April 1986

Report No. STAN-CS-86-1106

A Timely Resolution

by

Martin hbadi and Zohar Manna

Department of Computer Science

Stanford University
Stanford. CA 94305

A TIMELY RESOLUTION

Martin Abadi and Zohar Manna

Computer Science Department
Stanford University

We present a novel proof system **R** for First-order (Linear) Temporal Logic. This system extends our Propositional Temporal Logic proof system ([AM]). The system **R** is based on nonclausal resolution; proofs are natural and generally short. Special quantifier rules, unification techniques, and a resolution rule are introduced. We relate **R** to other proof systems for First-order Temporal Logic and discuss completeness issues. The system **R** should be useful as a tool for such tasks as verification of concurrent programs and reasoning about hardware devices.

1. INTRODUCTION

Temporal Logic ([Pn]) has been proposed as a framework to describe and reason about sequences of states. In particular, it is useful for specification (e.g., [L], [HO]), verification (e.g.7 [MP2], [OL]), and synthesis (e.g., [MWo], [CE]) 0f concurrent systems, as well as for synthesis of robot plans (e.g., [G]) and for verification of hardware devices (e.g., [M]).

In spite of the wide range of applications of Temporal Logic, proof techniques for Temporal Logic, especially for First-order Temporal Logic (**FTL**), are quite limited. A number of proof systems for Propositional Temporal Logic (**PTL**) have been proposed and shown to be complete. Most **PTL** systems are based on either tableaux (e.g., [W]) or on Hilbert and Gentzen proof techniques (e.g., [GPSS], [B]). Plaisted's tableau system ([Pl]) can handle certain first-order theories. Manna and Pnueli ([MP1]) suggested the extension of a **PTL** Hilbert system to **FTL**. The system uses modus ponens as the main inference rule and is therefore inadequate as an automatic or semi-automatic proof system.

Recently, Cavalli and Fariñas del Cerro ([Ca], [CF]) described a resolution system for **PTL** which provides a reasonable basis for theorem-proving. However, completeness is only

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 Symposium on Logic in Computer Science, Cambridge, MA, June 1986.

shown for a **PTL** fragment with the modalities $\bigcirc$ ("next"), $\Box$ ("always"), $\Diamond$ ("eventually"); the completeness proof does not seem to carry over to **PTL** with the more general operators $\mathcal{U}$ ("until") and $\mathcal{P}$ ("precedes"). The system is clausal and therefore requires that formulas be paraphrased into unnatural and long clausal forms. Venkatesh ([V]) also-proposed a similar clausal resolution approach.

In an earlier paper ([AM]), we presented complete nonclausal resolution systems for **PTL** with the modalities $\bigcirc$, $\Box$, $\Diamond$, and also with $\mathcal{U}$ and $\mathcal{P}$. In this paper, we generalize our **PTL** resolution system into an efficient proof system **R** for **FTL** and study its soundness and completeness. The system handles arbitrary formulas in **FTL;** they do not have to be in clausal form and may include the operators $\mathcal{U}$ and $\mathcal{P}$. While **R** can be used as a proof system by itself, special purpose rules as well as known decision procedures for fragments of **FTL,** such as Plaisted's, can be built into **R.**

The system **R** includes rules for propositional temporal reasoning, equality axioms or rules, auxiliary rules to move quantifiers, and a generalization of the classical resolution rule that treats quantifiers explicitly. Skolemization rules to remove quantifiers may be included in **R.** They are not essential to the completeness of **R,** but they are sometimes convenient to use.

In the next section we introduce the syntax and semantics of **FTL** informally and define the general notions of proof and rule. In section 3, we review the basic rules of **R** for **FTL,** mainly following our earlier rules for **PTL.** In sections 4 and 5, we present the rest of the **FTL** system, describing rules for quantifiers and the resolution rule. Section 6 contains an example. In section 7 we relate **R** to other proof systems for **FTL** and discuss completeness issues.

2. PRELIMINARIES

a. The language

The language of **FTL** is that of the predicate calculus with equality, with additional modal operators. For simplicity, we assume that the only connectives are $\neg$, A, V, and regard all other connectives as abbreviations. The modal operators we consider are the usual ones for discrete linear time: $\bigcirc$, $\Box$, $\Diamond$, and the more general $\mathcal{U}$ and $\mathcal{P}$. Formulas need not be in clausal form.

For formulas u and v,

- $\bigcirc$ u means "u is true in the next state";
- $\Box$ u means "u is always true (from now on)“;
- $\Diamond$ u means "u is eventually true"; that is, $\Diamond u \equiv \neg \Box \neg u$;
- $u\mathcal{U}v$ means "*u* is true until v is true"; in particular, u is true forever if v is never true (therefore, $\mathcal{U}$ is often called "weak until" or "unless");

- $u\mathcal{P}v$ means "u precedes v"; that is, $(u\mathcal{P}v) \equiv \neg((\neg u)\mathcal{U}v)$.

Predicate and function symbols are either ***flexible*** (time-dependent) or rigid (time-independent). Thus, if ***busy*** is a flexible unary predicate symbol and ***printer*** is a rigid constant symbol (that is, a nullary rigid function symbol),

$$busy(printer)\ A \neg \bigcirc \Box\, busy(printer)$$

expresses that the printer is busy in the initial state and not busy from there on. Note that the value for ***printer*** is the same in all states and the property of being ***busy*** may change with time.

