May 1986

Report No. STAN-CS-86-1100

Modal Theorem Proving

by

Martin Abadi and Zohar Manna

Department of Computer Science

Stanford University
Stanford, CA 94305

MODAL THEOREM PROVING

Martin Abadi & Zohar Manna

Computer Science Department
Stanford University

We describe resolution proof systems for several modal logics. First we present the propositional versions of the systems and prove their completeness. The first-order resolution rule for classical logic is then modified to handle quantifiers directly. This new resolution rule enables us to extend our propositional systems to complete first-order systems. The systems for the different modal logics are closely related.

1. INTRODUCTION

Modal logics ([HC]) have found a variety of uses in Artificial Intelligence (e.g., [MC]), in Logics of Programs (e.g., [P]), and in the analysis of distributed systems (e.g., [HM]). For such applications, natural and efficient automated proof systems are very desirable. A variety of decision procedures have been proposed for propositional modal logics (e.g., [W]). The traditional proof systems for first-order modal logics are simple; this makes them appropriate for metamathematical studies ([Fill). However, they often require much creative help from a user or give rise to long proofs. Thus, they are not suitable for automatic implementation.

Classical clausal resolution proofs ([R]) are usually short and their discovery requires little or no human guidance. Classical nonclausal resolution ([MW1], [Mu]) has the virtue of added clarity, since formulas do not need to be rephrased in unnatural and sometimes long clausal forms.

Fariñas del Cerro ([Fa1], [Fa2], [Fa3]) proposed imitating classical clausal resolution in some modal logics. The proposed methods are rather attractive, but fail to treat the full modal logics under consideration – quantifiers are not allowed in the scope of modal

This research was supported in part by the National Science Foundation under grant DCR-84-13230 and by the Defense Advanced Research Projects Agency under Contract N00039-84-C-0211.

To appear in the Proceedings of the Eighth Intornational Conference on Automated Deduction, Oxford, England, July 1986.

operators. Geissler and Konolige ([Ko], [GK]) attempted to solve this problem with the addition of a new operator, ● , and the introduction of "semantic attachment" procedures.

In this paper we extend nonclausal resolution to eight modal logics with the operators □ ("necessarily") and ◇ ("possibly"). Our approach is quite uniform and generalizes to a wide class of modal logics in different languages. For instance, this class includes logics of knowledge with a knowledge operator K_i for each knower. In fact, all "analytic logics" as well as some "non-analytic" ones (in the terminology of Fitting ([Fill)) are tractable by these techniques. Also, similar methods can be used for more comnlicated loeics. such as Temporal Logic ([AM1], [AM2]).

In the next section we introduce some basic definitions. In section 3 we present the propositional proof systems for K, T, K4, S4, S5, D, D4, and G; their completeness is proved in section 4. These propositional modal systems are lifted to first-order modal systems by adding some quantifier rules (section 5), special auxiliary rules (section 6), and an extended resolution rule (section 7). Skolemization rules (mentioned in section 5) are optional. Section 8 contains a simple example. The completeness of the first-order systems is proved in section 9.

2. PRELIMINARIES

a. Informal syntax and semantics

The propositional modal language includes propositions, modal operators, and connectives. All propositions are ***flexible***, i.e., they may change value from "world" to "world." The modal operators we consider are the usual ones: □ ("necessarily") and ◇ ("possibly"). The primitive connectives are just ¬, A, V, *true,* and *false.* It is practical to regard all other connectives as abbreviations. Formulas are not restricted to any special form such as clausal form.

For the first-order versions, the quantifiers V and 3, variables, and flexible predicate symbols are added. It is convenient and natural to include flexible function symbols and world-independent, *rigid* predicate and function symbols as well. Informally, we may say that variables are also rigid. For example, the formula $\exists x.[q(x) \vee \Box p(x)]$ expresses that the same object has property q or necessarily has property p.

Models and and the satisfaction relation can be described in terms of possible worlds ([HC]). A model is a tuple $\langle D, W, w_0, R, I\rangle$, where

- the domain D is a non-empty set (note that we require that there be just one domain rather than one for each element of W);
- W is a set with a distinguished element w_0; intuitively W is the set of possible worlds and w_0 the real world;
- R is a binary *accessibility relation* on *W;*

- the *interpretation* I gives a meaning over D to each predicate symbol and each function symbol at each world in *W;* the meaning of rigid symbols is required to be the same at all worlds.

An *assignment* α is a function from the set of variables to D. The *satisfaction* relation, $\models$, is then defined inductively over formulas. In particular, the semantics of $\Diamond$ and 3 are given by:

$$(\langle D, W, w_0, R, I\rangle, \alpha) \models \Diamond u \quad \text{if for some } w_1 \in W,\ w_0 R w_1 \text{ and } (\langle D, W, w_1, R, I\rangle, \alpha) \models u,$$

$$(\langle D, W, w_0, R, I\rangle, \alpha) \models \exists x.u \quad \text{if for some } d \in D,\ (\langle D, W, w_0, R, I\rangle, \alpha \cdot \langle x \leftarrow d\rangle) \models u.$$

As usual, the semantics of $\Box$ and V are dual to those of $\Diamond$ and 3, respectively, and *validity* is defined as the dual of satisfiability. Free variables are implicitly universally quantified: u is valid exactly when $\forall x.u$ is valid.

The different logics are characterized by properties of the accessibility relation *R:*

K: R does not need to satisfy any special conditions.

T: *R* is reflexive.

K4: *R* is transitive.

S4: *R* is reflexive and transitive.

S5: *R* is reflexive, symmetric, and transitive.

D: *R* is serial (i.e., there is some accessible world from every world).

D4: *R* is serial and transitive.

G: R^{-1} is transitive and well-founded.

b. Proofs and rules

$\vdash$ w denotes that the formula w can be proved by resolution, that is, that there is a sequence of formulas $S_0, \ldots, S_n$ such that $S_0 = \neg w$, $S_n =$ *false,* and S_{i+1} is obtained from S_i by an application of a rule. We refer to $S_0, \ldots, S_n$ as a *proof* of *w* or *a refutation* of $\neg w$.

Our proof systems include two kinds of rules: simplification rules and deduction rules.

