June 1985 Report No. STAN-CS-85-1056

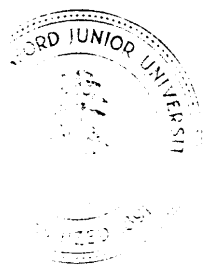
Nonclausal Temporal Deduction

by

Martin Abadi
Zohar Manna

Department of Computer Science

Stanford University
Stanford, CA 94305



NONCLAUSAL TEMPORAL DEDUCTION

Martin Abadi
Zohar Manna

Computer Science Department
Stanford University

Abstract

We present a proof system for propositional temporal logic. This system is based on nonclausal resolution; proofs are natural and generally short. Its extension to first-order temporal logic is considered.

Two variants of the system are described. The first one is for a logic with $\Box$ ("always"), $\Diamond$ ("sometime"), and $\bigcirc$ ("next"). The second variant is an extension of the first one to a logic with the additional operators $\mathcal{U}$ ("until") and $\mathcal{P}$ ("precedes"). Each of these variants is proved complete.

1. Introduction

Propositional temporal logic (PTL) is described in [MP]. The language of PTL contains the usual propositional connectives (say, A, $\vee$, $\neg$, $\supset$), and modal operators. Time is assumed discrete and linear. In PTL, if u and v range over formulas,

- $\bigcirc u$ 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"; in other words, $\Diamond u \equiv \neg \Box \neg u$;
- $u \mathcal{U} v$ 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"; in other words, u occurs at least once before v occurs, and u occurs even if v never does, i.e., $(u \mathcal{P} v) \equiv \neg((\neg u) \mathcal{U} v)$.

This research was supported in part by the National Science Foundation under grant **MCS-81-11586,** by Defense Advanced Research Projects Agency under Contract N00039-**84-C-0211,** and by the United States Air Force Office of Scientific Research under Contract **AFOSR-81-0014.**

To appear in the Proceedings of the Logics of Programs Conference **1985,** Springer-Verlag Lecture Notes in Computer Science.

Some other proof systems for PTL have been proposed in the literature. For instance, [GPSS] presents a complete Hilbert-style proof system; this system is theoretically interesting, but not very practical.

Wolper ([W]) discusses the tableau decision procedure for an extended temporal logic (ETL), a proper superset of PTL. The nondeterministic version of this decision procedure is optimal, in the sense that it achieves the worst-case lower bound for the complexity of the ETL decision problem. However, it acts uniformly on all inputs, and may do useless work on irrelevant parts of them. Moreover, neither the Hilbert-style system of [GPSS] nor the tableaux of [W] seem "liftable" to first-order temporal logic in a natural and efficient way.

Cavalli and Fariñas ([C], [CF]) extended the classical clausal resolution of [R] to PTL. Their approach is more promising than the previous ones in terms of both speed and extensibility. (We believe that their system is complete, but their completeness proof is incorrect.)

In this paper we extend nonclausal resolution ([MW], [M]) to PTL. Nonclausal resolution has the advantage over the classical clausal resolution of not requiring formulas to be in clause form, and thus making them more intelligible. For instance, we can express directly

$$\bigcirc\Box((p \supset \bigcirc q) \wedge (q \supset \bigcirc p)),$$

while in clause form, as defined in [CF], one would have to rewrite this as

Like regular nonclausal resolution proofs, our proofs are generally concise and clear, and do not require guessing clever lemmas. Moreover, some attractive refinements to resolution can easily be incorporated into our system.

In Section 2, we present a version $\mathfrak{R}$ of our system for $\bigcirc$, $\Box$ and $\Diamond$. In Section 4, we extend it to $\mathfrak{R}^+$ to deal also with $\mathcal{U}$ and ***P.*** Actually, in the extended system $\mathfrak{R}^+$, we can regard $\Box$ u and $\Diamond$ u as abbreviations for u $\mathcal{U}$ ***false*** and u ***P false,*** respectively. Both $\mathfrak{R}$ and $\mathfrak{R}^+$ are shown to be complete, in Sections 3 and 5, respectively. In Section 6 we suggest how to lift $\mathfrak{R}$ and $\mathfrak{R}^+$ to first-order temporal logic, despite the intrinsic incompleteness problems involved.

2. $\mathfrak{R}$: The resolution system for $\bigcirc$, $\Box$, $\Diamond$

We write $\vdash$ w to mean that the propositional temporal logic formula w is provable by refutation resolution, i.e., that there is a sequence of formulas So,. . . , S_n, such that $S_0 = \neg w$, S_n = ***false,*** and each S_i is obtained from previous formulas in the sequence by rules of the system. We refer to the S_i's as ***proof steps***. and to So,. . . , S_n as a ***proof.*** For each i, we call $T_i = S_0 \wedge \ldots \wedge S_i$ a ***proof truce.***

For our proof notion to be meaningful, we require that rules be sound, i.e., that they maintain satisfiability: T_{j+1} is satisfiable if and only if T_j is satisfiable, for all $j \leq (n-1)$.

