Junc 1983 Rcport No. STAN-G-83-967

Verification of Concurrent Programs: A Temporal Proof System

by

Zohar Manna and Amir Pnueli

Department of Computer Science

Stanford Uuivcrsity
Stanford, CA 94305

VERIFICATION OF CONCURRENT PROGRAMS: A TEMPORAL PROOF SYSTEM

by

ZOHAR MANNA
Computer Science Department
Stanford University
Stanford, CA
and
Applied Mathematics Department
The Weizmann Institute of Science
Rehovot, Israel

AMIR PNUELI
Applied Mathematics Department
The Weizmann Institute of science
Rehovot, Israel

ABSTRACT

A proof system based on temporal logic is presented for proving properties of concurrent programs based on the shared-variables computation model. The system consists of three parts: the general uninterpreted part, the domain dependent part and the program dependent part. In the general part we give a complete proof system for first-order temporal logic with detailed proofs of useful theorems. This logic enables reasoning about general time sequences. The domain dependent part characterizes the special properties of the domain over which the program operates. The program dependent part introduces program axioms which restrict the time sequences considered to be execution sequences of a given program.

The utility of the full system is demonstrated by proving invariance, liveness and precedence properties of several concurrent programs. Derived proof principles for these classes of properties, are obtained and lead to a compact representation of proofs.

This paper appears in the Proceedings of the 4-th School of Advanced Programming, Amsterdam, Holland (June 1982).

This research was supported in part by the National Science Foundation under grants MCS79-09495 and MCS80-06930, by DARPA under Contract N00039-82-C-0250, by the United States Air Force Office of Scientific Research under Grant AFOSR-81-0014, and by the Basic Research Foundation of the Israeli Academy of Sciences.

A. INTRODUCTION

In this work wc present a proof system based on temporal logic for proving the properties of concurrent programs. We refer thc reader to [MP1] f'or a more detailed discussion of the computational model of concurrent programs, and thc advantages offered by the language of temporal logic in formulating properties of concurrent programs.

1. THE TEMPORAL LANGUAGE: SYNTAX AND SEMANTICS

We first describe thc temporal language wc are going to use. This language contains special constructs that are suitable for reasoning about programs.

The language uses a set of basic symbols consisting of individual variables and constants, propositions, and function and predicate symbols. The set is partitioncd into two subsets: global and local symbols. Intuitively speaking, the global *symbols* dcnotc entities that do not change during a program exccution. The *local symbols,* on thc other hand, may change their meanings and values in diffcrcnt states throughout the exccution. For our purpose, the only local symbols that interest us arc local individual variables and propositions. Wc will have global symbols of all types.

Wc usc the usual set of boolean connectives: A, **V**, $\supset$, $\equiv$, and $\sim$ together with the equality predicate = and the first-order quantifiers $\forall$ and 3. These operators arc referred to as the *classical* operators. The quantifiers $\forall$ and 3 arc applied only to global individual variables.

Thc *modal operators* used arc: $\Box$, 0, 0, and $\mathcal{U}$, which arc called respectively the *always, sometime, next* and *until* operators. The first three operators arc unary while the $\mathcal{U}$ operator is binary. We use the *next* operator, 0, in two different ways -- as a temporal operator applied to formulas and as a temporal operator applied to terms.

A model (1, α, σ) for our language consists of a (global) interpretation I, a (global) assignment α and a sequence of states σ.

- The *interpretation I* specifies a noncmpty domain D and assigns concrete elements, functions and predicates to the (global) individual constants, function and predicate symbols.

- *The assignment* α assigns a value over the appropriate domain to cach of the global individual variables.

- *The sequence* $\sigma = s_0, s_1, \ldots$ is an infinitc sequence of states. Each *state* s_i assigns values to the local individual variables and propositions.

For a sequence

$$\sigma = s_0, s_1, \ldots$$

wc denote by

$$\sigma^{(i)} = s_i, s_{i+1}, \ldots$$

the i-truncated suffix of σ.

Given a temporal formula w, wc present below an inductive definition of the truth value of w in a model $(I, \alpha,$ a). The value of a subformula or term τ under (I, α, a) is denoted by $\tau|^{\alpha}_{\sigma}$, with I being implicitly understood.

Consider first the evaluation of tcrrns:

- For a local individual variable or local proposition y:
  $y|^{\alpha}_{\sigma} = s_0[y]$,
  i.e., the value assigned to y in s_0, the first state of σ.

- For a global individual variable u:
  $u|^{\alpha}_{\sigma} = \alpha[u]$,
  i.e., the value assigned to u by α.

- For an individual constant the evaluation is given by I:
  $c|^{\alpha}_{\sigma} = I[c]$.

- For a k-ary function f:
  $f(t_1, \ldots, t_k)|^{\alpha}_{\sigma} = I[f](t_1|^{\alpha}_{\sigma}, \ldots, t_k|^{\alpha}_{\sigma})$,
  i.e., the value is given by the application of the interpreted function $I[f]$ to the values of $t_1, \ldots, t_k$ evaluated in the model (I,α,σ).

- For a term t:
  $(\bigcirc\, t)|^{\alpha}_{\sigma} = t|^{\alpha}_{\sigma^{(1)}}$,
  i.e., the value of $0\ t$ in $\sigma = s_0, s_1, \ldots$ is given by the value of t in the 1-truncated suffix $\sigma^{(1)} = s_1, s_2, \ldots$.

Consider now the evaluation of formulas:

- For a k-ary predicate p (including equality):
  $p(t_1, \ldots, t_k)|^{\alpha}_{\sigma} = I[p](t_1|^{\alpha}_{\sigma}, \ldots, t_k|^{\alpha}_{\sigma})$.
  Mere again, wc first evaluate the arguments in the rnodcl and then test $I[p]$ on them.

- For a disjunction:
  $(w_1 \vee w_2)|^{\alpha}_{\sigma} = true$ *if and only if* $w_1|^{\alpha}_{\sigma} = true$ *or* $w_2|^{\alpha}_{\sigma} = true$.
  And similarly for the other binary boolean connectives $\vee$, $\supset$, and $\equiv$.

- For a negation:

  $(\sim w)|^{\alpha}_{\sigma}$ = *true if and only if* $W|^{\alpha}_{\sigma}$ = *false.*

- For a next-time application:

  $(\bigcirc w)|^{\alpha}_{\sigma} = w|^{\alpha}_{\sigma^{(1)}}$.

  Thus $\bigcirc$ w rncans: w will be true in the *next* instant — read "next w".

- For an all- times application:

  $(\square w)|^{\alpha}_{\sigma}$ = *true if and only if* for every $k \geq 0$, $w|^{\alpha}_{\sigma^{(k)}}$ = *true,*

  i.e., w is truc for all suffix sequences of σ. Thus $\square$ w means: w is true for *all* future instants (including thc present) -- read "always w" or "henceforth w".

- For a sornc- time application:

  $(\Diamond w)|^{\alpha}_{\sigma}$ = *true if and only if* there exists a $k \geq 0$ such that $w|^{\alpha}_{\sigma^{(k)}}$ = *true,*

  i.e., w is *true* on at least onc suffix of σ. Thus $\Diamond$ w rneans: w will bc true for *some* future instant (possibly the present) -- read "sornetime w" or "eventually w".

- For an until application:

  $w_1 \mathcal{U} w_2|^{\alpha}_{\sigma}$ = *true if and only if* for sornc $k \geq 0$, $w_2|^{\alpha}_{\sigma^{(k)}}$ = *true* and for all i, $0 \leq i < k$, $w_1|^{\alpha}_{\sigma^{(i)}}$ = *true.*

  Thus $w_1 \mathcal{U} w_2$ rncans: there is a future instant in which w_2 holds, and such that *until* that instant w_1 continuously holds -- rcad "w_1 until w_2" ([KAM], [GPSS]).

- For a universal quantification:

  $(\forall u.w)|^{\alpha}_{\sigma}$ = *true if und only if* for every $d \in D$, $w|^{\alpha'}_{\sigma}$ = *true,*

  where $\alpha' = \alpha \circ [u \leftarrow d]$ is the assignment, obtained from α by assigning d to u.

- For an existential quantification:

  $(\exists u.w)|^{\alpha}_{\sigma}$ = *true if and only if* for some $d \in D$, $w|^{\alpha'}_{\sigma}$ = *true,*

  where $\alpha' = \alpha \circ [u \leftarrow d]$.

Following are some examples of temporal expressions and thcir intuitive interpretations:

$u \supset \Diamond v$	If u is presently true, v will eventually becomc true,
$\square(u \supset \Diamond v)$	Whenever u becomes true it will eventually bc followed by v.
$\Diamond \square w$	At sornc future instant w will become permanently true.
$\Diamond(w \wedge \bigcirc \sim w)$	There will be a future instant such that w is true at that instant and false at the next.
$\square \Diamond w$	Every future instant is followed by a later one in which w is true,

thus w is truc infinitely often.

▯☎♦ ⊃ □ ❖✆ If u ever becomes true, then v is true at that instant and ever after.