Variables are rigid. For instance, $\exists x.(p(x)$ A $\bigcirc$ p(x)) means that the same value in the domain has property p in the initial state and in its successor. Free variables have an implicit universal quantification: u is valid if and only if $\forall x.u$ is valid.

b. Proofs

We write ⊢ w to mean that the FTL formula w is provable by refutation resolution, i.e., 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 derived from S_i by one of the rules of the system. We refer to $S_0, \ldots, S_n$ as a ***proof.***

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 also satisfiable.

c. Rules

Our proof system contains two types of rules: simplification rules and deduction rules. Both simplification and deduction rules may be constrained by side conditions to guarantee their soundness.

- ***Simplification*** rules are all of the form

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

If the formulas $u_1, \ldots, u_m$ are embedded as conjuncts in some conjunction in S_i (order is irrelevant), then we delete an occurrence of each of them and add the derived formula v to ·the conjuction.

Examples:

- If we apply the rule $\Box$ *false* $\Rightarrow$ ***false to***

$S_i =$ ☎☎🏳 $\wedge$ ❑ *false*,$\vee q)$

we get

$S_{i+1} = ((P \wedge false) \vee q)$.

- If we apply the rule v, v ⇒ v to

  $$S_i = \Diamond((q \vee P) \wedge r \wedge (q \vee P))$$

  we get

  $$S_{i+1} = \Diamond(r \wedge (q \vee p)).$$

- ***Deduction rules*** are all of the form

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

If the formulas $u_1, \ldots, u_m$ are embedded as conjuncts in some conjunction in S_i (order is irrelevant), then the derived formula v is added to that conjunction.

Examples:

- The rule $\mapsto$ (v $\vee$ $\neg v$) lets us introduce instances of (v $\vee$ $\neg v$) anywhere; thus,

  $$S_i = (q \wedge \Diamond r)$$

  can yield

  $$S_{i+1} = (q \wedge \Diamond(r \mathrm{A} (s \vee \neg s))).$$

- If we apply the rule v $\vee$ w, $\neg v$ $\vee$ w $\mapsto$ w to

  $$S_i = \bigcirc(s \wedge (p \vee q) \wedge (\neg p \vee q))$$

  we obtain

  $$S_{i+1} = \bigcirc(s \wedge (P \vee q) \wedge (\neg p \vee q) \wedge q).$$

Deduction rules differ from simplification rules only in that the conjuncts $u_1, \ldots, u_m$ are -kept in the derived formula. In practice, however, we often delete $u_1, \ldots, u_m$ immediately after applying a deduction rule, using the weakening rule (defined in section 3).

d. Polarity and soundness

An occurrence of a subformula has ***positive polarity*** in a formula if it is embedded 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. Thus, p occurs positively and q occurs negatively in $\neg(\neg p \vee q)$. One important observation is that P reverses the polarity of its second argument (e.g., p has negative polarity in $r\mathcal{P}(q \vee p)$).

We reduce the proof search space with a ***polarity restriction:***

Simplification rules and deduction rules are applied only to positive occurrences of $u_1, \ldots, u_m$.

We say that ***u* *entails*** v (and denote it $u \hookrightarrow v$) if $u \supset v$ is valid. The following lemma provides a criterion for soundness.

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 $w \hookrightarrow w'$.

Informally, the lemma states that a formula gets "truer" as its positive subformulas get "truer" and as its negative subformulas get "falser."

As a corollary, simplification rules are sound for negative occurrences of $u_1, \ldots, u_m$ if $v \hookrightarrow u_1 \wedge \ldots \wedge u_m$; for positive occurrences, it suffices that $u_1 \wedge \ldots \wedge u_m \hookrightarrow v$. Each of the simplification rules has the property that $u_1 \wedge \ldots \wedge u_m \hookrightarrow v$, except for the skolemization rules. Thus, with the polarity restriction, the soundness of all the simplification rules but the skolemization rules is guaranteed. We will prove the soundness of the skolemization rules with separate arguments.

Deduction rules are always sound when $u_1, \ldots, u_m$ occur with negative polarity (since the given formulas $u_1, \ldots, u_m$ are kept); for positive occurrences, it suffices that $u_1 \wedge \ldots \wedge u_m \hookrightarrow v$. Each of the deduction rules has the property that $u_1 \wedge \ldots \wedge u_m \hookrightarrow v$. This suffices for the soundness of deduction rules, independently of polarity considerations.

3. BASIC RULES

In this section we present the basic rules for $\bigcirc$, $\square$, and $\diamondsuit$. The rules for $\mathcal{U}$ and P are described in our earlier paper ([AM]). Sections 4 and 5 contain the remaining rules of our FTL system R, that is, the rules for quantifiers and the resolution rule.

a. Simplification rules

- ***true-false simplification*** rules:

These rules include

$$\square\, \textit{false} \Rightarrow \textit{false}, \quad \diamondsuit\, \textit{false} \Rightarrow \textit{false}, \quad \bigcirc\, \textit{false} \Rightarrow \textit{false},$$

and the regular ***true-false*** simplification rules, such as

$$\mathit{false},\ u \Rightarrow \mathit{false}, \quad \neg \mathit{true} \Rightarrow \mathit{false}.$$

- ***Weakening*** rule:

$$u,\ v \Rightarrow u.$$

This rule allows us to delete any conjunct that is considered useless.