- The *simplification rules* have the form

$$u_1, \ldots, u_m \ \Rightarrow\ v.$$

Suppose the formulas $u_1, \ldots, u_m$ occur in some conjunction in S_i, in any order. Then we delete an occurrence of each of them and add the derived formula v to the conjuction.

Example:

The rule u, $\neg u \Rightarrow$ *false* applied to

$$S_i : (q \vee \Diamond(\neg p \ \mathrm{A} \ q \ \mathrm{A} \ p))$$

yields

$$S_{i+1} : (q \vee \Diamond(q \ \mathrm{A} \ \mathit{false})).$$ ∎

- The *deduction* rules have the form

$$u_1, \ldots, u_m \mapsto v.$$

Suppose the formulas $u_1, \ldots, u_m$ occur in some conjunction in S_i, in any order. Then the derived formula v is added to that conjunction.

Unlike simplification rules, deduction rules do not discard the premises $u_1, \ldots, u_m$. Sometimes, however, we may use the weakening rule (defined in section 3) to discard $u_1, \ldots, u_m$ immediately after applying a deduction rule.

Example:

The rule $\Box u$, $\Diamond$v $\mapsto$ $\Diamond(u$ A v) applied to

$$S_i : q \vee [\Diamond q \wedge r \wedge \Box p]$$

yields

$$S_{i+1} : q \vee [\Diamond q \wedge r \wedge \Box p \wedge \Diamond(p \wedge q)].$$ ∎

An occurrence of a subformula has *positive polarity* in a formula if it is in the scope of an even number of explicit or implicit $\neg$'s. It has *negative polarity* if it is in the scope of an odd number of $\neg$'s. For instance, $\Box p$ occurs with positive polarity and *false* occurs with negative polarity in $\Diamond \neg(\mathit{false} \vee \neg \Box p)$.

We use the following *polarity restriction* to reduce the proof search space:
Rules are applied only to positive occurrences of $u_1, \ldots, u_m$.

c. Soundness

For our proof notion to be meaningful, we require that rules be sound, i.e., that they maintain satisfiability: if S_i is satisfiable then S_{i+1} is satisfiable as well.

We say that *u entails* v (and denote it u $\hookrightarrow$ v) if (u $\supset$ v) is valid. The following observation is often helpful in soundness arguments: a formula gets "truer" as its positive subformulas get "truer" and as its negative subformulas get "falser." More precisely, we can prove:

Lemma (Monotonicity of entailment):

For all u and v, if $u \hookrightarrow$ v and
w' is the result of replacing one positive occurrence of u by v in w, or
w' is the result of replacing one negative occurrence of v by u in w,
then w $\hookrightarrow$ w'.

Proof sketch: The lemma is proved by complete induction on pairs of formulas, with the order $\prec$ defined by: $(w,w') \prec (z,z')$ if w and w' are proper subformulas of z and z', respectively, or of z' and z, respectively. ▮

Suppose that for any S_i and any S_{i+1} obtained from S_i by applying a given rule we have $S_i \hookrightarrow S_{i+1}$. Then soundness is clearly guaranteed for the rule under consideration. Consequently, we can use the lemma to conclude that simplification rules are sound if v $\hookrightarrow$ $(u_1$ A . . . A $u_m)$ for negative occurrences of $u_1, \ldots, u_m$. For positive occurrences, it suffices that $(u_1$ A . . . A $u_m)$ $\hookrightarrow$ v. The entailment $(u_1$ A . . . A $u_m)$ $\hookrightarrow$ v holds for all the simplification rules we will present, except for the skolemization rules. With the polarity restriction, this guarantees the soundness of all the simplification rules except the skolemization rules. We prove the soundness of the skolemization rules with a different method.

Similarly, deduction rules are always sound for negative occurrences of $u_1, \ldots, u_m$ (since the given formulas $u_1, \ldots, u_m$ are kept); for positive occurrences, $(u_1$ A.. . A $u_m) \hookrightarrow v$ suffices. The entailment $(u_1$ A. . . A u $_m) \hookrightarrow$ v holds for all the deduct ion rules we will present. This guarantees the soundness of deduction rules, with no need for polarity arguments.

3. PROPOSITIONAL SYSTEMS

a. **Simplification** rules

- *true-false simplification* rules:

These are the regular *true-false* simplification rules, such as

false V *u* ⇒ *false* and *false, u* ⇒ *false,*

and the rule

$\Diamond$ *false* ⇒ *false.*

- *Negation* rules:

$\neg \Box u \Rightarrow \Diamond \neg u, \neg \Diamond u \Rightarrow$ ❑ ✋ ✞📪

$\neg(u \wedge v) \Rightarrow (\neg u \vee \neg v), \neg(u \vee$ v) $\Rightarrow (\neg u \wedge \neg v), \neg\neg u \Rightarrow u.$

- *Weakening* rule:

  $$u, v \Rightarrow u.$$

  The weakening rule lets us discard any conjunct v that we regard as no longer useful.

- Distribution rule:

  $$u, v_1 \vee \ldots \vee v_k \Rightarrow (u \wedge v_1) \vee \ldots \vee (u \wedge v_k).$$

b. The resolution rule

We write u(v) to indicate that v occurs in u, and then u(w) denotes the result of replacing exactly one occurrence of v by w in u. Similarly, $u[v]$ indicates that if v occurs in u then $u[w]$ denotes the result of replacing all occurrences of v by w in u.

The nonclausal resolution rule for classical propositional logic is:

$$A\langle u,\ldots,u\rangle,\ B\langle u,\ldots,\ u) \mapsto A(\ true) \vee B\langle false\rangle.$$

That is, if the formulas $A\langle u,\ldots, u)$ and $B\langle u,\ldots,$ u) have a common subformula u, then we can derive the resolvent $A\langle true\rangle \vee B\langle false\rangle$. This is obtained by substituting *true* for certain (one or more) occurrences of u in $A\langle u,\ldots,$ u), and *false* for certain occurrences of u in $B\langle u,\ldots,$ u), and taking the disjunction of the results.

In propositional modal logics, this rule is not sound. For instance, consider the formula (u A $\Diamond \neg u)$; it is satisfied by any model where u holds in the real world and fails in some possible world. We cannot soundly deduce (u $\wedge \Diamond \neg u$ A $(\Diamond \neg true \vee false))$, *as* the rule would suggest, since this formula is unsatisfiable. The problem is that while u occurs in both $\Diamond \neg u$ and u, it does not need to have the same truth value in all contexts. Intuitively, different occurrences of u may refer to u at different worlds.