While pure refutation systems are sometimes considered counterintuitive, we can easily modify our rules to show directly the validity of a sentence. The modified system we obtain is similar in spirit to the one in [MW].

We distinguish two classes of rules: simplification rules and resolution rules.

(A) Simplification rules

These rules simplify formulas, or put them in forms where the other rules can be applied. They are all of the form

$$G \Rightarrow D.$$

This can be read:

"let u_G be an instance of G, and u_D the corresponding instance of D;
if u_G occurs in the step S_i in T_j, then S_{j+1} is S_i with u_G replaced by u_D."

- ***true-false*** reduction rules:
  These rules include

$$\Box\, true \Rightarrow true \qquad \Diamond\, true \Rightarrow true \qquad \bigcirc\, true \Rightarrow true$$
$$\Box\, false \Rightarrow false \qquad \Diamond\, false \Rightarrow false \qquad \bigcirc\, false \Rightarrow false$$

  and the regular ***true-false*** reduction rules in propositional logic, such as ***false A u ⇒ false,*** etc..

- Weakening rules:

$$u \wedge v \Rightarrow u, \qquad u \wedge v \Rightarrow v.$$

  These rules are restricted to positive occurrences of instances of u A v (i.e., to occurrences embedded in an even number, explicit or implicit, of ¬'s). This polarity restriction guarantees the soundness of the rules.

- Distribution rules (¬ over modalities):

$$\neg \Box u \Rightarrow \Diamond \neg u, \qquad \neg \Diamond u \Rightarrow \Box \neg u, \qquad \neg \bigcirc u \Rightarrow \bigcirc \neg u.$$

- Distribution rules (¬ over connectives):

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

  Similar rules are added for the other connectives.

(B) Resolution rules

We will have **a** resolution operator R to obtain new formulas from given ones. Our resolution rules will be of the two following forms:

- unary:

$$R[G] \longmapsto D \ .$$

This can be read:

"let u_G be an instance of G, and u_D the corresponding instance of ***D;***
if u_G occurs in the step S_i in T_j, then S_{j+1} is S_i with u_G replaced by u_G A u_D
(in the special case where $u_G = S_i$, it suffices to take $S_{j+1} = u_D$)."

- binary:

$$R[G_1, G_2] \longmapsto D.$$

This can be read:

"let u_{G_1} and u_{G_2} be instances of G_1 and G_2, and u_D the corresponding instance of ***D;***
if u_1 A . . . A u_k occurs in the step S_i in T_j, and for some h and h' (possibly $h = h$), $u_{G_1} = u_h$ and $u_{G_2} = u_{h'}$, then S_{j+1} is S_i with u_1 A.. . A u_k replaced by u_1 A . . . $\wedge\, u_k \wedge u_D$;
if for some h and h' (possibly $h = h$), $u_{G_1} = S_h$ and $u_{G_2} = S_{h'}$, then we take $S_{j+1} = u_D$."

We will say that the given formulas u_G or u_{G_1}, u_{G_2} are being ***resolved upon,*** and that the derived sentence u_D is one of their ***resolvents.***

Polarity restriction: The rule is applied only to u_G, u_{G_1}, u_{G_2} which occur with positive polarity (i.e., the given formulas are embedded in an even number, explicit or implicit, of ¬'s). Each of our rules has the property that u_D is **a** logical consequence of u_G or $u_{G_1} \wedge u_{G_2}$. Therefore, with the polarity restriction, the soundness and usefulness of our rules is guaranteed.

Here are the resolution rules:

The basic rule

The basic nonclausal resolution rule for propositional logic is:

$$R[A\langle u\rangle,\ B\langle u\rangle] \longmapsto A(true) \vee B\langle false\rangle.$$

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

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

The basic rule is sound in PTL under the following restrictions:

The occurrences of u in A or B that are substituted by *true* or false, respectively, are all in the scope of the same number of $\bigcirc$'s, and not in the scope of any $\Box$ or $\Diamond$ in A or B. Intuitively, this means that all the occurrences of u refer to the same instant of time.

For example, consider the formulas

$$A: \bigcirc \neg \bigcirc(\Box p \vee q) \wedge \Diamond \Box p \text{ and } B: \bigcirc \bigcirc \Box p \vee \bigcirc \Box p.$$

Taking u to be $\Box$ p, the rule **allows us to derive the resolvent**

$$(\bigcirc \neg \bigcirc(true \vee q) \wedge 0 \quad \Box p) \vee (\bigcirc \bigcirc false \vee \quad \bigcirc \Box p).$$

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

This rule is quite general, but does not handle U's in the scope of $\Box$'s and $\Diamond$'s, and is certainly not complete. To complement this rule, we develop the following additional rules.

Modality rules

These are rules to handle formulas in the scope of $\Box$, $\Diamond$ and $\bigcirc$. They allow us to resolve upon formulas which are otherwise not accessible, because they are in the scope of some modal operator.