- ***Negation*** rules:

$$\neg \Box u \Rightarrow \Diamond \neg u, \quad \neg \Diamond u \Rightarrow \Box \neg u, \quad \neg \bigcirc u \Rightarrow \bigcirc \neg u,$$

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

- ***Distribution*** rule:

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

b. Modality rules

These are rules to handle subformulas in the scope of modal operators.

- $\Box$ rule:

$$\Box u \mapsto u \wedge \bigcirc \Box u.$$

- $\Diamond$ rule:

$$\Diamond u \mapsto u \vee \bigcirc \Diamond u.$$

- $\Box\,\Box$ rule:

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

- $\Box\,\Diamond$ rule:

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

- $\Diamond\,\Diamond$ rule:

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

- $\bigcirc\,\bigcirc$ rule:

$$\bigcirc u,\ \bigcirc v \mapsto \bigcirc(u \wedge v).$$

Two useful derived rules are:

- $\Box\,\bigcirc$ derived rule

$$\Box u,\ \bigcirc v \ \mapsto\ \bigcirc(\Box u \wedge v),$$

which is obtained from the $\Box$ and $\bigcirc\,\bigcirc$ rules, with weakening.

- $\Diamond\,\bigcirc$ derived rule

$$\Diamond u,\ \bigcirc v \ \mapsto\ u \vee \bigcirc(\Diamond u \wedge v),$$

which is obtained from the $\Diamond$ and $\bigcirc\,\bigcirc$ rules.

Due to the induction rule (presented below) most of the modality rules (in fact, all but the $\Box$, $\Diamond$, and $\bigcirc\,\bigcirc$ rules) are not essential for completeness. We include them because they often provide convenient and natural short-cuts in proofs.

c. The induction rule

The ***induction*** rule is:

$$w,\ \Diamond u \mapsto \Diamond\big(\neg u \wedge \bigcirc(u \wedge \neg w)\big) \quad \text{if} \quad \vdash \neg(w \wedge u).$$

To justify the rule informally, suppose that u and w cannot both hold at the same instant (that is, $\neg(w$ A u)). Assume that w is true in the present and u is eventually true. Then u must be false in the present; at some point u must change from false to true. Furthermore, w is false when u is true. Thus, the induction rule allows us to conclude that $\Diamond\big(\neg u$ A $\bigcirc(u$ A $\neg w)\big)$.

We frequently use a special case of the induction rule (where w = $\neg u$):

$$\neg u,\ \Diamond u \ \mapsto\ \Diamond(\neg u \wedge \bigcirc u).$$

In fact, this special case is as powerful as the general rule in presence of the following cut rule.

d. The cut rule

The cut rule is

$$\mapsto \text{ U V } \neg u.$$

While this rule is not essential for completeness for **PTL**, it is essential for **FTL**. The cut rule is quite convenient in interactive settings, where a user may suggest appropriate u's to obtain shorter proofs.

e. The frame rule

Let $\oplus$ be any string of modal operators, and u a formula with no occurrences of flexible symbols, then

$$\oplus u \quad \mapsto \quad \Box\, u.$$

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

f. Equality

Equality can be handled with the usual techniques of classical first-order logic, such as adding equality axioms or using variants of paramodulation or E-resolution (see [MWa2]).

4. QUANTIFIER RULES

We first introduce a few 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 ***permanent 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 ***eventual force*** if it is either $\Box$ and has negative polarity or $\Diamond$ and has positive polarity. An occurrence of a binary modal operator $M^{\mathcal{U}}$ is of ***until force*** if it is either $\mathcal{U}$ and has positive polarity or ⚐ and has negative polarity. An occurrence of $M^{\mathcal{P}}$ is of precedes ***force*** if it is either $\mathcal{U}$ and has negative polarity or ⚐ and has positive polarity.

- An occurrence of a formula u is in a ***permanent context*** if it is in the scope of a modal operator of permanent force, within the first argument of a modal operator of until force, or within the second argument of a modal operator of precedes force.

. In predicate calculus, we can always eliminate quantifiers by skolemization. This is very convenient, particularly in the case of quantifiers of existential force. Unfortunately, the usual skolemization rules are not sound for FTL. For example, consider the satisfiable sentence

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

where p is a flexible predicate symbol. The classical rule to eliminate quantifiers of existential force replaces x by a new rigid constant symbol a. We obtain the sentence

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

which is unsatisfiable. The problem is that the new sentence claims that there is an element in the domain that always has the property ***p***, while the original sentence only claims that at each instant of time there is some element with property ***p***. Thus, the classical skolemization rules fail to reflect implicit dependencies on time. However, if we introduce a flexible skolem constant symbol a, then the dependencies on time are captured.