. $\Box u \vee (u \mathfrak{U} v)$ Either u holds continuously or it holds until an occurrcnce of v.
This is *the* weak form of the *until* operator that states that u will hold continuously until thc first occurrence of v if v ever happens or indefini tely otherwise.

o v 3 $((\sim v) \mathfrak{U} u)$ I f v ever happens, its first occurrence is preceded by (or coincides with) u.

If w is true under the model $(I, \alpha,$ a), WC say that (I, α, a) *satisfies w* or that (C, a, a) is a *(satisfying) model* for w. We dcnotc this by

$(I, \alpha,$ a) ⊨ w.

A formula w is *satisfiable* if there exists a satisfying model for it.

A formula w is *valid* if it is true in every model; in this case WC write

⊨ w.

Somctirnns we are interested in a restricted class of models C. A formula w which is true for every model in C is said lo bc *C-valid,* denoted by

c l= w.

Example:

The formula $\Diamond(w_1$ A $w_2) \supset$ (0 w_1 A 0 $w_2)$ is valid, i.e.,

⊨ $\Diamond(w_1$ **A** $w_2) \supset$ **(0** w_1 **A** $\Diamond w_2)$.

It says that if there exists an instant in which both w_1 and w_2 are true then there exists an instant in which w_1 is truc and thcre exists an instant in which w_2 is true.

Reversing the implication does not yield a valid formula, i.e.,

t# $(\Diamond$ w_1 A 0 $w_2) \supset \Diamond(w_1$ **A** $w_2)$.

For, consider an interpretation consisting of a sequcncc of states:

$\sigma :$ $s_0, s_1, \ldots$

such that w_1 is true on all odd numbered states and false elsewhere, and w_2 is true on all the even numbcred states and false on the odd ones. Then certainly both 0 w_1 and 0 w_2 are true on σ, hence 0 w_1 A 0 w_2 is true. On thc other hand, there is no state on which both w_1 and w_2 are true sirnultancously. Hence $\Diamond(w_1$ A $w_2)$ is false. Consequently thc implication is false under the interpretation σ. ∎

2. THE PROOF SYSTEM

Having defined valid formulas, we nalurally look for a deductive system in which validity can be proved. In such a system we take some of the valid formulas as axioms and provide a set of sound inference rulcs by which we hopc to be ablc to prove the other valid formulas as thcorcms. A forrnula w is a theorem of thc system either if it is an axiom of the system or has a proof in which it is derived from the axioms using the inference rulcs of the system. We denote thc fact that, w is a theorem is *provable* wilhin the systcrn by $\vdash$ w.

Our interest in thc temporal logic formalism is mainly motivated by thc applicability of this logic to proving properties of concurrent programs. Therefore, apart from developing the general basic logical proper-tics of the opcrators and their intcrrclations, we will mostly be interested in properlies that are valid ovcr computations of a given concurrent program P. Thus, the notion of validity our system will try to capture is that of a formula being truc for all possible cornputations of thc givcn program, and not necessarily ovcr an arbitrary model. This corresponds to the concept of A(P)-validity where A(P) is thc class of all models corresponding to computations of P.

We structure our proof systcm into three main layers dependent on thc univcrsal validity of thc theorems that can bc derived in each laycr. In thc first layer, called the *general part,* we deal with thc general temporal propertics of discrctc linear scqucnccs (arbitrary models). Thcorerns proved in that part arc valid for all sequences over arbitrary domains. Thcy univcrsnlly hold for arbitrary cornputations of all programs ovcr such domains, as well as for scqucnces which cannot cvcn bc dcrived as thc computations of a program. In the ncxt layer thc *domain part,* we restrict our attcntion to a particular domain D and provide tools for proving validity over models all of which are intcrpretcd over D. The third, most restrictive laycr is the *program part.* Hcre we restrict our attcntion to a particular program P and develop tools for proving validity only over models whose sequcnces are legal computations of P.

In a forthcoming paper, the program dependent part is proved to be cornplcte relative to the general temporal theory over the data domain. Wc also show that its dcpcndcncc on the particular computation rnodcl studied is modular, by presenting a similar systcm for proving properties of CSP programs.

B. GENERAL PART

We start the general part by describing first the axiomatic system for propositional temporal logic in which we do not, admit predicates or quantification.

3. THE PROPOSITIONAL TEMPORAL SYSTEM ($\square, \diamond, \circ$ AND $\mathcal{U}$)

The proof system for the propositional part, consists of the following axioms:

AXIOMS:

A1. $\vdash \sim\diamond w \equiv \square\sim w$

A2. I- $\square(w_1 \supset w_2) \supset (\square w_1 \supset \square w_2)$

A3. $\vdash \square w \supset w$

A4. *I- 0* $\sim w \equiv \sim$ *0 w*

A5. $\vdash \circ(w_1 \supset w_2) \supset (\circ w_1 \supset \circ w_2)$

A6. ⊢ ❑ lw ⊃ Ow

A7. $\vdash \square w \supset \circ\square w$

A8. ✋📭 ❑☎♦ ⊃ $\circ w)$ ⊃ ☎♦ ⊃❑♦✆

A9. $\vdash (w_1 \mathcal{U} w_2) \equiv [w_2$ **V** $(w_1$ **A** $\circ(w_1 \mathcal{U} w_2))]$

A10. $\vdash (w_1 \mathcal{U} w_2) \supset \diamond w_2$.

Axiom A1 defines 0 as the dual of Cl; it slates that at all times w is false ***if and** only **if*** it is not the case that sometimes w holds. Axiom A2 states that if universally w_1 implies w_2 then if at all times w_1 is true then so is w_2. Axiom A3 establishes the present as part of the future by stating that if w is true at all future instants it must be true at the present. Axiom A4 establishes 0 as self-dual. Consequently it implies that the next instant exists and is unique, and restricts our models Lo linear sequences (no branching). Axiom A5 is the analogue of A2 for the 0 operator. Axiom A6 states that the next instant is one of the future states. Axiorn A7 states that if w holds in all future instants it also holds in all instants which lie in the future of the next instant. Axiom A8 is the "computational induction" axiom; it states that if a property is inherited over one step transitions, it is invariant, over any suffix sequence whose first state satisfies w. Axiom A9 characterizes the *until* operator by distributing its effect into what is implied for the present and what is implied for the next instant. Axiom A10 simply states that "w_1 until w_2" implies that w_2 will eventually happen.

INFERENCE RULES:

R1. *Propositional Tautology* — PT

If u is an instance of a propositional tautology then $\vdash u$

R2. *Modus Ponens* — MP

If I- u $\supset$ v and t- u then I- v

R3. $\Box$ *Insertion* -- $\Box$I

If I- u then $\vdash \Box$ u

All these rulcs are sound. The soundness of Rl and R2 is obvious. Notc that in Rl we also include temporal inslances of tautologies; wc may substitute an arbitrary temporal formula for a proposition lcttcr in obtaining an instance. For exarnplc, thc forrnula $\Box w \supset \Box w$ is a tcmporal instance of thc tautology p $\supset$ p. To justify R3, we recall that validity of w rncans that w is true in all models, hence Cl w is also valid.

DERIVED RULES AND THEOREMS:

Before giving some theorems that can bc proved in this system, we develop several useful dcrived rules:

Propositional Reasoning -- PR

$$\frac{\begin{array}{l} t\text{-}\ (u_1\ A\ u_2\ \text{A}\ .\ .\ .\ \text{A}\ u_n)\ 3\ \text{v} \\ \text{t-}\ u_1,\ \vdash u_2,\ .\ .\ .\ ,\ \text{and} \vdash u_n \end{array}}{\text{I-V}}$$

The notation above is used Lo describc inference rules. IL has the general form

$$\frac{\vdash \varphi_1,\ \vdash \varphi_2,\ \ldots,\ \vdash \varphi_m}{\vdash \psi}$$

and means that if we have already provcd $\varphi_1, \ldots, \varphi_m$ (thc *assumptions* or premises of the rule), wc are allowed by this rule to infer ψ (the *conclusion* or *consequent* of the rule).

Proof:

Thc rule PR follows from thc propositional tautology (Rule Rl)

$$\textbf{1-}\ [(u_1\ \mathbf{A}\ u_2\ \mathbf{A}\ .\ .\ .\ \text{A}\ u_n) \supset \text{v}] \supset [u_1 \supset (u_2 \supset (\ .\ .\ .\ (u_n \supset \text{v})\ .\ .\ .))]$$

by applying MI' (Rule R2) n + 1 times. ▟

Whenever we apply this derived rule without explicitly indicating the premise

⊢ $(u_1$ A u_2 A . . . A $u_n) \supset$ v,

it means that the premise is an instance of a propositional tautology.