- $\Box$ rule:

$$R[\Box u] \longmapsto u \quad A \quad \bigcirc \Box u.$$

- $\Diamond$ rule:

$$R[\Diamond u] \longmapsto u \quad v \quad \bigcirc \Diamond u.$$

- $\Box\Box$ rule:

$$R[\Box u, \Box v] \longmapsto \Box(\Box u \wedge v).$$

After applying this rule, we we will often attempt to resolve $\Box$ u and v. Thus, we could informally write

$$R[\Box u, \Box v] \longmapsto \Box R[\Box u, v].$$

(Other modality rules can be rephrased similarly, to reflect their intended use.)

- $\Box \Diamond$ rule:

$$R[\Box u, \Diamond v] \longmapsto \Diamond(\Box u \wedge v).$$

- $\Diamond \Diamond$ rule:

$$R[\Diamond u, \Diamond v] \longmapsto \Diamond(\Diamond u \wedge v) \vee \Diamond(u \wedge \Diamond v).$$

- $\bigcirc \bigcirc$ rule:

$$R[\bigcirc u, \bigcirc v] \longmapsto \bigcirc(u \wedge v).$$

Two useful derived rules are:

- $\Box \bigcirc$ derived rule (obtained from the $\Box$ and $\bigcirc \bigcirc$ rules, with weakening):

$$R[\Box u, \bigcirc v] \longmapsto \bigcirc(\Box u \wedge v).$$

- $\Diamond \bigcirc$ derived rule (obtained from the $\Diamond$ and $\bigcirc \bigcirc$ rules):

$$R[\Diamond u, \bigcirc v] \longmapsto u \vee \bigcirc(\Diamond u \wedge v).$$

Our powerful induction principle (presented below) makes some of these rules (in fact, all but $\Box$, $\Diamond$, and $\bigcirc \bigcirc$) unnecessary for completeness. We include them because they often provide convenient and natural short-cuts in proofs.

The induction rule

The induction rule (μ) is:

if $\vdash \neg(w \wedge u)$, then

$$R[w, \Diamond u] \longmapsto \Diamond(\neg u \wedge \bigcirc(u \wedge \neg w)).$$

A special case of this rule (when w = $\neg u$) is

$$R[\neg u, \Diamond u] \longmapsto \Diamond(\neg u \wedge \bigcirc u).$$

This special case of μ evokes a form of the least number principle: if $\neg\Phi(0)$ and, for some n, $\Phi(n)$, then, for some m, $d(m) \wedge \Phi(m+1)$. However, our rule is more powerful in that it does not require $\neg u$, but simply $\vdash \neg(w \wedge u)$, for some w.

The extra flexibility **of** having an arbitrary w as hypothesis to μ will be essential for some proofs. The special case of μ will yield those proofs only if we add a special rule to introduce lemmas (intuitively, these lemmas would be useful in guessing inductive sentences). Such a system would depend on clever heuristics to discover good lemmas; of course, this feature is undesirable.

To show that the special case of μ described above is too rudimentary, consider the unsatisfiable sentence S:

$$p \wedge \Box(p \supset \bigcirc p) \wedge (\Diamond \Diamond \neg p).$$

One would like to be able to refute this sentence by induction. The special induction principle must be applied to conjuncts of form $\neg u$ and $\Diamond$ u, and, therefore, we must take u to be $\Diamond \neg p$ to resolve upon $\Diamond \Diamond \neg p$. However, since $\neg \Diamond \neg p$ is not one of the conjuncts of S, the special rule cannot be applied. On the other hand, taking w to be $p \wedge \Box (p \supset \bigcirc p)$, the general rule μ can be applied; it requires that

$$\vdash \neg(p \wedge \Box(p \supset \bigcirc p) \wedge \Diamond \neg p),$$

which can easily be proved. Thus we can deduce

$$\Diamond(\neg 0 \neg p \wedge \bigcirc(\Diamond \neg p \wedge \neg(p \wedge \Box (p \supset \bigcirc p))))$$

and this leads to a refutation of S in just a few trivial steps.

Distribution rules

Consider the following example. We know that

$$\Box(\neg p \wedge \neg q) \wedge (\Diamond p \vee \Diamond q)$$

is unsatisfiable; we should therefore be able to derive *false* from it. A natural way to do this is to resolve $\Box (\neg p \wedge \neg q)$ with $\Diamond p$ to obtain false (by $\Box \Diamond$ and the basic rule), and $\Box(\neg p \wedge \neg q)$ with $\Diamond q$ to obtain *false* (by the same rules), and to form the disjunction of these results. We get false v false, and, then, false, concluding the refutation. A key step in this proof was our ability to resolve over v **in** $\Diamond p \vee \Diamond q$.