We present some skolemization rules for quantifiers of existential force. They sometimes provide convenient simplifications, but are not essential for completeness. In general, we will not attempt to eliminate quantifiers and the resolution rule will handle quantifiers directly. We use auxiliary rules to move quantifiers.

a. Skolemization

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

The classical skolemization rule to eliminate quantifiers of existential force can be soundly applied at any point in the derivation process outside the scope of $\Box$, $\Diamond$, $\mathcal{U}$, and P:

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

where f is a new rigid function symbol, x, $x_1, . . . , x_n$ are all the free variables in u, and u does not occur in the scope of any modal operator other than $\bigcirc$. The intuitive justification for the rule is that if u is not in the scope of $\Box$, $\Diamond$, $\mathcal{U}$, or $\mathcal{P}$, then x does not depend on implicit time variables.

A variant of the classical skolemization rule sometimes handles formulas in the scope of modal operators. For instance, suppose $\Box \exists x.p(x)$ holds. Then there must be a sequence of values for x that makes ***p(x)*** always true. Call this sequence a. Thus, we can deduce that for a new flexible constant symbol a, $\Box p(a)$. This reflects the classical elimination of existential quantifiers, with the exception that here a flexible constant is introduced. Similarly, we introduce flexible function symbols when free variables appear. For example, assume $\Box \exists x.p(x$, y). Then, for a new flexible function symbol f, $\Box$ $p(f(y)$, ***y)*** .

Thus, we obtain a ***flexible skolemization rule*** similar to the classical skolemization rule:

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

where ***f*** is a new flexible function symbol, x, xi, . . . , 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 $v\langle\exists x.u[x]\rangle$ is 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.

When x occurs in the scope of modal operators in u, this flexible skolemization rule is no longer satisfactory. Consider

$$\Box\, \exists x.\big(p(x) \wedge \bigcirc q(x)\big).$$

The rule would derive

$$\Box\big(p(a) \wedge \bigcirc q(a)\big),$$

for a new flexible constant symbol ***a.*** The derived sentence is weaker than the original one: the original sentence meant that for each state the same x satisfies p(x) at the present state and ***q(x)*** at the next state. Because ***a*** is flexible, □ ***(p(a)*** A ○ ***q(a))*** does not guarantee that a same value in the domain has property ***p*** in the initial state and property ***q*** in the next one.

An appropriate formula to derive from

□ 3x.$\big(p(x) \wedge \bigcirc$ ***q(x))***

will be

□ Vx. $\big[x = a \supset (p(x) \wedge \bigcirc q(x))\big]$.

Thus, we introduce V when we eliminate 3.

This idea allows us to eliminate all quantifiers of existential force. However, the resulting formulas contain new quantifiers of universal force and some equalities. The ***generalized flexible skolemization rule*** is

3X.U $\Rightarrow \forall x.\big(x = f(x_1, \ldots, x_n) \supset u\big)$

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, xi,. . . , x_n are all the free variables in u, and $\exists x.u$ occurs positively in v,
>
> then $v\langle\forall x.\big(x = f(x_1, \ldots, x_n\big) \supset$ u)) is also satisfiable.

b. Auxiliary quantifier rules

If $Q^{\forall}$ is a quantifier of universal force, it can be moved outside formulas:

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

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):

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

Similarly, we move quantifiers of existential force. The rule is restricted so that dependencies on other variables and implicit dependencies on time are not overlooked:

If x' is a new variable and $Q^{\exists}$ is a quantifier of existential force not in the scope of any quantifier of universal force or in a permanent context in u then

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

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

If the occurrence of $Q^{\exists}x.v[x]$ under consideration does not occur in the scope of any quantifier of universal force or in a permanent context in u,

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

5. THE RESOLUTION RULE

a. Resolution is affected by time

For classical quantifier-free first-order logic, the nonclausal resolution rule is

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

where the substitution θ is a most-general unifier of $v_1, \ldots, v_m$ and replaces only variables that are (implicitly) universally quantified ([MWa1]).

That is, if A has subformulas $v_1, \ldots, v_n$ and B has subformulas $v_{n+1}, \ldots, v_m$, we compute a most-general substitution θ such that $v_1\theta = \ldots = v_m\theta$. We denote $v_1\theta, \ldots, v_m\theta$ by $v\theta$. Then we derive $A\theta\langle \mathit{true}\rangle \vee B\theta\langle \mathit{false}\rangle$. This is obtained by replacing certain occurrences of $v\theta$ in $A8$ with *true,* and certain ocurrences of $v\theta$ in $B\theta$ with *false,* and taking the disjunction of the results.

This rule does not carry over to FTL. One problem is that while $v\theta$ occurs in both $A\theta$ and $B\theta$, it need not denote the same truth value in all its occurrences; intuitively, each occurrence of $v\theta$ may refer to different instants of time. For example, from $\neg u$ and $\Diamond$ u we cannot soundly deduce $\neg \mathit{true} \vee \Diamond$ ***false,*** because while the hypotheses are satisfiable (e.g., by the model which makes u false now, but true otherwise), $\neg \mathit{true} \vee \Diamond$ ***false*** is always false.

As in **PTL** ([AM]), this problem is dealt with by a ***same-time restriction:***

> If any flexible symbol occurs in $v\theta$ then the occurrences of $v\theta$ in $A\theta$ and in $B\theta$ that are replaced by ***true*** and ***false,*** respectively, are all in the scope of the same number of $\bigcirc$**'s** and are not in the scope of any other modal operator in either $A\theta$ or $B\theta$.

