computing Machinery,* 28(2): 193-214, April 1981.

[Bib8l] W. Bibel. On matrices with connections. *Journal of the Association for Computing Machinery,* 28(4):633-645, October 1981.

[Bib82aj W. BibeL *Automated Theorem Proving.* Friedr. Vieweg *k* Sohn, Braunschweig, 1982.

[Bib82b] W. Bibel. A comparative study of several proof procedures. *Artificial Intelligence,* 18:269-293, 1982.

[Bib82c] W. BibeL Computationally improved versions of Herbrand's Theorem. In J. Stern, editor, *Proceedings of the Herbrand Symposium, Logic Colloquium '81,* pages 11-28, North-Holland Publishing Co., 1982.

[BM72] R.S. Boyer and J.S. Moore. The sharing of structure in theorem-proving programs. In B. Meltser and D. Michie, editors, *Machine Intelligence 7,* pages 101-116, Edinburgh University Press, 1972.

[Far83] L. Farinas-del-Cerro. Temporal reasoning and termination of programs. In S. Amarel, editor, *8th International Joint Conference on Artificial Intelligence,* pages 926-929, 1983.

[Fit72| M.C. Fitting. Tableau methods of proof for modal logics. *Notre Dame Journal of Formal Logic,* XIII:237-247, 1972.

[Fit83] M.C. Fitting. *Proof methods for modal and intuitionistic logics.* Volume 169 of *Synthese library,* D. Reidel, Dordrecht, Holland, 1983.

[HM84] J.Y. Halpern and Y. Moses. Knowledge and common knowledge in a distributed environment. In *3rd ACM Conference on the Principles of Distributed Computing,* pages 50-61, 1984.

[HM85J J.Y. Halpern and Y. Moses. A guide to the modal logics of knowledge and belief:preliminary draft. In *9th International Joint Conference on Artificial Intelligence,* pages 479-490, 1985.

[JR87] P. Jackson and H. Reichgelt. A general proof method for first-order modal logic. Submitted to IJCAI-87, 1987.

[Kan57j S. Kanger. *Provability in logic.* Volume 1 of *Stockholm Studies in Philosophy,* Almqvist and Wiksell, Stockholm, 1957.

|Kon84] K. Konolige. *A Deduction Model of Belief and its Logics.* PhD thesis, Stanford University, 1984.

[Kon86] K. Konolige. Resolution and quantified epistemk logics. In J.H. Siekmann, editor, *8th International Conference on Automated Deduction,* pages[a] 199-208, July 1966. Lecture Notes in Computer Science, Volume 230, Springer Verlag.

[Mil84] D.A. Miller. Expansion tree proofs and their conversion to natural deduction proofs. In R.E. Shostak, editor, *7th International Conference on Automated Deduction,* pages 375-393, May 1984. Lecture Notes in Computer Science, Volume 170, Springer Verlag.

[Moo80] R.C. Moore. *Reasoning about knowledge and action.* Technical Note 191, SRI International, Menlo Park, Ca., 1980.

[Mur82] N.V. Murray. Completely non-clausal theorem proving. *Artificial Intelligence,* 18:67-85, 1982.

[MW80] Z. Manna and R. Waldinger. A deductive approach to program synthesis. *ACM Transactions on Programming Languages and Systems,* 2(1):90-121, 1980.

[Nis83] H. Nishimura. Hauptsats for higher-order modal logic. *Journal of Symbolic Logic,* 48(3):744-751, September 1983.

[Pne77] A. Pneuli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science,* pages 46-57, 1977.

[Pra60] D. Prawits. An improved proof procedure. *Theoria,* 26:102-139, 1960.

|Sie84] J.H. Siekmann. Universal unification. In Shostak R.E., editor, *7th International Conference on Automated Deduction,* pages 1-42, May 1984. Lecture Notes in Computer Science, Volume 170, Springer Verlag.

[Smu68] R.M. Smullyan. *First-Order Logic.* Volume 43 of *Ergebnisse der Mathematik,* Springer-Verlag, Berlin, 1968.

[Smu70] R.M. Smullyan. Abstract quantification theory. In J. Myhill and R.E. Vesley, editors, *Intuitionism and Proof Theory,* pages 79-91, North Holland, Amsterdam, 1970.

[Sti85a| M.E. StickeL Automated deduction by theory resolution. *Journal of Automated Reasoning,* 1:333-355, 1985.

[Sti85b] C. Stirling. *Modal logics for communicating systems.* Technical Report CSR-193-85, Dept. of Computer Science, Edinburgh University, 1985.

[Wal86] 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 *k* Sons Ltd., 1986.

[WW87] L.A. Wallen and G.V. Wilson. A computationally efficient proof system for S5 modal logic. In *Proceedings of AISB-87,* Wiley *k* Sons, 1987.