This motivates the following V and A rules:

$$R[u, v_1 \vee \ldots \vee v_k] \longmapsto (u \wedge v_1) \vee \ldots \vee (u \wedge v_k),$$

$$R[u, v_1 \wedge \ldots \wedge v_k] \longmapsto (u \wedge v_1 \ldots \wedge v_k).$$

One can write similar rules for the other propositional connectives, e.g., $\supset$ and ***if-then-else.***

An example

Let us prove the validity of the sentence

$$[p \wedge \Box(p \supset_0 \Diamond p)] \supset \Box \Diamond p.$$

In other words, we will refute the sentence

$$[p \wedge \Box(p \supset \bigcirc \Diamond p)] \wedge \neg \Box \Diamond p$$

The refutation is:

1) $[p \wedge \Box_{NP} \supset \bigcirc \Diamond p)] \wedge \neg \Box \Diamond p$	initial assertion
2) p	by weakening (line 1)
3) $\Box_{(p \supset_0 0\ P)}$	by weakening (line 1)
4) $\Diamond \neg \Diamond p$	by weakening (line 1), and distributing $\neg$ over $\Diamond$
5) $\Diamond(\neg\neg \Diamond p \wedge \bigcirc(\neg \Diamond p \wedge \neg p))$	by μ (from lines *3* and *4)* since $\vdash \neg(p \wedge \neg \Diamond p)$
6) $\Diamond(\Diamond p \wedge \bigcirc \neg \Diamond p)$	by simplifying $\neg\neg \Diamond p$ and weakening
7) $\Diamond((p \vee \bigcirc \Diamond p) \wedge \bigcirc \neg \Diamond p)$	by $_0$
8) $\Diamond((p \wedge \bigcirc \neg \Diamond p) \vee (\bigcirc \Diamond p \wedge \bigcirc \neg \Diamond p))$	by $\vee$ distribution
9) $\Diamond((p \wedge \bigcirc \neg \Diamond P) \vee (\bigcirc false \vee \bigcirc \neg true))$	by the basic rule (on $\Diamond p$) and weakening
10) $\Diamond(p \wedge_0 \neg \Diamond P)$	by simplification
11) $\Diamond((p \wedge \bigcirc \neg \Diamond p) \wedge \Box(p \supset \bigcirc \Diamond p))$	by $\Box \Diamond$ (with line *3)*
12) $_0((p \wedge_0 \neg_0 p) \wedge (p \supset_0 0\ p))$	by $\Box$ and weakening
13) $\Diamond((p \wedge \bigcirc \neg \Diamond p) \wedge ((false) \neg \Diamond p) \vee (true \supset \bigcirc \Diamond P)))$	by the basic rule (on p) and weakening
14) $_0((p \wedge_0 \neg_0 P) \wedge_0 0\ P)$	by simplification
15) $\Diamond((p \wedge \bigcirc \neg true) \wedge \bigcirc false)$	by the basic rule (on $\Diamond p$) and weakening
16) $\Diamond\ false$	by simplification
17) false	by *false* reduction

To justify line 5, we still need to show a refutation for $\neg\neg(p \wedge \neg \Diamond p)$.

1′) $\neg\neg(p \wedge \neg_0 p)$	initial assertion
2′) $p \wedge \Box \neg p$	by simplification
3′) $p \wedge \neg p$	by $\Box$ and weakening
4) false	by the basic rule and weakening

3. Soundness and completeness for $\mathfrak{R}$

We have

Soundness theorem. $\mathfrak{R}$ is sound.

and

Completeness theorem. $\mathfrak{R}$ is complete.

The proof of the soundness theorem is trivial. We give an outline of the completeness proof.

Proof outline

The tableau decision procedure of [W] is known to be complete. We show that if this decision procedure finds $\neg u$ unsatisfiable, then $\vdash$ u. The resolution refutation for $\neg u$ may actually be quite similar to the one found in the tableau.

The tableau decision procedure creates **a** finite graph with formulas at the nodes. The initial node contains $\neg u$; each node contains formulas derived from those of its parent. Intuitively, children either

- expand what their parent says about the present (for instance, if N contains $\Box$ u then it may have a child with $\Box$ u, $\bigcirc\Box$ u and u); or
- summarize what their parent says about tomorrow, by eliminating all formulas not of the form $\bigcirc$ v and erasing $\bigcirc$'s from all others; any node obtained in this way and the initial node are called ***pre-states.***

A node is eliminated in the following three cases:

1) ***Clash:*** it contains **a** proposition and its negation;

2) ***Propagation:*** all of its descendants have been eliminated;

9) Eventualities: if the node is a pre-state and contains $\Diamond v_1, \ldots, \Diamond v_k$, and on no path from the node do all the vi's occur.

$\neg u$ is found unsatisfiable if and only if all nodes of the tableau have been eliminated.

Our proof has three main parts; each of them corresponds to one of the ways to eliminate unsatisfiable nodes of the tableau. In each of them, we show that if a node is eliminated, then the formulas it contains can be refuted by $\mathfrak{R}$. They are proved together

by induction on the tableau. In this induction, the rank of a node N is smaller than that of another node M if N is not an ancestor of M, and M is an ancestor of N. In particular, all nodes in a cycle have the same rank.