Intuitively, this means that all occurrences of $v\theta$ refer to the same time instant. For example, consider the formulas

$$\bigcirc \neg \bigcirc(\Box p \vee q) \wedge \Diamond \Box p \quad \text{and} \quad \bigcirc \bigcirc \Box p \vee \bigcirc \Box p$$

where ***p*** is a flexible symbol. The resolution rule for **PTL** allows us to derive the resolvent

$$\left[\bigcirc \neg \bigcirc(\mathit{true} \vee q) \wedge \Diamond \Box p\right] \vee \left[\bigcirc \bigcirc \mathit{false} \vee \bigcirc \Box p\right].$$

We only substituted ***true*** or ***false*** for those occurrences of $\Box p$ in the scope of two $\bigcirc$**'s.** These occurrences are not in the scope of any $\Box$ or $\Diamond$ in either of the premises. We cannot replace the second occurrence of $\Box p$ in the first premise by ***true,*** since it is in the scope of a $\Diamond$. Also, we cannot replace the second occurrence of $\Box$ ***p*** in the second premise by ***false,*** since it is in the scope of only one $\bigcirc$.

The same-time restriction does not suffice to guarantee the soundness of the resolution rule in **FTL,** because quantifiers and flexible function symbols may appear in formulas.

We now describe an extension of the unification algorithm for **FTL.** Later we show how it can be used to obtain a sound **FTL** resolution rule. For the sake of clarity, we will temporarily assume that there are no flexible function symbols and reintroduce them in subsection d.

b. Unification

We use the classical unification algorithm with two minor extensions: a quantifier extension and a modality extension. These extensions to classical unification are superficial enough that we still obtain a most-general unifier θ when unifiers exist.

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

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

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

For example, $\forall x.p(x)$ and $\forall y.p($ y) unify because $p(x')$ unifies with itself, without binding x'. On the other hand, $\forall x.p(x)$ and Vy.p(a) do not unify because $p(x')$ and ***p(a)*** unify only by binding x' to ***a.*** Also, $\forall x.p(x)$ and ***p(a)*** do not unify because $\forall x.p(x)$ starts with a quantifier while $p(a)$ does not.

- ***Modality extension:*** Let M be any of ○, □, ◇.

$$\mathit{unifier}(Mu_1, \ldots, \textit{Mu,,,)}$$

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

In other words, ○, □, and ◇ are treated just like unary connectives as far as unification is concerned. Similarly, $\mathcal{U}$ and P are handled just like binary connectives.

c. The resolution rule

In the nonclausal resolution rule for FTL, quantifiers may appear explicitly in front of the resolved formulas A and B. The conclusion of the resolution rule will also be prefixed by a string of quantifiers (obtained by interleaving those for A and ***B).*** Furthermore, the formulas ***A,*** B, and, therefore, $[A\theta\langle \mathit{true}\rangle \vee B\theta\langle \mathit{false}\rangle]$ ma*y*contain quantifiers. The rule is:

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

where θ is a most-general unifier of $v_1, \ldots v_m$ and $Q_1, \ldots Q_h, R_1, \ldots, R_k, S_1,$ ● .-,Sh+k are quantifiers, with the following restrictions.

(i) The same-time restriction: If any flexible symbol occurs in $v\theta$ then the replaced occurrences of $v\theta$ are all in the scope of the same number of ○'s and are not in the scope of any other modal operator in either $A\theta$ or $B\theta$.

(ii) The replaced occurrences of $v\theta$ are not in the scope of any quantifier in either $A\theta$ or $B\theta$.

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

(iv) 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, $S_1z_1 \ldots S_{h+k}z_{h+k}$ has $Q_1x_1 \ldots Q_hx_h$ and $R_1y_1 \ldots R_ky_k$ as subsequences. (Redundant quantifiers in $S_1z_1 \ldots S_{h+k}z_{h+k}$ may be deleted once (v) is checked.)

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

Note that only condition (i) has to do with our working in modal logic. Conditions (ii)-(v) are concerned with classical logic problems. In fact, some of them are similar to restrictions studied in [MWa3] for resolution with quantifiers in classical logic. Condition (v) succinctly guarantees that θ only instantiates universally quantified variables; that no free variable is captured when θ is applied; and that if (x $\leftarrow t) \in \theta$ then t does not depend on x implicitly.

Example : Consider

$$\begin{array}{l}\forall x_1 \exists x_2 . \big(\neg p(x_1, x_2, a) \ \vee \ \bigcirc q(x_1)\big) \\ \qquad \wedge \\ \exists y_1 \forall y_2 . \big(p(y_1, y_2, a) \ \vee \ \diamond r(f(b))\big).\end{array}$$

We choose

$$\begin{array}{l}\mathrm{A} \quad = \quad \big(\neg p(x_1, x_2, a) \vee \bigcirc q(x_1)\big) \text{ and } B = \quad \big(p(y_1, y_2, a) \vee \diamond r(f(b))\big), \\ v_1 = p(x_1, x_2, a) \quad \text{and} \quad v_2 = p(y_1, y_2, a), \\ \theta = \{x_1 \leftarrow y_1, y_2 \leftarrow x_2\}.\end{array}$$

Conditions (i), (ii), and (iii) are clearly satisfied.