○ *Insertion* — 01

$$\frac{\text{t-U}}{\vdash \text{o u}}$$

Proof:

1. t-u given
2. ⊢ □ ♦ by 01
3. ⊢ **ou** by A6 and MP ∎

The first theorem that we derive in the system is:

T 1 . ⊢ $w \supset \Diamond w$

Proof:

1. ⊢ $(\Box{\sim}w) \supset$ - w by A3
2. **t-w** $\supset (\sim\Box{\sim}w)$ by PR
3. **t-w** ⊃ **ow** by **Al** and PR ∎

The theorem implies (by MP) the derived rule

◇ *Insertion* — 01

$$\frac{\vdash u}{\vdash \text{o u}}$$

T 2 . ⊢ Ow ⊃ O w

Proof:

1 . ⊢ $(\Box{\sim}w)$ 3 (O - w) by A6

2 . $\vdash (\sim \bigcirc \sim w) \supset (\sim \Box \sim w)$ by PR

3. *t - o w* ⊃ *o w* by A1, A4, and PR ▟

The following three rules (and a similar rule for the *until* operator presented later) show that all the temporal operators are monotonic in the sense that an argument may be replaced by a weaker statement yielding a weaker expression.

□ □ *Rules*
(a) $\dfrac{\text{t-u} \supset \text{V}}{\vdash \Box u \ 3\ \Box v}$ (b) $\dfrac{\vdash u \equiv v}{\vdash \Box u \equiv \Box v}$

Proof of (a):

1. t-u ⊃ v given

2. ⊢▯☎♦ ⊃ ❖✆ by □I

3. ⧫✍ ▯☎♦ ⊃ ❖✆ ⊃ (□*u* ▤ ▯❖✆ by A2

4. $\vdash \Box u \supset \Box v$ by 2, 3 and MP

Rule (b) then follows by propositional reasoning by using the tautology

[(U ⊃ v) A (v ⊃ *u*)] ≡ (*u* ≡ *v*). ▟

0 0 *Rules*
(a) $\dfrac{\textit{t-u} \supset \textit{v}}{\vdash \text{o u} \supset \text{o v}}$ (b) $\dfrac{\vdash u \equiv v}{\vdash \Diamond u \equiv \text{o v}}$

Proof of (a):

1. **t-u ⊃ v** given

2. **t- - v 3 - u** by PR

3. ⊢ □ ~ *v* ⊃ **c l - u** by❑ ICI

4. $\vdash \sim \Diamond v \supset \sim \Diamond u$ by Al and PR

5. ⊢ ◇*u* ⊃ **o v** by PR

Rule (b) then follows by propositional reasoning. ▟

0 0 *Rules*

$$\text{(a)}\ \frac{\vdash u \supset v}{\vdash \mathrm{o\ u\ 3\ o\ v}} \qquad \text{(b)}\ \frac{t\text{-}u \equiv v}{\vdash \mathrm{o\ u} \equiv \mathrm{o\ v}}$$

Proof of (a):

1.	$\vdash u \supset$ v	given
2.	t- ○($u \supset$ v)	by 01
3.	$\vdash$ ○$u \supset$ o v	by A5 and MP

Rulc (b) follows by propositional reasoning. ▟

Computational Induction Rule — CI

$$\frac{\vdash u \supset \mathrm{o\ \ u}}{\text{●📫♦} \supset \text{▯⌨☹}}$$

Proof:

1.	t-u ⊃ ○u	given
2.	⧫📫 ▯☎♦ ⊃○u)	by □I
3.	$\vdash \Box(u \supset \bigcirc u) \supset (u \supset \Box u)$	by A8
4.	$\vdash u \supset \Box u$	by 2, 3 and MP ▟

Derived Computational Induction Rule -- DCI

$$\frac{\vdash u \supset (v \wedge \bigcirc u)}{\vdash u \supset \Box v}$$

Proof:

1.	$\vdash u \supset (v \wedge \bigcirc u)$	given
2.	$\vdash u \supset \bigcirc u$	by PR
3.	$\vdash u \supset \Box u$	by CI
4.	t-u ⊃ v	by **1** and PR
5.	t-• u ⊃□v	by □□

6. $\vdash u \supset \Box v$ by 3, 5 and PR ▟

The following two thcorems show that the Cl and 0 operators are both idempotent:

T3.♦📭 ❑ ♦ ≡ □□w

Proof:

1. $\vdash \Box\Box w \supset \Box w$ by A3
2. $\vdash \Box w \supset \bigcirc\Box w$ by A7
3. $\vdash \Box w \supset \Box\Box w$ by CI
4. ⊢ □w ≡ ❑ ✋☜✋⌖ by 1, 3 and PR ▟

T4. **I - o w** ≡ **o o w**

Proof:

1. $\vdash \sim\Diamond w \equiv \Box\sim w$ by Al
2. ⊢ □∼w ≡ **0 0 - w** by T3

. . ⊢ □∼◇w ≡ . ▫ . . by 1 and El □

4. **I - E l - o w** ≡ ∼◇◇w by Al
5. $\vdash \sim\Diamond w \equiv \sim\Diamond\Diamond w$ by 1, 2, 3, 4 and PR
6. $\vdash \Diamond w \equiv \Diamond\Diamond w$ by PR ▟

Because of these last two thcorems we can collapse any string of consccutive identical modalities such as □ . . . ❑ lor 0 . . . 0 into a singlc modality of the same type.

The following theorem cstablishes thal □ is lhc dual of 0. Nolc that Λ1 statcs thal 0 is the dual of Cl, i.e., 0 w ≡ ∼ □ ∼w.

T 5 . ⊢ (O - w) ≡ (∼□w)

Proof:

1. $\vdash (\sim\sim w) \equiv w$ by PT

2. $\vdash (\Box \sim \sim w) \equiv$ ❑ ♦ by ❑ Cl

3. $\vdash (\sim \Diamond \sim w) \equiv$ ❑ ♦ by Al and PR

4. ✋📫 $(\Diamond \sim w) \equiv (\sim$❑♦✆ by PR ▟

T6. ❑☎⌖● $\supset w_2) \supset$ ☎🗀 $w_1 \supset \Diamond w_2)$

Proof:

1. t- $(w_1 \supset w_2) \equiv (\sim w_2 \supset \sim w_1)$ by PT

2. ♦📫 ❑☎⌖● $\supset w_2) \equiv \Box(\sim w_2 \supset \sim w_1)$ by ❑ ICI

3. $\vdash$❑☎ $\sim w_2 \supset \sim w_1) \supset (\Box \sim w_2 \supset \Box \sim w_1)$ by A2

4. $\vdash (\Box \sim w_2 \supset \Box \sim w_1) \equiv (\sim \Diamond w_2 \supset \sim \Diamond w_1)$ by Al and PR

5. $\vdash (\sim \Diamond w_2 \supset \sim \Diamond w_1) \equiv (\Diamond w_1 \supset \Diamond w_2)$ by PT

6. $\vdash$❑☎⌖● $\supset w_2) \supset (\Diamond w_1 \supset \Diamond w_2)$ by 2, 3, 4, 5 anti PR ▟

The following theorems show the interaction between the temporal and the boolean operators.

T7. I- ❑ (wl A $w_2) \equiv (\Box w_1$ A $\Box w_2)$

Proof:

1. $\vdash (w_1 \wedge w_2) \supset w_1$ by PT

2. $\vdash$ ❑ ☎⌖● ✌ $w_2) \supset \Box w_1$ by ❑ ICI

.3. $\vdash (w_1 \wedge w_2) \supset w_2$ by PT

4. ♦📫 ❑ (wl A$w_2) \supset$❑ w2 by ❑ IEJ

5. $\vdash$ ❑ (wl A $w_2) \supset (\Box w_1$ A ❑ w2) by 2, 4 and PR

6. $\vdash w_1 \supset (w_2 \supset w_1$A $w_2)$ by PT

7. 📫 ❑ ♦● $\supset$ ❑ $(w_2 \supset (w_1$A $w_2))$ by □□

. ✍📫 $\vdash$ ❑$(w_2 \supset (w_1$ A $w_2)) \supset (\Box w_2 \supset \Box(w_1$ A $w_2))$ by A2

9. $\vdash \Box w_1 \supset (\Box w_2 \supset$ ❑ (wl A $w_2))$ by 7, 8 and PR

10. $\vdash (\Box w_1$ A $\Box w_2) \supset$ ❑ (wl A $w_2)$ by PR

11. ⊢ ❑ (wl Aw_2)≡(□w_1A❑ w2) by 5, 10 and PR ▟

T8. ⊢ $\Diamond(w_1 \lor w_2) \equiv (\Diamond w_1 \lor \Diamond w_2)$

Proof:

1. ⊢ □$\sim(w_1 \lor w_2) \equiv$ ❑ (-wl A $\sim w_2$) by PT and Cl □

2. ⊢ ❑ $(\sim w_1$A $\sim w_2) \equiv (\Box \sim w_1$ A $\Box \sim w_2)$ by T7

3. ⊢ $(\Box \sim w_1$ A $\Box \sim w_2) \equiv \sim(\sim \Box \sim w_1 \lor \sim \Box \sim w_2)$ by PR

4. ⊢□$\sim(w_1$v$w_2)\equiv \sim(\sim$ ❑ $\sim w_1$ ❖ $\sim \Box \sim w_2)$ by 1, 2, 3 and PR

5. ⊢ $\sim \Diamond(w_1$. $w_2) \equiv \sim(\Diamond w_1$ **v** $\Diamond w_2)$ by A1 and PR

6. **t-** $\Diamond(w_1$ **v** $w_2) \equiv$ **(0** w_1 **v** $\Diamond w_2)$ by PR ▟

Note that because of the universal character of Cl it can be distributed over A (Theorem T7), while 0, which is of existential character can be distributed over V (Theorem T8). Next, wc show that interchanging a temporal operator with a boolean operator of the opposite character yields implication in one direction only; the implication is not necessarily truc in the other direction.