In this proof we will often (informally) identify a node with the set of formulas it contains. Sometimes this set will be identified with the conjunction of the formulas it contains; of course, this is merely for convenience, and involves no special assumptions.

1) Clash: The first case deals with nodes eliminated because they contain some proposition and its negation. Clearly, the formulas in such nodes could just as well be refuted in $\mathfrak{R}$, with the basic rule and the reduction rules for ***true*** and false.

2) Propagation: The second case deals with nodes whose children have all been eliminated. We can assume (by inductive hypothesis) that all those eliminated children could also be refuted in $\mathfrak{R}$, and show that the parent node can be refuted in $\mathfrak{R}$ as well. The proof of this inductive step involves a case analysis, where we consider which tableau rule the children where created by. Two examples of such cases are:

- The parent node contains $u_1, \ldots, u_n, \Box$ v, and the child is created by the rule that expands $\Box$'s, i.e., the child contains $u_1, \ldots, u_n, \Box v, v, \bigcirc \Box$ v. Assume there is a refutation of $u_1, \ldots, u_n, \Box$ v, v, $\bigcirc \Box$ v, to show that there is a refutation of $u_1, \ldots, u_n, \Box$ v. This is trivial, by the $\Box$ rule.

- The parent node contains $v_1, \ldots, v_n, \bigcirc u_1, \ldots, \bigcirc u_k$, where the v_i's are not of the form $\bigcirc$ w, and the child is created by the rule that erases $\bigcirc$'s, i.e., the child contains $u_1, \ldots, u_k$. Assume there is a refutation of $u_1, \ldots, u_k$, to show that there is a refutation of vi,. . . , $v_n, \bigcirc u_1, \ldots, \bigcirc u_k$. The $\bigcirc\bigcirc$ rule will give us a refutation for $\bigcirc u_1, \ldots, \bigcirc u_k$, which can be extended to a refutation for $v_1, \ldots, v_n, \bigcirc u_1, \ldots, \bigcirc u_k$.

3) Eventualities: Finally, the tableau method eliminates pre-states w which contain some unfulfillable eventualities (in other words, w = v A $\Diamond u_1$ A . . . A $\Diamond u_k$ and on no path from w do all the u_i's occur). We prove than w can be refuted.

- Since the tableau is finite, some paths from w must cycle back to w, and others may have been stopped further down in the tableau; we will only need to consider those that cycle back to w, and show that all nodes on these cycles can be eliminated (the other paths are dealt with by inductive hypothesis).

- Our goal will be to exploit the finite model property, and derive a formula Φ which expresses that at some point we are in one of the pre-states of the cycles, and at the next moment we are no longer in one. Furthermore, it will be easy to show that Φ can be refuted in $\mathfrak{R}$. The following five lemmas implement this basic idea.

- We will want to construct future pre-states from a given pre-state p_0. More precisely, we get w_i's that describe what the world can be like after i steps: w_i says that one of

the pre-states at depth i will be true then. We will call w_i the i^{th} ***fringe*** of p_0. For instance, if $p_0 = (\Box q \vee \Box r) \wedge \Diamond r$ then $w_0 = p_0$, and

$$w_1 = \bigcirc(\Box q \wedge \Diamond r) \vee \bigcirc \Box q \vee \bigcirc(\Box r \wedge \Diamond r) \vee \bigcirc \Diamond r.$$

While this lemma "normalizes" formulas (roughly, into disjunctions of conjunctions), this does not necessarily correspond to how most proofs work-generally, formulas seem to stay close to their original form.

Lemma 1:
Given a pre-state p_0, $\Re$ can derive formulas w_i built up from the pre-states p_i^l at depth i from p_0 with $\vee$ and $\bigcirc$. Furthermore, each p_i^l occurs in the scope of exactly i $\bigcirc$'s.

Proof:
Let $w_0 = p_0$. For all i, it is easy to obtain w_{i+1} from w_i:
Use the $\Box$ and $\Diamond$ rules to expand p_0, just like in the corresponding tableau. Use the rules for the connectives to push $\neg$'s inwards and to obtain a disjunction of conjunctions. Apply the $\bigcirc\bigcirc$ rule to pull $\bigcirc$'s out of conjunctions. Apply the basic rule whenever both a proposition and its negation appear in a conjunction at depth i. Then, by weakening, throw away all conjuncts at depth i; we are left with those at depth i + **1.**

This lemma does not depend on any inductive hypothesis, and applies to any p_0.

- Lemma **2** uses the inductive hypothesis to eliminate some unsatisfiable pre-states from the w_i's of Lemma 1. In particular, we can eliminate pre-states outside the cycles for

Lemma 2:
If in Lemma 1 p_0 = w then the w_i's can be built in such a way that they only include pre-states from the cycles back to w.

Proof:
As soon as we get a pre-state p outside the cycles for w in the construction of Lemma 1, we apply the inductive hypothesis to refute it.