The conclusion

$$\exists y_1 \forall x_1 \exists x_2 \forall y_2 . \quad \left.\begin{array}{c}\big(\neg \mathit{true} \ \vee \ \bigcirc q(y_1)\big) \\ \vee \\ [\big(\mathit{false} \ \vee \ \diamond r(f(b))\big)\end{array}\right]$$

satisfies conditions (iv) and (v). The quantifiers $\forall x_1$, $\exists x_2$, and $\forall y_2$ are redundant. Thus, we can derive

$$\exists y_1 . \ \big[\big(\neg \mathit{true} \ \vee \ \mathbf{0} \ q(y_1)\big) \ \vee \ \big(\mathit{false} \ \vee \ \diamond r(f(b))\big)\big]$$

which simplifies to

$$\exists y_1 . \big[\bigcirc q(y_1) \vee \ \diamond r(f(b))\big].$$

Example: A slight change in the formulas in the previous example makes the resolution rule no longer applicable. Consider

$$\begin{array}{l}\forall x_1 \exists x_2 . \big(\neg p(x_1, x_2, a) \ \vee \ \bigcirc q(x_1)\big) \\ \qquad \wedge \\ \forall y_2 \exists y_1 . \big(p(y_1, y_2, a) \ \vee \ \diamond r(f(b))\big).\end{array}$$

We choose

$$\begin{array}{l}A \quad = \quad \big(\neg p(x_1, x_2, a) \vee \bigcirc q(x_1)\big) \text{ and } B = \big(p(y_1, y_2, a) \vee \diamond r(f(b))\big), \\ v_1 = p(x_1, x_2, a) \quad \text{and} \quad v_2 = p(y_1, y_2, a), \\ \theta = \{x_1 \leftarrow y_1, y_2 \leftarrow x_2\}.\end{array}$$

Conditions (i), (ii), and (iii) are clearly satisfied.

The conclusion

$$\forall x_1 \exists x_2 \forall y_2 \exists y_1 . \begin{bmatrix} (\neg \mathit{true} \vee \bigcirc q(y_1)) \\ \vee \\ (\mathit{false} \vee \Diamond r(f(b))) \end{bmatrix}$$

violates condition (v), since (xi $\leftarrow y_1) \in \theta$ and y_1 is bound in the scope of $\forall x_1$. Other possible conclusions run into similar problems.

A simple-minded implementation of the rule could be quite inefficient: while conditions (i), (ii), and (iii) are trivially handled, the sequence $S_1 z_1 \ldots S_{h+k} z_{h+k}$ is described fairly non-constructively. One could blindly build sequences with quantifiers from the premises and hope to fulfill all restrictions. However, we can suggest a more efficient approach.

After checking (i), (ii), and (iii), construct a directed graph with nodes labelled by the quantifiers from the premises, $S_1 z_1, \ldots, S_{h+k} z_{h+k}$. We put an edge from $S_i z_i$ to $S_j z_j$ if $(z_j \leftarrow t\langle z_i \rangle) \in \theta$ for some t or if $S_j z_j$ occurs in the scope of $S_i z_i$ in either $Q_1 x_1 \ldots Q_h x_h$ or $R_1 y_1 \ldots R_k y_k$. Thus, an edge from $S_i z_i$ to $S_j z_j$ denotes that z_j depends on z_i (that is, $S_i z_i$ has to appear to the left of $S_j z_j$ in the conclusion of the rule).

If possible, flatten the directed graph into a string; in other words, topologically sort the graph-the rule is applicable only when this is possible. When arbitrary choices are available, put $\exists$'s close to the source of the string (that is, to the left in $S_1 z_1 \ldots S_{h+k} z_{h+k}$) in order to get a stronger conclusion. We obtain the sequence $S_1 z_1 \ldots S_{h+k} z_{h+k}$. Since the construction respects the original order of the quantifiers and dependencies, (iv) and (v) are satisfied. Finally, delete redundant quantifiers from the conclusion.

Example: In our first example above, the graph is

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

It can be flattened into the string

$$\exists y_1 \longrightarrow \forall x_1 \longrightarrow \exists x_2 \quad - \quad \forall y_2$$

Example: In our second example, the graph is

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

Since it is cyclic, it cannot be flattened into a string. Therefore, the resolution rule is not applicable.

d. Flexible function symbols reintroduced

For terms containing flexible symbols, substitutivity of equals for equals fails in the scope of modal operators. This affects the soundness of resolution, as illustrated by the following examples.

- Unification in modal contexts:

  The formula

  $$u = \neg \Diamond p(a) \; A \; \forall x. \Diamond p(x),$$

  where a and p are flexible, is satisfied by the model M with domain $D = \{0,1\}, a =$ (O,l, 1.. . $\rangle$, where $p[0]$ is false at the initial state and true elsewhere, $p[1]$ is true at the initial state and false elsewhere. Take $A = \neg \Diamond p(a)$, $B = \Diamond p(x)$, $v_1 = \Diamond p(a)$, $v_2 = \Diamond p(x)$. The most-general classical unifier of v_1 and v_2 is θ = {x t a}. The resolution rule allows us to derive

  $$\neg \Diamond p(a) \; A \; \text{VX.} \; 0 \; p(x) \; A \; (\neg \mathit{true} \vee \mathit{false})$$

  which simplifies to ***false.*** This derivation would (unsoundly) show that u is not satisfiable.