The resolution rule is sound in propositional modal logics under the following ***same-world*** restriction:

> The occurrences of u in *A* or *B* that are replaced by *true* or *false,* respectively, are not in the scope of any $\Box$ or $\Diamond$ in *A* or *B*.

Informally, this imposes that all the occurrences of u under consideration are evaluated in the same world.

c. Modality rules

These rules deal with formulas in the scope of modal operators. For each modal logic there is a set of modality rules:

- K:

$$\Box u,\ \Diamond v \ \mapsto\ \Diamond(u \wedge v).$$

- T:

$\Box u,\ \Diamond v \ \mapsto\ \Diamond(u \wedge v),$ ❑♦ ↦ ♦📬

- K4:

$$\Box u,\ \Diamond v \ \mapsto\ \Diamond(u \wedge v), \qquad \Box u,\ \Diamond v \ \mapsto\ \Diamond(\Box u \wedge v).$$

- s4:

$\Box u,\ \Diamond v \ \mapsto\ \Diamond(\Box u \wedge v),$ ❑♦ ↦ ♦📬

. s5:

$$\Box u,\ \Diamond v \ \mapsto\ \Diamond(\Box u \wedge v), \qquad \Box u \ \mapsto\ u,$$

$$\Diamond u,\ \Diamond v \ \mapsto\ \Diamond(\Diamond u \wedge v), \qquad \mathrm{u} \ \mapsto\ \Diamond u.$$

- D:

$\Box u, \Diamond v \mapsto \Diamond(u$ A v) , $\mapsto \Diamond$ *true.*

- D4:

$\Box u,\ \Diamond\ v \mapsto \Diamond(u\ \wedge\ v), \qquad \Box u,\ \Diamond v \ \mapsto\ \Diamond(\Box u \wedge v), \qquad \mapsto \Diamond$ *true.*

- ☝💻

❑ *U,OV* $\mapsto \Diamond(u \wedge \Box u \wedge v \wedge \neg \Diamond v).$

4. COMPLETENESS FOR PROPOSITIONAL SYSTEMS

Theorem: The resolution systems for propositional K, T, K4, S4, S5, D, D4, and G are complete for the corresponding classes of models.

Proof sketch: We exploit some known abstract characterizations of completeness for these logics. Specifically, *model existence lemmas* (stated in terms of *consistency properties)* ([Fill) turn out to provide simple and uniform proofs for all the systems. A consistency property is a syntactic property of sets of sentences that satisfies certain conditions depending on the logic. Typically, consistency properties have the form "is not refutable (in a given proof system)." Model existence lemmas guarantee that if a set of sentences satisfies a consistency property then all the sentences in the set are satisfiable (in fact, all the sentences are simultaneously satisfiable in some logics).

We give a proof sketch for K and point out where it should be modified to apply to the other systems. Consider restricting the proof system for K so that negation rules are applied as early as possible. It suffices to show that the restricted system is complete.

We say that a set S of sentences is *admissible* (for K) if no finite conjunction of members of S can be refuted (in the resolution system for K). More precisely, S is admissible if for all distinct $w_1, \ldots, w_k \in$ S there is a permutation $\pi : (1, \ldots, k\} \rightarrow \{1, \ldots, k\}$ such that $w_{\pi(1)} \wedge \ldots$ A $w_{\pi(k)}$ cannot be refuted (or, as we often say for simplicity, "$w_1, \ldots, w_k \in$ S cannot be refuted"). We show that admissible is a consistency property for K. To this end we check that admissible satisfies the conditions in the definition of consistency property for K:

if S is admissible and $S^{\#} = \{u | \Box\, u \in S\} \cup \{\neg u | \neg \Diamond\, u \in S\}$ then

1) S contains no proposition and its negation; *false* $\notin S$, $\neg \textit{true} \notin S$;
2) if (u A v) $\in$ S then S U {u, v} is admissible;
3) if $\neg(u$ V v) $\in$ S then S U $\{\neg u, \neg v\}$ is admissible;
4) if (u V v) $\in$ S then S U {u} is admissible or S U {v} is admissible;
5) if $\neg(u$ A v) $\in$ S then S U$\{\neg u\}$ is admissible or S U$\{\neg v\}$ is admissible;
6) if $\Diamond$ u $\in$ S then $S^{\#}$ U {u } is admissible;
7) if $\neg \Box$ u $\in$ S then $S^{\#}$ U$\{\neg u\}$ is admissible.

Thus, admissible is a consistency property for K. Hence, by the model-existence lemma for K, if S is admissible then each member of S is satisfiable. It follows (taking S = {u }) that if u cannot be refuted then u is satisfiable. Therefore, the propositional proof system for K is complete.

The completeness arguments for the other logics only differ from the one for K in the definition of consistency property that admissible needs to satisfy. ▮

5. QUANTIFIER RULES

Starting in this section, we consider the extension of the resolution systems to first-order modal logics. The propositional language is extended with quantifiers, variables, predicate symbols, and function symbols. The definition of models imposes that the **Barcan** formula (Vx. $\Box$ u(x)) $\supset$ ($\Box\ \forall x.u(x)$) and its converse ($\Box\ \forall x.u(x)$) $\supset$ (Vx. $\Box$ u(x)) are theorems of the first-order systems.

We first give four definitions:

- An occurrence of a quantifier $Q^{\forall}$ is of universal *force* if it is either a universal quantifier V and has positive polarity or an existential quantifier 3 and has negative polarity. An occurrence of a quantifier $Q^{\exists}$ is of *existential force* if it is either a universal quantifier V and has negative polarity or an existential quantifier 3 and has positive polarity.

- An occurrence of a modal operator $M^{\Box}$ is of necessary *force* if it is either $\Box$ and has positive polarity or $\Diamond$ and has negative polarity. An occurrence of a modal operator $M^{\Diamond}$ is of *possible farce* if it is either $\Box$ and has negative polarity or $\Diamond$ and has positive polarity.