T9.♦🗇 (□w_1 ❖ ❑ w2)⊃ ❑ (Wlvw_2)

Proof:

1. ⊢ □w_1 ⊃❑☎♦🗀❖ w_2) by PT and El Cl

2. ⊢□ w_2⊃ ❑ ☎⚲● ❖ w_2) by PT and □□

3. ⊢(□w_1□w_2)⊃ ❑ ☎⚲● ❖ w_2) by 1, 2 and PR ▟

T10. t - $\Diamond(w_1$ A $w_2) \supset (\Diamond w_1$ A $\Diamond w_2)$

Proof:

1. t- $\Diamond(w_1$ A $w_2)$ 3 $\Diamond w_1$ by PT and 0 0

2. ⊢ $\Diamond(w_1$ A $w_2) \supset \Diamond w_2$ by PT and 0 0

3. ⊢ $\Diamond(w_1$ A $w_2) \supset (\Diamond w_1$ A $\Diamond w_2)$ by 1, 2 and PR ▟

T11. $\vdash (\Box w_1\ A\ 0\ \ w_2) \supset \Diamond(w_1\ A\ \ w_2)$

Proof:

1. I- $w_1 \supset (w_2 \supset (w_1 \wedge w_2))$ by PT
2. $\vdash \Box w_1 \supset \Box\ (w_2 \supset (w_1\ A\ w_2))$ by □□
3. **⊢** □ $(\mathbf{w2} \supset (w_1\ A\ w_2)) \supset (\Diamond w_2 \supset \Diamond(w_1\ A\ w_2))$ by T6
4. $\vdash \Box w_1 \supset (\Diamond w_2 \supset \Diamond(w_1\ A\ w_2))$ by 2, 3 and PR
5. $\vdash (\Box w_1\ A\ \ \Diamond w_2) \supset \Diamond(w_1\ A\ \ w_2)$ by PR ∎

Next we consider the commutativity properties of the *next* operator 0. In view of A4, 0 is self-dual and can be considered Lo be of both existential and universal character. Indeed it cornmutes with every other boolean or temporal operator as well as with quantifiers.

'1'12. $\vdash O(w_1\ A\ w_2) \equiv (O w_1\ A\ 0\ \ w_2)$

Proof:

1. $\vdash w_1\ 3\ (w_2 \supset (w_1\ A\ w_2))$ by PT
2. **t-** $O\ w_1 \supset O(w_2 \supset (w_1\ A\ w_2))$ by 0 0
3. $\vdash O(w_2 \supset (w_1\ A\ w_2)) \supset (0\ w_2 \supset O(w_1\ A\ w_2))$ by A5
4. $\vdash O w_1\ A\ (O w_2 \supset O(w_1\ A\ w_2))$ by **2, 3** and PR
5. $\vdash (O\ w_1\ A\ \ O w_2) \supset O(w_1\ A\ \ w_2)$ by PR
6. $\vdash (w_1 \wedge w_2) \supset w_1$ by PT
7. $\vdash O(w_1\ A\ w_2) \supset O w_1$ by 0 0
8. $\vdash (w_1 \wedge w_2) \supset w_2$ by PT
9. $\vdash O(w_1\ A\ w_2) \supset O w_2$ by 0 0
10. $\vdash O(w_1\ A\ w_2) \supset (O w_1\ A\ O w_2)$ by 7, 9 and PR
11. $\vdash O(w_1\ A\ \ w_2) \equiv (O w_1\ A\ \ O w_2)$ by 5, 10 and PR ∎

T13. t - $O(w_1\ v\ \ w_2) \equiv (O w_1\ v\ \ O w_2)$

Proof:

1. $\vdash \bigcirc(\sim w_1 \text{ A } \sim w_2) \equiv [(\bigcirc \sim w_1) \text{ A } (\bigcirc \sim w_2)]$ by T12
2. $\vdash \bigcirc(\sim w_1 \text{ A } \sim w_2) \equiv [(\sim \bigcirc w_1) \text{ A } (\sim 0\ w_2)]$ by A4 and PR
3. $\vdash \bigcirc \sim(w_1 \vee w_2) \equiv [(\sim \bigcirc w_1) \text{ A } (\sim \bigcirc w_2)]$ by 0 0 and PR
4. $\vdash \sim \bigcirc(w_1 \text{ v } w_2) \equiv \sim(\bigcirc w_1 \mathbf{v} \bigcirc w_2)$ by A4 and PR
5. $\vdash \bigcirc(w_1 \text{ v } w_2) \equiv (\bigcirc w_1 \mathbf{v} \bigcirc w_2)$ by PR

T14. t - $\bigcirc(w_1 \supset w_2) \equiv (0\ w_1 \supset \bigcirc w_2)$

Proof:

1. $\vdash \bigcirc(\sim w_1 \text{ v } w_2) \equiv (0 \sim w_1) \vee (\bigcirc w_2)$ by T13
2. I- $\bigcirc(\sim w_1 \text{ v } w_2) \equiv (\sim \bigcirc w_1) \text{ v } (\bigcirc w_2)$ by A4 and PR
3. I - $\bigcirc(w_1 \supset w_2) \equiv (\bigcirc w_1 \supset \bigcirc w_2)$ by 0 0 and PR

T15. $\vdash \bigcirc(w_1 \equiv w_2) \equiv (\bigcirc w_1 \equiv 0\ w_2)$

Proof:

1. t- $[\bigcirc(w_1 \supset w_2) \text{ A } \bigcirc(w_2 \supset w_1)] \equiv [(\text{owl } 3\ 0\ w_2) \text{ A } (\bigcirc w_2 \supset \text{owl})]$ by 7'14 and PR
2. $\vdash \bigcirc[(w_1 \supset w_2) \text{ A } (w_2 \supset w_1)] \equiv [(\bigcirc w_1 \supset \bigcirc w_2) \text{ A } (\bigcirc w_2 \supset \bigcirc w_1)]$ by T12 and PR
3. $\vdash \bigcirc(w_1 \equiv w_2) \equiv (\bigcirc w_1 \equiv \bigcirc w_2)$ by 0 0 and PR

The previous theorems show that the next operator, 0, commutes with each of the boolean operators. The following two theorems establish commutation of 0 with Lhc temporal operators Cl and 0.

T16. $\vdash \bigcirc \Box w \equiv$ ❑ ow

Proof:

1. I- ow $\supset$ (w $\supset \bigcirc w$) by PT

2. ✋🖆□○w ⊃❑☎♦ ⊃○w) by □□

📎 ⊢❑☎♦ ⊃ ○w) ⊃ ○□(w ⊃ ○w) by A7

4. ✋🖆○□(w ⊃ ○w) ⊃ ○(w ⊃❑ ♦◐ by A8 and ○ 0

5. ✋🖆○(w⊃ ❑♦◐ ⊃ (○w ⊃ ○□w) by A5

6. ⊢ □○w ⊃ (○w ⊃ ○□w) by 2, 3, 4, 5 and PR

7. ⊢ □○w ⊃ o w by A3

8. t-•ow ⊃oow by 6, 7 and PR

9. t-oclw ⊃ooow by A7 and 0 0

10. ⊢ ○□w ⊃ ❑ □◆♦ by CI

11. ⊢ ○□w ⊃ o w by A3 and 0 0

12. ⊢ □○□w ⊃ n o w by □□

13. I- ○□w ⊃ □○w by 10, 12 and PR

14. ♦🖆□◆♦ ≡ ❑ □♦ by 8, 13 and PR ▟

T17. ⊢ ○◇w ≡ ◇○w

Proof:

1. ⊢ ○□~w ≡ 0 0 - w by T16

2. I--0ow ≡~◇○w by Al, A4, □ Cl, 0 0 and PR

3. I-oow ≡oow by PR ▟

T18. ⊢ □ ◇□w ≡ ◇□w

Proof:

1. I-• ou ⊃ ◇□w by A3

. 2. t-•w ⊃ouw by A7

,3. ⊢ ◇□w ⊃ ◇○□w by ○○

4. ⊢ ◇○□w 3 ○◇□w by 1'17 and PR

5. $\vdash \Diamond \Box w$ **3** $\bigcirc \Diamond \Box w$ by 3, 4 and PR

6. $\vdash \Diamond \Box w \supset \Box \Diamond \Box w$ by CI

7. $\vdash \Box \Diamond \Box w \equiv \Diamond \Box w$ by 1, 6 and PR ∎

T19. $\vdash \Diamond \Box \Diamond w \equiv$ ❑ Ow

Proof: By duality from T18.