- The following lemma about the tableau decision procedure shows that we only need to worry about one eventuality at a time.

Lemma 3: If w contains the unfulfillable set of eventualities $\{\Diamond u_1, \ldots, \Diamond u_n\}$, then some u_i does not occur on any of the loops back to w.

Proof:
Suppose, on the contrary, that each eventuality is fulfilled on at least one of the loops back to w. Then we can find a path where they are all fulfilled: go through the loop where u_1 occurs, then through the loop where u_2 occurs, . . ., then through the loop where u_n occurs. Since this contradicts our hypothesis, there must be one eventuality that is not fulfilled on any of the loops back to w.

Thus, from now on, we will have w = v A ◇ u, where u does not occur on any of the loops back to w.

- The unfulfillable eventuality ◇ u is blocked at every time in $\mathfrak{R}$.

Lemma 4:
For any i, $\vdash \neg(w_i \text{ A } \bigcirc^i u)$.

Proof:
Distribute $\bigcirc^i u$ over the pre-states in w_i and eliminate some occurrences of ◇ u (by weakening), in order to derive w_i' built up from pre-states of the tableau for w with V and ○ (and, as usual, all pre-states at the same depth i). Let p be one of these pre-states. p cannot be on the cycles back to w, since it contains u. Hence, we can refute p in $\mathfrak{R}$ (by inductive hypothesis). Therefore, w_i' and w_i A $\bigcirc^i u$ can also be refuted.

- Lemma **5** is the main lemma of the completeness proof.

Lemma 5:
If for all $i \vdash \neg(w_i \text{ A } \bigcirc^i u)$ then $\vdash \neg w$.

Proof:
W = w_0. From w_0 and ◇ u, μ and weakening yield

$$\Diamond \bigcirc(u \wedge \neg w_0)$$

since $\vdash \neg(w_0$ A u) by hypothesis.
By Lemma 2, we can derive w **1**. Furthermore, $\vdash \neg(w_1$ A ○ u) by hypothesis. Thus, μ yields

$$\Diamond \bigcirc(\bigcirc(u \wedge \neg w_0)) \wedge \neg w_1).$$

In general, we can get all w_i's, and check that $\vdash \neg(w_i \text{ A } \bigcirc^i u)$, by hypothesis. Successive applications of μ will give

$$\Diamond \bigcirc(\bigcirc(\bigcirc \ldots \bigcirc(u \text{ A } \neg w_0) \text{ A} \ldots \text{ A } \neg w_{t-1}) \text{ A } \neg w_t)$$

for any t. We weaken this to

$$\Diamond \bigcirc(\bigcirc(\bigcirc \ldots \bigcirc(\neg w_0) \text{ A} \ldots \text{ A } \neg w_{t-1}) \text{ A } \neg w_t).$$

Call this formula Ψ_t, and define also Ω_t by $\Psi_t = \Diamond \Omega_t$. Ψ_t says that at some point we will not be in any of the pre-states in the first t + **1** fringes of w_0.

The finite model property tells us that there are only finitely many fringes of w_0, up to collapsing. Thus, for some s, Ψ_s says that we are in no pre-state reachable from w_0. (As usual, Lemma **2** tells us that we can limit ourselves to pre-states in the cycles for w.)

$\vdash \neg(w_0 \wedge \Omega_s)$, simply by writing the s^{th} fringe of w_0, and observing that all its pre-states are denied in Ω_s. Thus, we can apply μ (and weakening) once more and get

$$\Diamond(\neg\Omega_s \wedge \bigcirc \Omega_s).$$

Call this formula Φ. Φ says that at some point we are in one of the pre-states reachable from w_0, and that at the next instant we are in none of them. Of course, this cannot be the case; in fact, we can refute $\neg\Omega_s \wedge \bigcirc \Omega_s$: we derive the first fringe of all pre-states in $\neg\Omega_s$, and check that all the pre-states in these fringes were already in $\neg\Omega_s$. Thus, we derived $\Diamond$ false, and hence false.

Remarks: Note that w contains $\Diamond$ u. Let $r = \neg u$. Thus, Lemma 5 connects r being proved at all instants with a proof for $\Box r$. This observation leads us to another formulation of Lemma 5, which makes clear that we could have reduced infinitary systems to $\Re$ (instead of tableaux). Such a reduction is successful only because of the finite model property.

The ω Lemma: If l-v $\supset \bigcirc^i r$ for all i then $\vdash$ v $\supset \bullet] r$.

We will not explore this relation with infinitary logic any further.

- From Lemma 4 and Lemma 5 we obtain that w can be refuted in $\Re$.