This section discusses skolemization and gives some skolemization rules. Completeness of the systems does not depend on the inclusion of the skolemization rules, but the rules may sometimes give rise to short-cuts in proofs. In general, we do not rely on skolemization to eliminate quantifiers. Instead, we describe some rules to move quantifiers; we manipulate formulas with quantifiers, and, therefore, the resolution rule presented in the next section takes quantifiers into account.

a. Skolemization

In classical logic, all quantifiers can be eliminated by applications of skolemization rules. This is elegant for quantifiers of both universal and existential force, and very practical for quantifiers of existential force. The classical skolemization rule for eliminating quantifiers of existential force is:

$$\exists x.u[x] \;\Rightarrow\; u\big[f(x_1, \ldots, x_n)\big],$$

where f is a new rigid function symbol and $x, x_1, \ldots, x_n$ are all the free variables in u.

In modal logics, this rule is sound as long as u is not in the scope of any $\Box$ or $\Diamond$. Unfortunately, this rule is not sound in general. For instance, consider the formula

$$(\forall x.\ \Diamond\, p(x)) \wedge (\Box\ \exists y.\neg p(y)),$$

where p is a flexible predicate symbol. The formula is satisfied by the model $\mathcal{M}$ with $D = \{0, 1\}$, $W = \{0, 1\}$, $w_0 = 0$, $R = W^2$, where p holds for 0 only in the real world and p fails for 1 only in the real world. The rule replaces y by a new rigid constant symbol a, yielding the formula

$$(\forall x.\ \mathit{0}\ p(x)) \wedge (\Box\ \neg p(a)),$$

which is unsatisfiable. Notice that the new formula states that there is an element in the domain that has the property $\neg p$ in all possible worlds. The original sentence, on the other hand, only claimed that in each possible world there was some element with property $\neg p$. Therefore, the classical rule does not capture implicit dependencies on worlds.

A variant of the rule with flexible skolem symbols does capture implicit dependencies on worlds and soundly eliminates some quantifiers of existential force in the scope of modal operators. Consider, for instance, the formula $\Box\, \exists x$ p(x). If $\Box\, \exists x.p(x)$ holds then in each world there must be some element with property p. In each world, denote this element by

a. Thus, we may derive that for a new flexible constant symbol a, $\Box\, p(a)$ holds. More generally, flexible function symbols are introduced when free variables appear. For instance, assume $\Box\, \exists x.p(x, y)$ holds. Then, for a new flexible function symbol f, $\Box\quad p(f(y), y)$ holds. This resembles the classical method to eliminate quantifiers of existential force, with the exception that now a flexible function symbol is introduced.

We obtain a *flexible skolemization* rule of the same form as the classical skolemization rule:

$$\exists x.u[x] \;\Rightarrow\; u[f(x_1, \ldots, x_n)],$$

where f is a new flexible function symbol, x, $x_1, \ldots, x_n$ are all the free variables in u, and x does not occur in the scope of any modal operator in u.

Proposition (Soundness of flexible skolemization):

If $v\langle \exists x.u[x] \rangle$ satisfiable, f is a new flexible function symbol, x, $x_1, \ldots, x_n$ are all the free variables in u, x does not occur in the scope of any modal operator in u, and $\exists x.u[x]$ occurs positively in v,
then $v\langle u[f(x_1, \ldots, x_n)] \rangle$ is also satisfiable.

The rule is not always satisfactory when x occurs in the scope of modal operators in u. For instance, the formula

$$\Box\, \exists x. (p(x) \wedge \Diamond\, p(x))$$

yields

$$\Box\, (p(a) \wedge \Diamond\, p(a))$$

for a flexible constant symbol a. The original formula is stronger than the one we deduce: the original formula asserts that for each world the same x satisfies $p(x)$ in the real world and in some possible world. On the other hand, since a is world-dependent, the formula $\Box\, (p(a) \wedge \Diamond\, p(a))$ does not guarantee that the **same** element of the domain has property p in the real world and in some possible world.

Instead, we could deduce the formula

[illegible] $a \supset (p(x) \wedge \Diamond\, q(x))]$.

This formula is as strong as the original one. Note that it involves a V instead of a 3.

This suggests how to eliminate all quantifiers of existential force. The price paid is that the deduced formulas involve some new equations and some new quantifiers of universal force. The general rule is

$$\exists x.u \Rightarrow \forall x.[x = f(x_1, \ldots, x_n) \supset u],$$

where f is a new flexible function symbol and $x, x_1, \ldots, x_n$ are all the free variables in u.

Proposition (Soundness of generalized flexible skolemization):

If $v\langle \exists x.u \rangle$ is satisfiable, f is a new flexible function symbol, $x, x_1, \ldots, x_n$ are all the free variables in u, and $\exists x.u$ occurs positively in v,
then $v\langle \forall x.(x = f(x_1, \ldots, x_n) \supset$ u)) is also satisfiable.

b. Quantifier extraction rules

The quantifier extraction rules move quantifiers to the outside of formulas. We can always extract quantifiers of universal force:

$$u\langle Q^{\forall}x.v[x]\rangle \;\Rightarrow\; \forall x'.u\langle v[x']\rangle,$$

where x' is a new variable. ($Q^{\forall}$ is V or 3, whichever is of universal force in the context under consideration.)

Proposition (Soundness of $Q^{\forall}$ rule):

$$u\langle Q^{\forall}x.v[x]\rangle \hookrightarrow \forall x'.u\langle v[x']\rangle.$$

Sometimes we can extract quantifiers of existential force in a similar way:

$$u\langle Q^{\exists}x.v[x]\rangle \;\Rightarrow\; \exists x'.u\langle v[x']\rangle,$$

where x' is a new variable. The rule is restricted so that dependencies on other variables and implicit dependencies on worlds are not overlooked: the replaced occurrence of $Q^{\exists}x.v[x]$ should not occur in the scope of any quantifier of universal force or modal operator of necessary force in u.

Proposition (Soundness of $Q^{\exists}$ rule):

If the replaced occurrence of $Q^{\exists}x.v[x]$ is not in the scope of any quantifier of universal force or modal operator of necessary force in u,
then $u\langle Q^{\exists}x.v[x]\rangle \hookrightarrow \exists x'.u\langle v[x']\rangle$.