These last two theorems together with T3 and T4 ($\Box$ Cl $w \equiv$ ❑ l?u and $\Diamond \Diamond w \equiv$ Ow, respectively) give us a normal prefix form for a string of the form

$$m_1 m_2 \ldots m_k(w),$$

where each m_i is either El or 0. We use first T2 and T3 to collapse any substring of the form Cl" and $\Diamond^n$ to a single $\Box$ or 0. What remains must be a string of alternating $\Box$ and 0. If it contains more than one operator then it is equivalent by T18 and T19 to a string with just two operators -- the last two. Consequently any string such as the above must be equivalent to one of the following four possibilities:

❑ ●♦ $\Diamond w$, ❑ lOw or $\Diamond \Box w$.

In the more general case that the string also contains some occurrences of the next-time operator 0, we may use the commutation of 0 with both Cl and 0 to obtain Lhc four normal forms:

$\bigcirc^k \Box w$, $\bigcirc^k \Diamond w$, $\bigcirc^k \Box \Diamond w$ and $\bigcirc^k \Diamond \Box w$

for some $k \geq 0$.

T20. t- ❑ lw $\equiv$ (w A $\bigcirc \Box w$)

Proof:

1. $\vdash \Box w$ 3 w by A3

2. **t-•JW ⊃ o o w** by A7

3. $\vdash$ ❑ u) ⊃ (w A $\bigcirc \Box w$) by 1, 2 and PR

4. **t- oclw** $\supset \bigcirc(w$ **A** $\bigcirc \Box w)$ by 00

5. $\vdash (w$ A $\bigcirc \Box w) \supset \bigcirc(w$ A $\bigcirc \Box w)$ by PR

6. ⊢ (w A ○ □ *w*) ⊃ ❑ (w A ○ □ *w*) by CI

7. ♦📫 ❑ (wA○ □ *w*)⊃❑ w by PT and □ Cl

8. ⊢ (*w* A ○ □ *w*) ⊃ ❑ w by 6, 7 and PR

9. ⊢ □ *w* ≡ (w A ○ □ *w*) by 3, 8 and PR ▟

T21. ⊢ O w ≡ (w v ○ ◇ *w*)

Proof:

1. ⊢ □ ~*w* ≡ (~*w* A ○ □ ~*w*) by T20

2. ⊢ ~ ◇ *w* ≡ ~(*w* **v -0 0 - w)** by Λ1 and PR

3. ⊢ ~ ○ □ ~*w* **3 o o w** by Λ4, Al, 0 0 and PR

4. **I-** ◇ *w* ≡ (*w* **v** ○ ◇ *w*) by 2, 3 and PR ▟

Theorerns T20 and T21 give a fixpoint characterization of the □ and 0 operators respectively. They each give an equation using only boolean operators, the formula w and the operator 0. The solutions to these equations are Cl *w* and 0 *w* respectively. This shows that in some sense 0 is the most basic operator since the other operators may be defined by means of fixpoint equations using 0. Axiorn A9 similarly characterizes the U operator by a Gxpoint equation.

7 22. ⊢ (w A ◇ ~*w*) ⊃ ◇(*w* A ○ ~*w*).

This is the dual of the "computational induction" axiom Λ8. It states that if *w* is true now and is false sometirne in the future, then there exists some instant such that w is true at that instant and false at the next.

Proof:

1. ✋📫 □(*w* ⊃ ▯ *w*) ⊃ ☎♦ ⊃❑✍✝✆ by A8

2. ♦📫 ~(*w* ⊃❑ ♦✆ ⊃ ~ □(*w* ⊃ ○ *w*) by PR

3. ⊢ (w A ~ □ *w*) ⊃ ◇ ~(*w* ⊃ ○ *w*) by T5 and PR

4 . t - ◇ ~(*w* ⊃ ○ *w*) ≡ ◇(*w* A ~ ○ *w*) by PT and 0 0

5. ⊢ (w A ~ □ *w*) ⊃ ◇(*w* A ~ ○ *w*) by 3, 4 and PR

6. I- (w A ◇ ~*w*) ⊃ ◇(*w* A ○ ~*w*) by T5, A4 and PR ▟

The following derived rules correspond to proof rules existing in most axiomatic verification systems:

Consequence Rules

□Q *rule*	◇Q□♦●‖	OQ *rule*
I- u_1 3 u_2	⊢ $u_1 \supset u_2$	I- u_1 3 u_2
⊢ $u_2 \supset$ □♦♦	⊢ $u_2 \supset \Diamond v_1$	t- $u_2 \supset$ 0 v_1
t- $v_1 \supset v_2$	⊢ v_1 3 v_2	⊢ v_1 3 v_2
t- $u_1 \supset \Box v_2$	⊢ $u_1 \supset \Diamond v_2$	I- $u_1 \supset$ Ov_2

Proof of ◇Q:

1. ⊢ $u_1 \supset u_2$ given

2. ⊢ u_2 3 0 v_1 given

3 t- $v_1 \supset v_2$ given

4. ⊢ $\Diamond v_1 \supset \Diamond v_2$ by 3 and 0 0

5. ⊢ $u_1 \supset \Diamond v_2$ by t, 2, 4 and PR ▟

The ❑ Q and OQ rules are proved similarly by the □ Cl-rule and 0 O-rule, respectively.

Concatenation Rules

□ ✋ □♦●‖	◇C rule
⊢ $u \supset \Box v$	⊢ $u \supset$ □❖
⊢ $v \supset \Box w$	●⎙❖ $\supset \Diamond w$
♦⎙ $u \supset$ □ ♦	⊢ $u \supset$ □♦

Proof of UC:

1. ⊢ u 3 $\Box v$ given

2. ⊢ $v \supset \Box w$ gi vcn

3. ⊢ $\Box v$ **3** $\Box\Box w$ by 2 and □ Cl

4. ⊢ $\Box v \supset$ ❑ ♦ by T3 and PR

5. ⊢ $u \supset \Box w$ by **1,** 4 and PR ▟

The OC rule is proved similarly by the 0 O-rule. Note that the corresponding OC rule does not hold.

UNTIL DERIVED RULES AND THEOREMS:

Right Until Introduction -- $\mathcal{RU}I$

$$l - w \ \supset \Diamond v$$

$$I\text{-} \ w \supset [v \ v \ (u \ A \ \bigcirc w)]$$

$$\vdash w \supset (u \mathcal{U} v)$$

Proof:

1. $t - w \ 3 \ \Diamond v$ given

2. $\vdash w \supset [v \ v \ (u \ A \ 0 \ w)]$ given

3. $I\text{-} \ [v \ v \ (u \ A \ \bigcirc(u\mathcal{U}v))] \supset (u\mathcal{U}v)$ by A9 and PR

4. $\vdash \sim(u\mathcal{U}v) \supset [\sim v \ A \ (\sim u \ v \ 0 \ \sim(u\mathcal{U}v))]$ by A4 and PR

5. $\vdash [w \ A \ \sim(u\mathcal{U}v)] \supset [\sim v \ A \ \bigcirc \ w \ A \ \bigcirc \sim(u\mathcal{U}v)]$ by 2, 4 and PR

6. $\vdash [w \ \wedge \sim(u\mathcal{U}v)] \supset [\sim v \ A \ \bigcirc(w \ A \ \sim(u\mathcal{U}v))]$ by T12 and PR

7. $t\text{-} \ [w \ A \ \sim(u\mathcal{U}v)] \supset \Box \sim v$ by DCI, taking u to be $w \ A \ \sim(u\mathcal{U}v)$ and v to be $\sim v$

8. $\vdash [w \ A \ \sim(u\mathcal{U}v)] \supset \sim \Box \sim v$ by 1, T5 and PR

9. $\vdash w \supset (u\mathcal{U}v)$ by 7, 8 and PR ∎

The $\mathcal{RU}I$ rule, together with axioms A9 and A10, can bc viewed as a characterization of the $u\mathcal{U}v$ construct *as a maximal* solution of thc two implications:

$$(*) \quad \begin{cases} x \supset [v \ V \ (u \ A \ \bigcirc \ x)] \\ x \supset \Diamond v \end{cases}$$

The ordering by which maximality is defined is the ordering inducd by defining *false* $\sqsubset$ *true.*

Axioms A9 and A1 0 imply that

$$(u\mathcal{U}v) \supset [v \ V \ (u \ A \ \bigcirc u\mathcal{U}v)]$$

$$(u\mathcal{U}v) \supset 0 \ \ v$$

Thus they show $x = u\mathcal{U}v$ to be a solution of the implications (t). The rule $\mathcal{RU}I$ states that any other solution $x = w$ must satisfy $w \supset (u\mathcal{U}v)$ which implies that whenever w is true so is $u\mathcal{U}v$. Interpreted in our ordering this is representable as $w \sqsubseteq (u\mathcal{U}v)$. Thus $x = u\mathcal{U}v$ is the maximal solution to $(*)$.