4. $\Re^+$: The resolution system for $\bigcirc$, $\mathcal{U}$, $\mathcal{P}$ (and $\Box$, $\Diamond$)

The resolution system $\Re^+$ for $\bigcirc$, $\mathcal{U}$, and $\mathcal{P}$ is a generalization of $\Re$. In fact, all rules of $\Re$ are natural special cases of rules of $\Re^+$, when we regard $\Box$ u and $\Diamond$ u as abbreviations for $u\ \mathcal{U}$ *false* and u $\mathcal{P}$ false, respectively. One important qualification is that $\mathcal{P}$ reverses the polarity of its second argument. Therefore, it may be very convenient to include rules to act on formulas of negative polarity, like the "goal-goal" rules of [MW]. These rules are dual to those for formulas of positive polarity. Since this extension is not necessary, we will not discuss it here.

(A) Simplification rules

- ***true-false*** reduction rules:

$false\ \mathcal{U}\ v \Rightarrow v$	$false\ P\ v \Rightarrow false$	$\bigcirc\ false \Rightarrow false$
$true\ \mathcal{U}\ v \Rightarrow true$	$true\ P\ v \Rightarrow \neg v$	$\bigcirc\ true \Rightarrow true$
$u\ \mathcal{U}\ true \Rightarrow true$	$u\ P\ true \Rightarrow false$	

  and the regular ***true-false*** reduction rules in propositional logic.

- Weakening rules: same as for $\Re$.
- Distribution rules ($\neg$ over modalities):

$$\neg(u\ \mathcal{U}\ v) \Rightarrow (\neg u)\ P\ v, \qquad \neg(u\ P\ v) \Rightarrow (\neg u)\ \mathcal{U}\ v, \qquad \neg \bigcirc u \Rightarrow \bigcirc \neg u.$$

- Distribution rules ($\neg$ over connectives): same as for $\Re$.

(B) Resolution rules

The basic rule

The basic rule for $\mathfrak{R}^+$ is similar to that for $\mathfrak{R}$. Note, however, that the restriction is now:

> the occurrences of u in A or B that are substituted by *true* or false, respectively, are all in the scope of the same number of $\bigcirc$'s, and not in the scope of any $\mathcal{U}$ or $\mathcal{P}$ of A or B.

Modality rules

These are the rules that relate $\bigcirc$, $\mathcal{U}$ and $\mathcal{P}$ by allowing us to resolve formulas in their scope.

- $\mathcal{U}$ rule:

$$R[u\ \mathcal{U}\ u] \longmapsto u\ v\ (u\ A\ \bigcirc(u\ \mathcal{U}\ u)).$$

- $\mathcal{P}$ rule:

$$R[u\ P\ u] \longmapsto \neg v\ A\ (u\ v\ \bigcirc(u\ P\ v)).$$

- $\mathcal{U}\mathcal{U}$ rule:

$$R[u\ \mathcal{U}\ v, u'\ \mathcal{U}\ u'] \longmapsto u\ P\ v'\ v\ (\neg v\ A\ (u\ \mathcal{U}\ u)\ A\ u')\ \mathcal{U}\ u'.$$

That is, unless u P u', u does not occur before u', and hence u $\mathcal{U}$ u is true until u', and so are $\neg v$, and, of course, u'.

- UP rule:

$$R[u\ \mathcal{U}\ v, u'\ P\ v'] \longmapsto u\ P\ v'\ v\ (\neg v\ A\ (u\ \mathcal{U}\ u)\ A\ u')\ P\ v'.$$

That is, unless v $\mathcal{P}$ u', u does not occur before u', and hence $\neg v$ and u $\mathcal{U}$ v are true at the point where u' is true before v'.

- PP rule:

$$R[u\ P\ v, u'\ P\ u'] \longmapsto ((u\ A\ u'\ P\ v')\ P\ v)\ v\ ((u\ P\ v\ A\ u')\ P\ v').$$

That is, if u occurs no later than u', then at some point before v we have u and $u'\ \mathcal{P}\ v'$; on the other hand, if u' occurs no later than u, then at some point before v' we have u' and u P v.

- $\bigcirc\bigcirc$ rule: same as for $\mathfrak{R}$.

We also have two useful derived rules:

- $\mathcal{U}\bigcirc$ derived rule (obtained from the $\mathcal{U}$ and $\bigcirc\bigcirc$ rules, with weakening):

$$R[u\ \mathcal{U}\ u,\ \bigcirc w] \longmapsto u\ v\ \bigcirc(u\ \mathcal{U}\ v\ A\ w).$$

- $P\bigcirc$ derived rule (obtained from the P and $\bigcirc\bigcirc$ rules):

$$R[u\ P\ v,\ \bigcirc w] \longmapsto u\ V\ \bigcirc(u\ P\ v\ A\ w).$$

Like in $\mathfrak{R}$, some of these rules are not essential for completeness, and we present them simply because of their usefulness. In fact, of all these modality rules, only $\mathcal{U}$, P, and $\bigcirc\bigcirc$ are indispensable.