6. AUXILIARY RULES

a. Rigid symbols and the frame rules

It is convenient to include rigid symbols for world-independent functions and predicates in the first-order modal language. The frame rules reflect the fact that the meanings of these symbols do not depend on the world where they are evaluated:

if u is a formula with no occurrences of flexible symbols, then

$$\Diamond u \;\mapsto\; u \quad \text{and} \quad u \;\mapsto\; \Box u.$$

For instance, if p is a rigid proposition symbol. then $\Diamond\, p$ can yield p, and then $\Box\;\; p$.

b. Equality

As in classical logic, we can add axioms for the equality symbol. Alternatively, we can include an extension of paramodulation or E-resolution (see [MW2]).

c. The cut rule

The cut rule is

$$\mapsto u \vee \neg u.$$

Note that the cut rule requires heuristics to choose u. This may be impractical in fully automatic systems. On the other hand, the cut rule is quite convenient in interactive settings, where a user may suggest appropriate u's to obtain shorter proofs.

This rule is not essential for completeness for the propositional modal systems, but it is essential in the first-order systems. Other first-order modal systems include similar devices. In fact, there exists proof-theoretic evidence that some rule like the cut rule is necessary for the logics in question ([Fill).

7. THE RESOLUTION RULE

In subsections a, b, and c we describe a unification algorithm and a resolution rule for first-order modal logics. For the sake of simplicity, the language is temporarily restricted not to contain flexible function symbols. In subsection d this restriction is abandoned.

a. Unification

We extend the classical unification algorithm to handle formulas with modal operators and quantifiers. Suppose we have one of the usual recursive definitions of the function unifier to compute most-general unifiers of classical quantifier-free expressions. Two clauses are added to the recursive definition, one for modal operators and one for quantifiers.

- *Modality extension:* Let M be a modal operator.

$$\mathit{unifier}(Mu_1, \ldots, Mu_m) \text{ is } \begin{cases} \mathit{unifier}(u_1, \ldots, u_m) & \text{if it exists} \\ \text{fail} & \text{otherwise} \end{cases}$$

In other words, □ and ◇ are treated just like unary connectives as far as unification is concerned.

- Quantifier extension: Let Q be a quantifier and x' a new variable.

$$\mathit{unifier}(Qx_1 .u_1[x_1], \ldots, Qx_m.u_m[x_m])$$

$$\text{is } \begin{cases} \mathit{unifier}(u_1[x'], \ldots, u_m[x']) & \text{if it exists and does not bind } x' \\ \text{fail} & \text{otherwise} \end{cases}$$

For instance, $\forall x.p(x)$ and $\forall y.p(y)$ unify because $p(x')$ unifies with itself and the unifier (the empty substitution) does not bind x'. On the other hand, $\forall x.p(a)$

and $\forall y.p(y)$ do not unify, since the most-general unifier of *p(a)* and $p(x')$ binds x' to a. The formulas $\forall x.p(x)$ and p(y) do not unify: the main operator of the latter formula is not a V.

These additions to the recursive definition of *unifier* are simple enough that most-general unifiers can still be computed when unifiers exist at all.

b. The resolution rule

The classical nonclausal resolution rule can be written

$$A\langle v_1, \ldots, v_n\rangle, B\langle v_{n+1}, \ldots, v_m\rangle \mapsto A\theta\langle true\rangle \vee B\theta\langle false\rangle$$

where θ is a most-general unifier of $v_1, \ldots, v_m$ and replaces only variables that are (implicitly) universally quantified ([MW1]). As might be expected, the classical rule is not sound for formulas with quantifiers, modal operators, and flexible symbols.

Since we do not rely on skolemization and the quantifier extraction rules only shift quantifiers outwards, the modal nonclausal resolution rule should handle quantifiers in front of *A* and *B*. Also, the conclusion of the resolution rule, $A\theta\langle true\rangle \vee B\theta\langle false\rangle$, may be preceded by some quantifiers (obtained by mixing those in front of *A* and *B).* Moreover, the formulas *A, B,* and $A\theta\langle true\rangle \vee B\theta\langle false\rangle$ may contain quantifiers. Some restrictions guarantee that the presence of quantifiers does not make the rule unsound. Other restrictions deal with flexible symbols and modal operators.

The rule is:

$$\begin{aligned} &Q_1x_1 \ldots Q_hx_h.A\langle v_1, \ldots, v_n\rangle, \quad R_1y_1 \ldots R_ky_k.B\langle v_{n+1}, \ldots, v_m\rangle \\ &\quad \mapsto S_1z_1 \ldots S_{h+k}z_{h+k}.\left[A\theta\langle\ true)\ \vee B\theta\langle false\rangle\right] \end{aligned}$$

where θ is a most-general unifier of $v_1, \ldots, v_m$ and $Q_1, \ldots, Q_h, R_1, \ldots, R_k, S_1, \ldots, S_{h+k}$ are quantifiers, under the restrictions:

(i) The variables $x_1, \ldots, x_h, y_1, \ldots, y_k$ are all different.

(ii) The sequence $S_1z_1 \ldots S_{h+k}z_{h+k}$ is a merge of $Q_1x_1 \ldots Q_hx_h$ and $R_1y_1 \ldots R_ky_k$, that is, $Q_1x_1 \ldots Q_hx_h$ and $R_1y_1 \ldots R_ky_k$ are subsequences of $S_1z_1 \ldots S_{h+k}z_{h+k}$.

(iii) The same-world restriction: If the replaced occurrences of $v_1\theta, \ldots, v_m\theta$ are in the scope of any modal operator in $A\theta$ or $B\theta$ then $v_1\theta, \ldots, v_m\theta$ contain only rigid symbols.

(iv) The replaced occurrences of $v_1\theta, \ldots, v_m\theta$ are not in the scope of any quantifier in *A8* or $B\theta$.

(v) If (x ← *t)* $\in \theta$ then for some i, 1 $\leq$ i $\leq h+k$, S_i = V, z_i = x, and no variable in *t* occurs bound in $\forall x S_{i+1}z_{i+1} \ldots S_{h+k}z_{h+k}.$ *(A* V*B).*

Once all the restricticion are checked, redundant quantifiers in $S_1 z_1 \ldots S_{h+k} z_{h+k}$ may be discarded.