- Substitution into modal contexts:

  The formula

  $$u = \neg p(a) \wedge \forall x. \left[p(x) \vee \bigcirc\left(p(x) \wedge \neg p(a)\right)\right],$$

  where a and p are flexible, is satisfied by the model $\mathcal{M}$ described in the -previous example. Take A = -p(a), $B = \left[p(x) \vee \bigcirc\left(p(x) \wedge \neg p(a)\right)\right]$, $v_1 = p(a), v_2 = p(x)$. The most-general classical unifier of v_1 and v_2 is $\theta = \{x \leftarrow a\}$. The resolution rule allows us to derive

  $$\begin{array}{l} -p(a) \wedge \forall x. \; [\mathrm{p(X)} \vee \bigcirc(p(x) \wedge \neg p(a))] \\ \qquad \wedge \\ [\neg \mathit{true} \vee [\mathit{false} \vee 0 \, (p(a) \wedge \neg p(a)) \; 11 \end{array}$$

  which simplifies to

  $$\bigcirc\left(p(a) \wedge \neg p(a)\right).$$

  Another simple application of resolution immediately yields ***false.*** Thus, we could (unsoundly) show that u is not satisfiable. We can trace back this error to applying {x t a) to $\bigcirc\left(p(x)\right.$ A $\left.\neg p(a)\right)$: while we make x = a for the current state, this does not guarantee that x = a in the next state (since a may change value).

A restriction is added to the resolution rule to deal with these difficulties:

(vi) Suppose the replaced occurrences of $v\theta$ are all in the scope of c $\bigcirc$'s and are not in the scope of any other modal operator in either $A\theta$ or $B\theta$. If $(x \leftarrow t) \in \theta$ and a flexible symbol occurs in t then all occurrences of x in A and B are in the scope of c $\bigcirc$'s and are not in the scope of any other modal operators in either A or B.

Intuitively, the new restriction guarantees that if θ indicates x should be equal to t, then θ refers to the value of t in c time units and is only applied in contexts where this would be clear (that is, in the scope of c O's).

This final restriction on the resolution rule allows us to prove:

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

6. AN EXAMPLE

Let p and q be flexible predicate symbols and let a be a flexible constant symbol. To prove that

$$[\bigcirc(p(a) \vee q(a)) \wedge \Box(\forall x.\neg p(x))] \supset \bigcirc q(a)$$

we will attempt to derive ***false*** from

$$S_0 = \neg[\neg[\bigcirc(p(a) \vee q(a)> \wedge \Box \quad \mathit{(VXJP(X))]} \vee \bigcirc \mathit{q(a)]} .$$

By simplification, we first get

$$\bigcirc(p(a) \vee \mathit{q(a))}\ \mathit{A}\ \Box \quad \mathit{(VXJP(X))} \wedge \bigcirc \neg q(a).$$

Take $A = \bigcirc \neg q(a)$, $B = \bigcirc(p(a) \vee q(a))$, $v_1 = q(a)$, $v_2 = q(a)$. Resolution yields

$$\bigcirc(p(a) \vee q(a)) \wedge \Box \quad (\forall x.\neg p(x)) \wedge 0 \neg q(a)$$
$$\mathrm{A}$$
$$[\bigcirc \neg \mathit{true} \vee \bigcirc(p(a) \vee \mathit{false})].$$

true-false simplifications yield

$$\bigcirc(p(a) \vee q(a))\ \mathrm{A}\ \Box \quad (\forall x.\neg p(x))\ \mathrm{A}\ \bigcirc \neg q(a)\ \mathrm{A}\ \bigcirc p(a).$$

Weakening reduces this sentence to

$$\Box \quad \mathit{(Vx.lp(x))} \wedge 0\ p(a).$$

An application of the □ rule yields

$$\Box \quad (\forall x.\neg p(x)) \wedge \bigcirc p(a)$$

A

$$(\forall x.\neg p(x)) \text{ A } \bigcirc \Box \quad \text{(V-p(x))}$$

and another application of the □ rule yields

$$\Box \quad \textit{(VX*‘-?P(X))} \wedge \bigcirc p(a)$$

∧

$$(\forall x.\neg p(x)) \text{ A } \bigcirc \left[\Box(\forall x.\neg p(x)) \text{ A } (\forall x.\neg p(x)) \text{ A } \bigcirc \Box(\forall x.\neg p(x))\right].$$

Weakening reduces this sentence to

$$0\ p(a) \wedge \bigcirc(\forall x.\neg p(x)).$$

The ○ ○ rule and weakening yield

$$\bigcirc[p(a) \wedge (\forall x.\neg p(x))].$$

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

$$\bigcirc\left[p(a) \text{ A } (\forall x.\neg p(x)) \text{ A } [(\forall x.\neg \mathit{true}) \vee \mathit{false}]\right].$$

true-false simplifications yield

$$\mathit{false}.$$

7. COMPLETENESS ISSUES

This section sketches some of the basic theory of **R.**

Incompleteness Theorem: The standard notion of validity in FTL, ⊨, is xi-complete.

This theorem was fist proved by Parikh ([Pa]).