The induction rule

The induction rule of $\mathfrak{R}$ was:

if $\vdash \neg(w$ A u), then

$$R[w,\ \Diamond u] \longmapsto \Diamond(\neg u \wedge \bigcirc(u \wedge \neg w)).$$

The μ rule carries over to $\mathfrak{R}^+$, with only minor changes; in $\mathfrak{R}^+$, it has the form:

if $\vdash$ (w A u) $\supset$ v, then

$$R[w,\ u\ P\ u] \longmapsto (\neg u\ A\ \bigcirc(u\ A\ \neg w))\ P\ (v\ V\ \bigcirc v).$$

(Notice that in the special case v = *false* we obtain $\mathfrak{R}$'s μ rule.)

The μ rule in $\mathfrak{R}$ stated that, under the assumption that u and w cannot be true simultaneously, if w is true now, and u is true at some future instant, then eventually u must change from false to true. In other words, at eventually $\neg u\ A\ \bigcirc(u\ A\ \neg w)$. In $\mathfrak{R}^+$ this is refined to take into account that u must be true before v, and hence $\neg u\ A\ \bigcirc(u\ A\ \neg w)$ must be true before v and $\bigcirc v$.

Distribution rules

The distribution rules are exactly those for $\mathfrak{R}$.

5. Soundness and completeness for $\mathfrak{R}^+$

We have

Soundness theorem. $\mathfrak{R}^+$ *is* sound.

and

Completeness theorem. $\mathfrak{R}^+$ is complete.

The proof of the soundness theorem is trivial. We sketch the proof of the completeness theorem,

Proof sketch

The proof of completeness for $\mathfrak{R}^+$ is a relatively straighforward generalization of that for $\mathfrak{R}$. In particular, $\mathfrak{R}^+$, like $\mathfrak{R}$, is closely related to the tableau decision procedure for the corresponding version of PTL. We prove that any formula that can be shown unsatisfiable by the tableau decision procedure can be refuted by our $\mathfrak{R}^+$ system. The structure of the proof is parallel to that of the completeness proof for $\mathfrak{R}$. However, since there are new ways to create nodes in the tableau, we need to work out some more cases of "propagation." Also, the proof that cycles with unfulfillable eventualities are eliminated is slightly more complex than for $\mathfrak{R}$.

6. Concluding remarks: first-order temporal deduction

We have presented a nonclausal resolution approach to theorem proving in PTL, with modal operators $\bigcirc$, $\Box$, $\Diamond$, and also with the additional modal operators $\mathcal{U}$ and $\mathcal{P}$. Both versions were shown complete. We expect to be able to generalize this approach to get a viable proof system for first-order temporal logic. In particular, we attempt to combine the classical "cut" and "substitution" rules, which are usually expensive in their use of heuristics, into resolution rules with unification.

In the proposed first-order system, the unification would be deferred to the end of the refutation. In this way, we will not need to find unifiers, but just to check that there exist appropriate unifiers for the refutation. Huet ([H72, 75]) discusses some of the benefits of such a "constrained resolution" approach. Another benefit of this approach in temporal logic is that we avoid unsound substitutions into the modal formulas under consideration.

While this lifting is rather natural, it has one major problem: the system we obtain is not complete. A rule to introduce lemmas may prove helpful in enlarging the set of provable sentences. At any rate, there is no hope of constructing a complete system, since arithmetic can be embedded in first-order temporal logic. This makes first-order temporal logic totally undecidable. We expect, however, that most practically useful theorems of first-order temporal logic will have short and elegant proofs in our nonclausal resolution system.

Acknowledgements We are grateful to Gianluigi Bellin, Yoni Malachi, Eric Muller, and Pierre Wolper, for their careful reading of the manuscript and many interesting discussions.

References

[C]
Cavalli, A., "A method of automatic proof for the specification and verification of protocols," ***ACM SIGCOMM 84 Symposium,*** Montreal, Canada, **1984.**

[CF]
Cavalli, A., and L. Fariñas del Cerro, "A decision method for linear temporal logic," ***Seventh Conference on Automated Deduction,*** Napa, CA, May **1984,** pp. **113-127.**

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

[H72]
Huet, G. P., "Constrained resolution: a complete method for higher order logic," Ph.D. Thesis, Case Western University, Jennings Computing Center Report **1117,** August 1972.

[H75]
Huet, G. P., "A unification algorithm for typed X-calculus," ***Theoretical Computer Science,*** Vol. **1,** No. **1,** pp. **27-57.**

[MP]
Manna, Z. and A. Pnueli, "Verification of concurrent programs: The temporal framework," in ***The Correctness Problem in Computer Science*** (R.S. Boyer and J S. Moore, eds.), International Lecture Series in Computer Science, Academic Press, London, **1982,** pp. **215-273.**

[MW]
Manna, Z. and R. Waldinger, "A deductive approach to program synthesis," ***ACM Transactions on Programming Languages and Systems,*** Vol. **2,** No. **1,** January **1980,** pp. **92-121.**

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

[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]
Wolper, P., "Temporal logic can be more expressive," ***Proc. 22nd IEEE Symp. on Foundations of Computer*** **Science,** Nashville, **1981,** pp. **340-348.**