Restriction (iii) is necessary for modal logics, even at the propositional level. On the other hand, restrictions (i), (ii), (iv), and (v) are intended to solve classical logic problems; some of them are actually related to restrictions described by Manna and Waldinger ([MW3]) for resolution with quantifiers in classical logic. Restriction (v) is intended to enforce that the application of θ does not cause any capture of free variable, that θ only instantiates universally quantified variables, and that if $(x \leftarrow t) \in \theta$ then t does not depend on x implicitly.

Example: When we apply the resolution rule to

$$\exists x_1 \forall x_2 \exists x_3 . (\Diamond p(x_1, x_2) \vee q(x_2, x_3))$$
$$A$$
$$\exists y_1 \forall y_2 . \neg q(y_1, \mathrm{Y2}).$$

with

$$A = \neg q(y_1, y_2) \text{ and } B = (\Diamond p(x_1, x_2) \vee q(x_2, x_3)),$$
$$v_1 = q(y_1, y_2) \text{ and } v_2 = q(x_2, x_3),$$
$$\theta = \{x_2 \leftarrow y_1, y_2 \leftarrow x_3\},$$

restrictions (i), (iii), and (iv) are satisfied.

To satisfy the remaining restrictions, we choose

$$\exists x_1 \exists y_1 \forall x_2 \exists x_3 \forall y_2 . [\neg \mathit{true} \vee (\Diamond p(x_1, y_1) \vee \mathit{false})]$$

as the derived formula. We delete redudant quantifiers to obtain

$$\exists x_1 \exists y_1 . [\neg \mathit{true} \vee (\Diamond p(x_1, y_1) \vee \mathit{false})].$$

Simplification yields

$$\exists x_1 \exists y_1 . \ \Diamond p(x_1, y_1). \quad \mathrm{I}$$

Example: Whether the resolution rule is applicable or not can be extremely sensitive to the order of the quantifiers in the premises. For instance, suppose we change the formula in the previous example to

$$\exists x_1 \forall x_2 \exists x_3 . (\Diamond p(x_1, x_2) \vee q(x_2, x_3))$$
$$\wedge$$
$$\forall y_2 \exists y_1 . \neg q(y_1, y_2)$$

and take

$$\begin{aligned}
&A = \neg q(y_1, y_2) \quad \text{and} \quad B = (\Diamond p(x_1, x_2) \lor q(x_2, x_3)),\\
&v_1 = q(y_1, y_2) \quad \text{and} \quad v_2 = q(x_2, x_3),\\
&\theta = \{\mathrm{X2} \leftarrow y_1, y_2 \leftarrow x_3\}.
\end{aligned}$$

Restrictions (i), (iii), and (iv) are still satisfied, but it is not possible to satisfy restrictions (ii) and (v) simultaneously. For instance, if we derive the formula

$$\exists x_1 \forall y_2 \exists y_1 \forall x_2 \exists x_3 . [\neg \mathit{true} \lor (\Diamond p(x_1, y_1) \lor \mathit{false})]$$

restriction (v) is not satisfied: $(y_2 \leftarrow x_3) \in \theta$ and x_3 is bound in

$$\forall y_2 \exists y_1 \forall x_2 \exists x_3 . [\neg q(y_1, y_2) \lor (\Diamond p(x_1, x_2) \lor q(x_2, x_3))].$$

Other formulas we may want to derive give rise to similar restriction violations. ▮

c. Merging the quantifiers

The resolution rule does not explicitly specify the order of $S_1 z_1, \ldots, S_{h+k} z_{h+k}$. A method for obtaining the sequence $S_1 z_1 \ldots S_{h+k} z_{h+k}$ is based on systematically merging the sequences $Q_1 x_1 \ldots Q_h x_h$ and $R_1 y_1 \ldots R_k y_k$ in different ways, until one of the results satisfies all the restrictions at once. Fortunately, there are less expensive implementations.

For instance, the one sketched here is based on choosing a partial order for the quantifiers and then running a topological sort. As a preliminary step, we check that conditions (i), (iii), and (iv) are satisfied. Then we build a directed graph with nodes labelled by the quantifiers from the premises of the rule, that is, $S_1 z_1, \ldots, S_{h+k} z_{h+k}$. There is an edge from $S_i z_i$ to $S_j z_j$ if $(z_j \leftarrow t\langle z_i \rangle) \in e$ for some term t or if $S_j z_j$ is in the scope of $S_i z_i$ in either of the premises' quantifier sequences, $Q_1 x_1 \ldots Q_h x_h$ and $R_1 y_1 \ldots R_k y_k$. An edge from $S_i z_i$ to $S_j z_j$ can be interpreted as expressing that z_j depends on z_i and implies that $S_i z_i$ should occur to the left of $S_j z_j$ in the formula derived by the rule.

If the graph is cyclic, the rule is not applicable. Otherwise, the graph can be mapped into a string by a topological sort. The output string is just $S_1 z_1 \ldots S_{h+k} z_{h+k}$. When arbitrary choices are possible, it is convenient to place 3's close to the source (that is, to the left in $S_1 z_1 \ldots S_{h+k} z_{h+k}$) in order to get a stronger conclusion. This construction respects the original order of the quantifiers and dependencies; therefore, restrictions (ii) and (v) are satisfied. Finally, redundant quantifiers may be discarded in the derived formula.

Example: The graph for the first example above is

$$\begin{tikzcd}
\exists x_1 \arrow[r] & \forall x_2 \arrow[r] & \exists x_3 \arrow[d] \\
 & \exists y_1 \arrow[u] \arrow[r] & \forall y_2
\end{tikzcd}$$

It can be flattened into the string

$$\exists x_1 \longrightarrow \exists y_1 \longrightarrow \forall x_2 \longrightarrow \mathit{3x3} \longrightarrow \mathit{vyp.} \quad I$$

Example: The graph for the second example is

$$\begin{tikzcd}
\exists x_1 \arrow[r] & \forall x_2 \arrow[r] & \exists x_3 \arrow[d] \\
& \exists y_1 \arrow[u] & \forall y_2 \arrow[l]
\end{tikzcd}$$

The resolution rule is not applicable because the graph is cyclic. ▮

d. Resolution with flexible function symbols

In the presence of flexible function symbols, a new restriction on the resolution rule is necessary. The following examples show that the current rule is not sound for formulas with flexible function symbols.