An intuitive explanation as to why $u\mathcal{U}v$ is indeed the maximal solution of $(*)$ can be given as follows:

Lct w be any proposition satisfying (t) cverywhcre in a sequence $\sigma = s_0, s_1, \ldots$. WC notc that $(*)$ may havc many solutions. In particular x = *false* is a trivial solution. Howevcr an obvious property of every solution w is that if w is truc in somc statc s_i, this state must satisfy u and the next statc s_{i+1} must also satisfy w unless s_i satisfies v. Thus once w is truc it can stop bcing true only in a v-state. In view of thc sccond implication such a v-state is guaranteed. Conscqucntly whencver w is true in a statc, $u\mathcal{U}v$ must also be truc in that state.

Left Until Introduction — L𝒰I

$$\vdash [\text{v V (u A } \bigcirc w)] \supset \text{w}$$

$$\overline{\text{I- } (u\mathcal{U}v) \supset \text{w}}$$

Proof:

1.	$\vdash$ [v v(u A $\bigcirc w$)] $\supset$ w	given
2.	$\vdash$ uuv $\supset$ [v v (u A $\bigcirc(u\mathcal{U}v)$)]	by A9 and PR
3.	$\vdash \sim w \supset [\sim v$ A $(\sim u$ v 0 $\sim w)]$	by 1, A4 and PR
4.	$\vdash [u\mathcal{U}v$ A $\sim w] \supset [\sim v$ A u A $\bigcirc(u\mathcal{U}v)$ A o-w]	by 2, 3 and PR
5.	$\vdash [u\mathcal{U}v$ A $\sim w] \supset [\bigcirc(u\mathcal{U}v)$ ‘A o-w]	by PR
6.	$\vdash [u\mathcal{U}v$ A $\sim w] \supset \bigcirc(u\mathcal{U}v$ A $\sim w)$	by T12 and PR
7.	I- $[u\mathcal{U}v$ A $\sim w] \supset \Box(u\mathcal{U}v$ A $\sim w)$	by CI
8.	$\vdash [u\mathcal{U}v$ A $\sim w] \supset \sim v$	by 3 and PR
9.	$\vdash$ ❑ (uuvA$\sim w)\supset\Box\sim v$	by □□
10.	$\vdash [u\mathcal{U}v$ A $\sim w] \supset \sim\Diamond$v	by 7, 9, Al and PR
11.	$\vdash [u\mathcal{U}v \wedge \sim w] \supset \Diamond v$	by A10 and PR
12.	$\vdash u\mathcal{U}v \supset$ w	by 10, 11 and PR ■

The L𝒰I rulc, togcther with axiorn A9, can be vicwcd as a characterization of the $u\mathcal{U}v$ construct as the *minimal* solution of lhc implication:

$$(**)\ [\text{V V (u A } \bigcirc x)] \supset \text{x}$$

Axiom A9 implies that x = $u\mathcal{U}v$ is a solution of $(**)$. Thc L𝒰I rule states that any other solution of $(**)$, x = w, is implied by $u\mathcal{U}v$. This rneans that whenever $u\mathcal{U}v$ is truc so is w, which is interpretable in our ordering as $u\mathcal{U}v \sqsubseteq$ w. Thus $u\mathcal{U}v$ is the minimal of all possible solutions.

Note that $(**)$ posscsscs many solutions. [n particular x = *true* is a trivial solution. Howcvcr, thc minimal solution is unique and is givcn by $u\mathcal{U}v$.

UU *Rules*

$$\text{(a)}\ \begin{array}{l}\vdash u_1 \supset u_2\\ \vdash v_1 \supset v_2\\ \hline \text{t-}\ u_1 \mathcal{U} v_1 \supset u_2 \mathcal{U} v_2\end{array} \qquad \text{(b)}\ \begin{array}{l}\vdash u_1 \equiv u_2\\ \vdash v_1 \equiv v_2\\ \hline \vdash u_1 \mathcal{U} v_1 \equiv u_2 \mathcal{U} v_2\end{array}$$

Proof of (a):

1.	$\vdash u_1 \supset u_2$	given
2.	$\vdash v_1 \supset v_2$	given
3.	$\vdash [v_2 \text{ v } (u_2 \text{ A } \bigcirc(u_2 \mathcal{U} v_2))] \text{ 3 } u_2 \mathcal{U} v_2$	by A9
4.	$\vdash [v_1 \text{ v } (u_1 \text{ A } \bigcirc(u_2 \mathcal{U} v_2))] \supset u_2 \mathcal{U} v_2$	by 1, 2, 3 and PR
5.	$\vdash u_1 \mathcal{U} v_1 \supset u_2 \mathcal{U} v_2$	by LUI

The proof of part (b) follows from (a) by propositional reasoning and the symmetric application of (a). ■

This rule together with the ❑ □, 0 0 and 0 0 rules show that all the temporal operators are monotonic in all their arguments.

T23. $\vdash (\sim w)\mathcal{U} w \equiv \text{O w}$

Proof:

1.	$\vdash (\sim w)\mathcal{U} w \supset \textbf{o w}$	by A10
2.	$\vdash \Diamond w \supset [\text{w v } \bigcirc \Diamond w]$	by T21 and PR
3.	$\vdash \text{o w} \supset [\text{w v } (\sim w \text{ A } \bigcirc \Diamond w)]$	by PR
4.	$\vdash \Diamond w \supset \text{o w}$	by PT
5.	$\vdash \text{o w} \supset (\sim w)\mathcal{U} w$	by 3, 4 and RUI
6.	$\vdash (\sim w)\mathcal{U} w \equiv \text{OW}$	by 1, 5 and PR ■

T24. $\vdash (\Box w_1 \text{ A } \Diamond w_2) \supset (w_1 \mathcal{U} w_2)$

Proof:

1.	$\vdash [\Box w_1 \textbf{ A } \Diamond w_2] \supset \Diamond w_2$	by PR

2 . $\vdash [\Box w_1$ A $\Diamond w_2] \supset [(w_1$ A $\bigcirc\Box w_1) \wedge (w_2 \vee \bigcirc\Diamond w_2)]$ by PR, T20 and T21

3 . $\vdash (\Box w_1$ A $\Diamond w_2) \supset [w_2 \vee (w_1$ A $\bigcirc\Box w_1$ A $\bigcirc\Diamond w_2)]$ by PR

4. $\vdash (\Box w_1$ A $\Diamond w_2) \supset [w_2 \vee (w_1 \wedge \bigcirc(\Box w_1 \wedge \Diamond w_2))]]$ by T12 and PR

5. **I-** $[\Box w_1$ **A** $\Diamond w_2] \supset w_1 \mathfrak{U} w_2$ by 1, 4 and RUI,

taking w to be Cl w_1 A 0 w_2, u to be w_1, and v to be w_2 ∎

T25. t- $(w_1 \mathfrak{U} w_2) \mathfrak{U} w_2 \equiv w_1 \mathfrak{U} w_2$

Proof:

1. I - $(w_1 \mathfrak{U} w_2) \mathfrak{U} w_2 \supset [w_2$ v $w_1 \mathfrak{U} w_2]$ by A9 and PR

2. **I-** $w_2 \supset w_1 \mathfrak{U} w_2$ by A9 and PR

3. $\vdash (w_1 \mathfrak{U} w_2) \mathfrak{U} w_2 \supset w_1 \mathfrak{U} w_2$ by 1, 2 and PR

4. $\vdash w_1 \mathfrak{U} w_2 \supset \Diamond w_2$ by A10

5. $\vdash w_1 \mathfrak{U} w_2 \supset [w_2 \vee (w_1$ A $\bigcirc(w_1 \mathfrak{U} w_2))]$ by A9 and PR

6. $\vdash w_1 \mathfrak{U} w_2 \supset [w_2 \vee (w_1 \mathfrak{U} w_2$ A $\bigcirc(w_1 \mathfrak{U} w_2))]$ by PR

7. $\vdash w_1 \mathfrak{U} w_2 \supset (w_1 \mathfrak{U} w_2) \mathfrak{U} w_2$ by 4, 6 and RUI

8. $\vdash (w_1 \mathfrak{U} w_2) \mathfrak{U} w_2 \equiv w_1 \mathfrak{U} w_2$ by 3, 7 and PR ∎

T26. I- $w_1 \mathfrak{U} w_2 \equiv w_1 \mathfrak{U}(w_1 \mathfrak{U} w_2)$

Proof:

1. **I-** $w_2 \supset w_1 \mathfrak{U} w_2$ by A9 and PR

2. **I-** $w_1 \mathfrak{U} w_2$ **3** $w_1 \mathfrak{U}(w_1 \mathfrak{U} w_2)$ by UU

3. I- $w_1 \mathfrak{U}(w_1 \mathfrak{U} w_2) \supset [w_1 \mathfrak{U} w_2 \vee [w_1$ A $\bigcirc(w_1 \mathfrak{U}(w_1 \mathfrak{U} w_2))]]$ by A9 and PR

4. **I-** $w_1 \mathfrak{U}(w_1 \mathfrak{U} w_2)$ **3** $\{w_2 \vee [w_1 \wedge \bigcirc(w_1 \mathfrak{U} w_2)] \vee [w_1 \wedge \bigcirc(w_1 \mathfrak{U}(w_1 \mathfrak{U} w_2))]\}$
by A9 and PR