It follows that no effective system for FTL can be complete for the standard models. In particular, **R** is incomplete. Therefore, it is natural to ask whether other proposed FTL systems are more or less powerful than **R.** For instance, a natural Hilbert system **T** for FTL (inspired from the one in [MP1]), adds some rules and axioms for quantifiers and a variant of the Barcan axiom to a usual PTL proof system. This defines the concept $\vdash_T$.

- If $\vdash_T u$ and $\vdash_T (u \supset v)$ then $\vdash_T v$.
- If $\vdash_T u$ then $\vdash_T \Box u$.

- If $\vdash_T(u \supset v)$ and x is not free in u then $\vdash_T(u \supset \forall x.v)$.
- If u is an instance of a schema valid in Propositional Temporal Logic then $\vdash_T u$.
- If u is an equality axiom then $\vdash_T u$.
- $\vdash_T \exists x.\neg u \equiv \neg\forall x.u$.
- $\vdash_T(\forall x.w) \supset w\theta$ where θ is {x ← ♦“ and does not create any new bound occurrences of variables or any new occurrences of flexible terms in the scope of modal operators.
- If u does not contain any flexible symbols then $\vdash_T u \equiv \Box\, u$.
- $\vdash_T(\forall x. \bigcirc \mathbf{u}) \equiv (\bigcirc \forall x.u)$.

The resolution system **R** (even without skolemization rules) is as powerful as the above Hilbert system:

Theorem: For all formulas u, $\vdash$ u $\Leftrightarrow \vdash_T u$.

Acknowledgements: We are grateful to Marianne Baudinet, Alexandre Bronstein, and . Bengt Jonsson for critical reading of the manuscript.

REFERENCES

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

[B] G. Bellin, unpublished memo, 1985.

[Ca] A. Cavalli, “A method of automatic proof for the specification and verification of protocols ,” ACM SigComm ‘84 Symposium, Communications Architectures and Protocols, 1984, pp. 100-106.

[CF] A. Cavalli and L. Fariñas del Cerro, “A decision method for linear temporal logic,” *7th International Conference on Automated Deduction (R.* E. Shostak, ed.), Springer-Verlag LNCS 1’70, 1984, pp. 113-127.

[CE] E.M. Clarke and E.A. Emerson, “Design and synthesis of synchronization skeletons using branching time temporal logic,” in *Logics of Programs* (D. Kozen, ed.), Springer-Verlag LNCS 131, 1981, pp. 52–71.

[G] M. Georgeff, “Communication and interaction in multi-agent planning,” Proc. of AAAI, Aug. 1983, pp. 125-129.

[GPSS] D. Gabbay, A. Pnueli, S. Shelah, and J. Stavi, "The temporal analysis of fairness," Seventh ACM Symposium on Principles of Programming Languages, 1980, pp. 163-173.

[HO] B. Hailpern and S. Owicki, "Modular verification of computer communication protocols," ***IEEE*** *Trans. on Communications*, Vol. COM-31, No. 1, Jan. 1983, pp. 56-68.

[L] L. Lamport, "Specifying concurrent program modules," ***ACM*** *Transactions on* ***Programming, Languages, and Systems,*** Vol. 5, No. 2, April 1983, pp. 190-222.

[M] B. Moszkowski, ***Reasoning about Digital Circuits,*** Doctoral Dissertation, Computer Science Department, Stanford University, 1983.

[MP1] Z. Manna and A. Pnueli, "Verification of concurrent programs: A temporal proof system," Report No. STAN-CS-83-967, Computer Science Department, Stanford University, June 1983.

[MP2] Z. Manna and A. Pnueli, "Adequate proof principles for invariance and liveness properties of concurrent programs," ***Science*** *of* ***Computer Programming,*** Vol. 4, No. 3, Dec. 1984, pp. 257-289.

[MWa1] 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.

[MWa2] Z. Manna and R. Waldinger, "Special relations in automated deduction," ***JACM***, Vol. 33, No. 1, Jan. 1986, pp. l-59.

[MWa3] Z. Manna and R. Waldinger, "Special relations in program-synthetic deduction," Report No. STAN-CS-82-902, Computer Science Department, Stanford University, March 1982.

[MWo] Z. Manna and P. Wolper, "Synthesis of communicating processes from temporal logic specifications," ***ACM Transactions on Programming, Languages, and Systems,*** Vol. 6, No. 1, Jan. 1984, pp. 68-93.

[OL] S. Owicki and L. Lamport, "Proving liveness properties of concurrent programs," ***ACM Transactions on Programming, Languages, and Systems,*** Vol. 4, No. 3, July 1982, pp. 455-495.

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

[Pa] R. Parikh, private communication.

[Pl] D. Plaisted, "A decision procedure for combinations of propositional temporal logic and other specialized theories ," ***Journal*** *of* ***Automated Reasoning*** (to appear).

[S] I. Sain, “Relative program verifying powers of the various temporal logics,” unpublished (submitted to ***Information and Control).***

[V] G. Venkatesh, “A decision method for temporal logic based on resolution,” in ***Foundations*** *of* ***Software Technology and Theoretical Computer*** Science, ***Fifth*** Conference (S. N. Maheshwari, ed.), Springer-Verlag LNCS 206, 1985, pp. 273-***288.***

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