Example: Consider the formula

$$u \ : \ (\forall x. \neg \Box p(x)) \wedge \Box p(a),$$

where a and p are flexible. The formula u is satisfied by the model M with $D = \{0,1\}$, $W = \{0,1\}$, $w_0 = 0$, $R = W^2$, where a has value 0 in the real world and 1 elsewhere, p holds for 0 only in the real world, and p fails for 1 only in the real world. Take $A = \neg \Box p(x), B$ ▮❏●❐☎♋◗⌂ v_1 ▮ $\Box p(x), v_2 = \Box p(u)$. The most-general unifier of v_1 and v_2 is $\theta =$ {x t a}. With the restrictions we have presented so far, the resolution rule allows us to deduce

$$(\forall x. \neg \Box p(x)) \text{ A } \Box p(a) \text{ A } (\neg \mathit{true} \vee \mathit{false}).$$

Simplification yields *false.* According to this proof, u is unsatisfiable. ▮

**Example:* Consider the formula

$$u \ : p(u) \ A \ \Box p(a) \wedge \forall x. (\neg p(x) \vee \Diamond \neg p(x)) ,$$

where a and p are flexible. The model M described in the previous example satisfies u. Take $A = (\text{-}p(x) \vee \Diamond \neg p(x))$, $B = p(a)$, $v_1 = p(x)$, $v_2 = p(a)$. The most-general classical unifier of v_1 and v_2 is $\theta =$ {x t *a}.* With the restrictions we have presented so far, the resolution rule allows us to deduce

$$p(a) \wedge \Box p(a) \wedge \forall x. (\neg p(x) \ v \ \Diamond \neg p(x)) \wedge \left[(\neg \mathit{true} \vee \Diamond \neg p(a)) \vee \mathit{false} \right].$$

Simplification yields

$$\Box p(a) \wedge \Diamond \neg p(a),$$

a clearly provably unsatisfiable formula. According to this proof, then, u is unsatisfiable.
▮

Unification in the scope of modal operators and substitution into the scope of modal operators give rise to incorrect derivations in these examples. The basic problem is simply that equals cannot be substituted for equals in modal logics. The resolution rule is restricted further in order to avoid this problem:

(vi) If (x t t) $\in \theta$ and a flexible symbol occurs in t then x does not occur in the scope of any modal operator in either A or B.

e. Soundness of resolution

The restrictions presented in the last two subsections are actually sufficient to guarantee the soundness of the resolution rule. We first show:

Lemma (Soundness of instantiation):

Given the substitution θ, the quantifiers $T_1, \ldots, T_\ell$, and v = $T_1 w_1 \ldots T_\ell w_\ell . u$, and $v' = T_1 w_1 \ldots T_\ell w_\ell . u\theta$, such that

if (x $\leftarrow$ t) $\in \theta$ then for some i, 1 $\leq$ i $\leq \ell$, T_i = $\forall$, w; = x, and no variable in t occurs bound in $\forall x T_{i+1} w_{i+1} \ldots T_\ell w_\ell . u$,

if (x $\leftarrow$ t) $\in \theta$ and t contains flexible symbols then x does not occur in the scope of any modal operator in u,

then v $\hookrightarrow v'$.

Theorem: The resolution rule, with restrictions (i), (ii), (iii), (iv), (v), and (vi), is sound.

Proof sketch: It suffices to show that the premises entail the conclusion, that is,

$$Q_1 x_1 \ldots Q_h x_h . A\langle v_1, \ldots, v_n \rangle \wedge R_1 y_1 \ldots R_k y_k . B\langle v_{n+1}, \ldots, v_m \rangle$$
$$\hookrightarrow S_1 z_1 \ldots S_{h+k} z_{h+k} . \left[A\theta\langle \mathit{true} \rangle \vee B\theta\langle \mathit{false} \rangle \right].$$

Assume the premises $Q_1 x_1 \ldots Q_h x_h . A$ and $R_1 y_1 \ldots R_k y_k . B$ hold. Conditions (i) and (ii) guarantee that the (sound) quantifier rules allow us to derive $S_1 z_1 \ldots S_{h+k} z_{h+k} . (A$ A $B)$. This formula and θ fulfill the hypotheses of the lemma by conditions (v) and (vi). Therefore, we can derive $S_1 z_1 \ldots S_{h+k} z_{h+k} . (A \wedge B)\theta$, that is, $S_1 z_1 \ldots S_{h+k} z_{h+k} . (A\theta \wedge B\theta)$. (At this point redundant quantifiers can be deleted from the conclusion without harm.)

We have shown that

$$Q_1 x_1 \ldots Q_h x_h . A \wedge R_1 y_1 \ldots R_k y_k . B$$
$$\hookrightarrow S_1 z_1 \ldots S_{h+k} z_{h+k} . \left(A\theta\langle v_1, \ldots, v_n \rangle \wedge B\theta\langle v_{n+1}, \ldots, v_m \rangle \right).$$

It suffices to show that

$$S_1 z_1 \ldots S_{h+k} z_{h+k}.(A\theta\langle v_1, \ldots, v_n\rangle \text{ A } B\theta\langle v_{n+1}, \ldots, v_m\rangle) \\ \hookrightarrow S_1 z_1 \ldots S_{h+k} z_{h+k}.\big[A\theta\langle \mathit{true}\rangle \vee B\theta\langle \mathit{false}\rangle\big].$$

This can be proved by purely propositional modal reasoning: by the monotonicity of entailment lemma, it suffices to show that

$$\big(A\theta\langle v_1, \ldots, v_n\rangle \text{ A } B\theta\langle v_{n+1}, \ldots, v_m\rangle\big) \hookrightarrow \big[A\theta\langle \mathit{true}\rangle \ v \ B\theta\langle \mathit{false}\rangle\big].$$

The formulas $A\theta$ and $B\theta$ have some subformulas in common, since $v_1\theta = \ldots = v_m\theta$. Let $v\theta$ denote $v_1\theta, \ldots, v_m\theta$. Consider occurrences of $v\theta$ not in the scope of any quantifier and, if $v\theta$ contains any flexible symbols, not in the scope of any modal operator. Assume that $A\theta\langle v_1, \ldots, v_n\rangle$ and $B\theta\langle v_{n+1}, \ldots, v_m\rangle$ hold. If $v\theta$ is true then $A\theta\langle \mathit{true}\rangle$ holds; otherwise, $B\theta\langle \mathit{false}\rangle$ holds. In either case, $A\theta\langle \mathit{true}\rangle \vee B\theta\langle \mathit{false}\rangle$ holds, as we wanted to show. ▮

8. AN EXAMPLE

We prove that

❑ *WP(z))* $\supset (\forall x.\Box\, p(x))$

in the resolution system for K. We will derive *false* from

♦□ 💻 ¬ [¬ ❑ *(Vx.p(x))*$\vee(\forall x.\Box\, p(x))]$

By the negation rules, we first get

❑ *('dx.p(x))*$\wedge(\exists x.\Diamond\neg p(x))$.