5 . $\vdash w_1 \mathfrak{U}(w_1 \mathfrak{U} w_2)$ **3** $\{w_2 \vee [w_1$ A $\bigcirc(w_1 \mathfrak{U} w_2 \vee w_1 \mathfrak{U}(w_1 \mathfrak{U} w_2))]\}$ by T13 and PR

6. $\vdash [w_1 \mathfrak{U} w_2$ **v** $w_1 \mathfrak{U}(w_1 \mathfrak{U} w_2)] \supset w_1 \mathfrak{U}(w_1 \mathfrak{U} w_2)$ by 2 and PR

7. $\vdash w_1 \mathcal{U}(w_1 \mathcal{U} w_2) \supset \{w_2$ V $[w_1$ A $\bigcirc(w_1 \mathcal{U}(w_1\ \mathcal{U} w_2))]\}$
by 6 with 0 0, 5, and PR

8. $\vdash w_1 \mathcal{U}(w_1 \mathcal{U} w_2) \supset \Diamond(w_1 \mathcal{U} w_2)$ by A10

9. $\vdash w_1 \mathcal{U} w_2 \supset \Diamond w_2$ by Al0

10. I - $\Diamond(w_1 \mathcal{U} w_2) \supset$ 0 $\Diamond w_2$ by *0 0*

11. $\vdash w_1 \mathcal{U}(w_1 \mathcal{U} w_2) \supset \Diamond w_2$ by 8, 10, T4 and PR

12. $\vdash w_1 \mathcal{U}(w_1 \mathcal{U} w_2)$ 3 $w_1 \mathcal{U} w_2$ by 11, 7 and RUI,
taking w to be $w_1 \mathcal{U}(w_1 \mathcal{U} w_2)$, u to be w_1, and v to be w_2

15. $\vdash w_1 \mathcal{U} w_2 \equiv w_1 \mathcal{U}(w_1 \mathcal{U} w_2)$. by 2, 12 and PR ▟

U *Insertion* -- UI

(a) $\dfrac{\text{t - V}}{\vdash u \mathcal{U} v}$ for an arbitrary u

(b) $\dfrac{\text{l - u , } \vdash \Diamond v}{\text{t- uuv}}$

Proof:

(a) 1. t-v given

2. t- v 3 uuv by A9 and PR

3. ⊢ uuv by 1, 2 and PR

(b) 1. l-u given

2 $\vdash \Diamond v$ given

3. ⊢ ▯ ♦ by l and □I

4. $\vdash (\Box u$ A $\Diamond v) \supset u \mathcal{U} v$ by T24

5. ⊢ uuv by 2, 3, 4 and PR ▟

U *Concatenation* -- *UC*

$\vdash v_1$ *3* $u \mathcal{U} v_2$

I· v_2 3 $u \mathcal{U} v_3$

I- $v_1 \supset u \mathcal{U} v_3$

Proof:

1. $\vdash v_1 \ 3 \ u\mathfrak{U}v_2$ given
2. $\vdash v_2 \supset u\mathfrak{U}v_3$ given
3. t- $u\mathfrak{U}v_2 \supset u\mathfrak{U}(u\mathfrak{U}v_3)$ by UU
4. $\vdash v_1 \ 3 \ u\mathfrak{U}(u\mathfrak{U}v_3)$ by 1, 3 and PR
5. $\vdash v_1 \supset u\mathfrak{U}v_3$ by T26 and PR 4

T27. I - $[\Box w_1 \ \text{A} \ w_2\mathfrak{U}w_3] \supset (w_1 \ \text{A} \ w_2)\mathfrak{U}(w_1 \wedge w_3)$

Proof:

1. $\vdash w_2\mathfrak{U}w_3 \supset \Diamond w_3$ by A10
2. $\vdash [\Box w_1 \ \text{A} \ w_2\mathfrak{U}w_3] \supset (\Box w_1 \ \text{A} \ \Diamond w_3)$ by PR
3. $\vdash [\Box w_1 \ \text{A} \ w_2\mathfrak{U}w_3] \supset \Diamond(w_1 \ \text{A} \ w_3)$ by T11 and PR
4 . $\vdash w_2\mathfrak{U}w_3 \supset [w_3 \ \text{V} \ (w_2 \ \text{A} \ \text{O}(w_2\mathfrak{U}w_3))]$ by A9 and PR
5. $\vdash [\Box w_1 \ \text{A} \ w_2\mathfrak{U}w_3] \supset [(\Box w_1 \ \text{A} \ w_3) \ \text{V} \ (\Box w_1 \ \text{A} \ w_2 \ \text{A} \ \text{O}(w_2\mathfrak{U}w_3))]$ by PR
6. $\vdash (\Box \ w_1 \ \text{A} \ w_3) \supset (w_1 \ \text{A} \ w_3)$ by A3 and PR
7. I- $[\Box w_1 \ \text{A} \ w_2 \ \text{A} \ \text{O}(w_2\mathfrak{U}w_3)] \supset [w_1 \ \text{A} \ w_2 \ \text{A} \ \text{O} \ \Box \ \text{wl} \ \text{A} \ \text{O}(w_2\mathfrak{U}w_3)]$
by T20 and **PR**
8. t - $[\Box w_1 \ \text{A} \ w_2 \ \text{A} \ \text{O}(w_2\mathfrak{U}w_3)] \supset [(w_1 \ \text{A} \ w_2) \ \text{A} \ \text{O}(\Box w_1 \ \text{A} \ w_2\mathfrak{U}w_3)]$
by T12 and **PR**
9. $\vdash [\Box w_1 \ \text{A} \ w_2\mathfrak{U}w_3] \supset \{(w_1 \ \text{A} \ w_3) \ \text{V} \ [(w_1 \ \text{A} \ w_2) \ \text{A} \ \text{O}(\Box w_1 \ \text{A} \ w_2\mathfrak{U}w_3)]\}$
by 5, 6, 8 and PR
10. t- $[\Box w_1 \ \text{A} \ w_2\mathfrak{U}w_3] \supset (w_1 \ \text{A} \ w_2)\mathfrak{U}(w_1 \ \text{A} \ w_3)$ by 3, 9 and R$\mathfrak{U}$I ■

The next theorem displays the commutation relation between the 0 and the $\mathfrak{U}$ operators.

T28.I - $(\text{O} w_1)\mathfrak{U}(\text{O} w_2) \equiv \text{O}(w_1\mathfrak{U}w_2)$

Proof:

1. **I-** $w_1\mathfrak{U}w_2 \equiv [w_2 \ \text{V} \ (w_1 \ \text{A} \ \text{O}(w_1\mathfrak{U}w_2))]$ by A9

2 . $\vdash \mathrm{O}(w_1 \mathcal{U} w_2) \equiv [w_2 \vee (\mathrm{O}\, w_1 \text{ A } \mathrm{O}\ \mathrm{O}(w_1 \mathcal{U} w_2))]$
by T12, T13, 0 0 and PR

3. $\vdash$ $[\mathrm{O}\, w_2 \ \vee (\mathrm{O}\, w_1 \text{ A } \mathrm{O}\ \mathrm{O}(w_1 \mathcal{U} w_2))] \supset \mathrm{O}(w_1 \mathcal{U} w_2)$ by PR

4 . t - $(\mathrm{O}\, w_1)\mathcal{U}(\mathrm{O}\, w_2) \supset \mathrm{O}(w_1 \mathcal{U} w_2)$ by LUI, taking w to be $w_1 \mathcal{U} w_2$

5. $\vdash$ $w_1 \mathcal{U} w_2 \supset \diamond w_2$ by A10

6. I - $\mathrm{O}(w_1 \mathcal{U} w_2) \supset \mathrm{O} \diamond w_2$ by 0 0

7. t - $\mathrm{O}(w_1 \mathcal{U} w_2) \supset \diamond\, \mathrm{O}\, w_2$ by T17 and PR

8. ⊦ $\mathrm{O}(w_1 \mathcal{U} w_2) \supset \{\mathrm{O}\, w_2 \vee [\mathrm{O}\, w_1 \text{ A } \mathrm{O}\, \mathrm{O}(w_1 \mathcal{U} w_2)]\}$ by 2 and PR

9. $\vdash \mathrm{O}(w_1 \mathcal{U} w_2) \supset (\mathrm{O}\, w_1)\mathcal{U}(\mathrm{O}\, w_2)$ by 7, 8 and RUT,
taking w to be $\mathrm{O}(w_1 \mathcal{U} w_2)$, u to be 0 w_1, and v to be 0 w_2

10 . t - $(\mathrm{O}\, w_1)\mathcal{U}(\mathrm{O}\, w_2) \equiv \mathrm{O}(w_1 \mathcal{U} w_2)$ by 4, 9 and PR ■

Having classified □ as a universal operator, 0 as an existential operator and 0 as being both universal and existential, we observe that U is universal with respect to its first argument and existential with respect to its second argument. This yields the commutation properties listed in T29 and T30.