The rule for moving quantifiers of existential force yields

3x'. $\big[\Box(\forall x.p(x)) \text{ A } \Diamond\neg p(x')\big]$.

The modality rule in the system for K yields

3x'. $\big[\Box(\forall x.p(x))\ A\ \Diamond\neg p(x') \wedge \Diamond\big((\forall x.p(x)) \wedge \neg p(x')\big)\big]$.

Weakening reduces this sentence to

3x'. $\Diamond\big[(\forall x.p(x)) \text{ A } \neg p(x')\big]$.

Take $A = \neg p(x')$, $B = p(x)$, $v_1 = p(x')$, $v_2 = p(x)$. Resolution yields

3x'. $\Diamond\big[(\forall x.p(x)) \text{ A } \neg p(x')\ A\ (\neg \mathit{true} \vee \mathit{false})\big]$.

true-false simplifications yield *false.*

9. COMPLETENESS FOR FIRST-ORDER SYSTEMS

Our propositional modal resolution systems together with the quantifier rules, the auxiliary rules, and the resolution rule for the first-order language with flexible function symbols, constitute first-order resolution systems. Skolemization rules may be added, but are not essential.

Theorem: The first-order resolution systems for K, T, K4, S4, S5, D, and D4 are complete for the corresponding classes of models.

Proof sketch: Some Hilbert systems are known to be complete for these logics, at least for the language with no rigid symbols and no function symbols (e.g., [HC], [Fill). We can extend these completeness results to the language with rigid symbols and function symbols. Then we show that each of the resolution systems is at least as powerful as one such complete Hilbert system. Specifically, we show that any Hilbert proof can be transformed into a resolution proof, by induction on the structure of Hilbert proofs. ∎

Remark: We will not discuss completeness issues for first-order G. Several notions of completeness have been proposed for this logic and none of those based on Kripke models seems fully satisfactory ([Fi2]).

Acknowledgements:

We are grateful to Bengt Jonsson and John Lamping for critical reading of the manuscript.

REFERENCES

[AMl] M. Abadi and Z. Manna, "Nonclausal temporal deduction," in *Logics of Programs* (R. Parikh, ed.), Springer-Verlag LNCS 193, 1985, pp. 1-15.

[AM2] M. Abadi and Z. Manna, "A timely resolution," Proceedings of the Symposium on Logic in Computer Science (LICS), 1986.

[Fal] L. Fariñas del Cerro, "Temporal reasoning and termination of programs," Eighth International Joint Conference on Artificial Intelligence, 1983, pp. 926-929.

[Fa2] L. Fariñas del Cerro, "Un principe de r&solution en logique modale," *RAIRO Informatique Théorique*, Vol. 18, No. 2, 1984, pp. 161-170.

[Fa3] L. Fariñas del Cerro, "Resolution modal logics," in *Logics and Models of Concurrent Systems* (K.R. Apt, ed.), Springer-Verlag, Heidelberg, 1985, pp. 27-55.

[Fil] M. Fitting, *Proof Methods for Modal and Intuitionistic Logics*. D. Reidel Publishing Co., Dordrecht, 1983.

[Fi2] M. Fitting, private communication.

[GK] C. Geissler and K. Konolige, "A resolution method for quantified modal logics of knowledge and belief," in *Theoretical Aspects of Reasoning about Knowledge* (J.Y. Halpern, ed.), Morgan Kaufmann Publishers, Palo Alto, 1986, pp. 309-324.

[HC] G.E. Hughes and M.J. Cresswell, *An Introduction to Modal Logic,* Methuen & Co., London, 1968.

[HM] J.Y. Halpern and Y. Moses, "Knowledge and Common Knowledge in a Distributed Environment ," Third ACM Conference on the Principles of Distributed Computing, 1984, pp. 50-61. A revised version appears as IBM RJ 4421, 1984.

[Ko] K. Konolige, *A Deduction Model of Belief* and its *Logics*, Ph.D. Thesis, Computer Science Department, Stanford University, 1984.

[MC] D. McDermott, "Nonmonotonic Logic II: Nonmonotonic Modal Theories," *Journal of the ACM,* Vol. 29, No. 1, Jan. 1982, pp. 33-57.

[Mu] N.V. Murray, "Completely nonclausal theorem proving," *Artificial Intelligence,* Vol. 18, No. 1, January 1982, pp. 67-85.

[MW1] Z. Manna and R. Waldinger, "A deductive approach to program synthesis," *ACM Transactions on Programming, Languages, and Systems,* Vol. 2, No. 1, Jan. 1980, pp. 90-121.

[MW2] Z. Manna and R. Waldinger, "Special relations in automated deduction," *Journal of the ACM,* Vol. 33, No. 1, Jan. 1986, pp. 1-59.

[MW3] Z. Manna and R. Waldinger, "Special relations in program-synthetic deduction," Report No. STAN-CS-82-902, Computer Science Department, Stanford University, March 1982.

[P] A. Pnueli, "The temporal logic of programs," 18th Annual Symposium on Foundations of Computer Science, 1977, pp. 46-57.

[R] J.A. Robinson, "A machine-oriented logic based on the resolution principle," *Journal of the ACM,* Vol. 12, No. 1, January 1965, pp. 23-41.

[W] P. Wolper, "Temporal Logic can be more expressive," 22nd Annual Symposium on Foundations of Computer Science, 1981, pp. 340-348.