T29. $\vdash (w_1 \text{ A } w_2)\mathcal{U} w_3 \equiv [w_1\, \mathcal{U} w_3 \text{ A } w_2 \mathcal{U} w_3]$

Proof:

1. $\vdash$ $(w_1 \wedge w_2) \supset w_1$ by PT

2. $\vdash (w_1 \text{ A } w_2)\mathcal{U} w_3 \supset w_1 \mathcal{U} w_3$ by UU

3. $\vdash (w_1 \text{ A } w_2)\mathcal{U} w_3 \ \supset\ w_2 \mathcal{U} w_3$ similarly

4. $\vdash (w_1 \text{ A } w_2)\mathcal{U} w_3 \ \supset\ [w_1 \mathcal{U} w_3 \text{ A } w_2 \mathcal{U} w_3]$ by 2, 3 and PR

5. $\vdash$ $w_1 \mathcal{U} w_3 \supset \diamond w_3$ by A10

6. I - $[w_1 \mathcal{U} w_3 \text{ A } w_2 \mathcal{U} w_3] \supset \diamond\, w_3$ by PR

7. $\vdash$ $w_1\, \mathcal{U} w_3 \supset \{w_3 \vee [w_1 \wedge \mathrm{O}(w_1 \mathcal{U} w_3)]\}$ by A9 and 1'11

8. $\vdash w_2 \mathcal{U} w_3 \supset \{w_3 \vee [w_2 \text{ A } \mathrm{O}(w_2 \mathcal{U} w_3)]\}$ by A9 and PR

9. I - $[w_1 \mathcal{U} w_3 \text{ A } w_2 \mathcal{U} w_3] \supset \{w_3 \vee [(w_1 \text{ A } w_2) \text{ A } \mathrm{O}(w_1 \mathcal{U} w_3 \text{ A } w_2 \mathcal{U} w_3)]\}$
by 7, 8, T12 and PR

10. $\vdash [w_1 \mathcal{U} w_3 \text{ A } w_2 \mathcal{U} w_3] \supset (w_1 \text{ A } w_2)\mathcal{U} w_3$ by 6, 9 and RUI,
taking w to be $(w_1\, \mathcal{U} w_3) \text{ A } (w_2 \mathcal{U} w_3)$, u to bc $w_1 \text{ A } w_2$, and v to bc w_3

11. $\vdash (w_1 \text{ A } w_2)\mathcal{U}w_3 \equiv [w_1\mathcal{U}w_3 \text{ A } w_2\mathcal{U}w_3]$ by 4, LO and PR ∎

T30. t- $w_1\mathcal{U}(w_2 \vee w_3) \equiv [w_1\mathcal{U}w_2 \text{ v } w_1\mathcal{U}w_3]$

Proof:

1. $\vdash w_2 \supset (w_2 \vee w_3)$ by PT

2. $\vdash w_1\mathcal{U}w_2 \supset w_1\mathcal{U}(w_2 \vee w_3)$ by UU

3. $\vdash w_1\mathcal{U}w_3 \supset w_1\mathcal{U}(w_2 \text{ v } w_3)$ similarly

4. $\vdash [w_1\mathcal{U}w_2 \text{ v } w_1\mathcal{U}w_3] \supset w_1\mathcal{U}(w_2$ v w 3) by 2, 3 and PR

5. I- $w_1\mathcal{U}(w_2$ v w3) $\supset \{(w_2 \vee w_3) \vee [w_1 \wedge \bigcirc(w_1\mathcal{U}(w_2 \vee w_3))]\}$ by A9 and PR

6. $\vdash [w_2 \vee (w_1 \text{ A } \bigcirc(w_1\mathcal{U}w_2))] \supset w_1\mathcal{U}w_2$ by A9 and PR

7. I- $\sim(w_1\mathcal{U}w_2) \supset \{\sim w_2 \text{ A } [\sim w_1 \vee \bigcirc\sim(w_1\mathcal{U}w_2)]\}$ by A4 and PR

8 . $\vdash \sim(w_1\mathcal{U}w_3) \supset \{\sim w_3 \text{ A } [\sim w_1 \vee \bigcirc\sim(w_1\mathcal{U}w_3)]\}$ similarly

9. $\vdash [w_1\mathcal{U}(w_2 \vee w_3) \text{ A } \sim(w_1\mathcal{U}w_2) \text{ A } \sim(w_1\mathcal{U}w_3)] \supset$
$[\sim w_2 \text{ A } \sim w_3 \text{ A } w_1 \text{ A } \bigcirc(w_1\mathcal{U}(w_2 \vee w_3)) \text{ A } \bigcirc\sim(w_1\mathcal{U}w_2) \text{ A } \bigcirc\sim(w_1\mathcal{U}w_3)]$
by 5, 7, 8 and PR

10. $\vdash [w_1\mathcal{U}(w_2 \vee w_3) \text{ A } \sim(w_1\mathcal{U}w_2) \text{ A } \sim(w_1\mathcal{U}w_3)] \supset$
$\{\sim(w_2 \vee w_3) \text{ A } \bigcirc[w_1\mathcal{U}(w_2 \vee w_3) \text{ A } \sim(w_1\mathcal{U}w_2) \text{ A } \sim(w_1\mathcal{U}w_3)]\}$
by T12 and PR

ft. t- $[w_1\mathcal{U}(w_2 \vee w_3) \text{ A } \sim(w_1\mathcal{U}w_2) \text{ A } \sim(w_1\mathcal{U}w_3)] \supset \Box\sim(w_2 \vee w_3)$ by DCI

12. $\vdash w_1\mathcal{U}(w_2$ v w 3) $\supset \Diamond(w_2 \text{ v } w_3)$ by A10

13. $\vdash w_1\mathcal{U}(w_2 \vee w_3) \supset \sim[\sim(w_1\mathcal{U}w_2) \text{ A } \sim(w_1\mathcal{U}w_3)]$ by 11, 12, Al and PR

14. I - $w_1\mathcal{U}(w_2 \text{ v } w_3) \supset [w_1\mathcal{U}w_2 \text{ v } w_1\mathcal{U}w_3]$ by PR

15. $\vdash w_1\mathcal{U}(w_2 \text{ v } w_3) \equiv [w_1\mathcal{U}w_2 \text{ v } w_1\mathcal{U}w_3]$ by 4, 14 and PR ∎

T31. $\vdash [\Diamond w_1 \vee \Diamond w_2] \supset [(\sim w_1)\mathcal{U}w_2 \vee (\sim w_2)\mathcal{U}w_1]$

Proof:

1. $\vdash [\Diamond w_1 \vee \Diamond w_2] \supset \Diamond(w_1 \text{ v } w_2)$ by T8 and PR

2. $\vdash \Diamond(w_1 \vee w_2) \supset (\sim(w_1 \text{ V } w_2))\mathfrak{U}(w_1 \text{V } w_2)$ by T23 and PR

3. t- $\Diamond(w_1 \vee w_2) \supset (\sim w_1 \text{ A } \sim w_2)\mathfrak{U}(w_1 \vee w_2)$ by UU and PR

4. $\vdash \Diamond(w_1 \vee w_2)$ 3 $[(\sim w_1 \text{ A } \sim w_2)\mathfrak{U}w_1 \vee (\sim w_1 \text{ A } \sim w_2)\mathfrak{U}w_2]$ by T30 and PR

5. $\vdash (\sim w_1 \text{ A } \sim w_2)\mathfrak{U}w_1 \supset (\sim w_2)\mathfrak{U}w_1$ by $\mathfrak{U}\mathfrak{U}$ and PR

6. $\vdash (\sim w_1 \text{ A } \sim w_2)\mathfrak{U}w_2 \supset (\sim w_1)\mathfrak{U}w_2$ by UU and PR

7. $\vdash \Diamond(w_1 \vee w_2) \supset [(\sim w_1)\mathfrak{U}w_2 \vee (\sim w_2)\mathfrak{U}w_1]$ by 4, 5, 6 and PR

8. $\vdash (\Diamond w_1 \vee 0 \quad w_2) \supset [(\sim w_1)\mathfrak{U}w_2 \text{ v } (\sim w_2)\mathfrak{U}w_1]$ by 1, 7 and PR ∎

The following two theorems display the one way implication resulting frorn the interchange of the U with a boolean operator of the opposite character.

1'32. t- $w_1\mathfrak{U}(w_2 \wedge w_3) \supset [w_1\mathfrak{U}w_2 \text{ A } w_1\mathfrak{U}w_3]$

Proof:

1. $\vdash (w_2 \text{ A } w_3) \supset w_2$ by PT

2. $\vdash w_1\mathfrak{U}(w_2 \wedge w_3) \supset w_1\mathfrak{U}w_2$ by $\mathfrak{U}$ U and PR

3. $\vdash w_1\mathfrak{U}(w_2 \text{ A } w_3) \supset w_1\mathfrak{U}w_3$ similarly

4. $\vdash w_1\mathfrak{U}(w_2 \text{ A } \quad w_3) \supset [w_1\mathfrak{U}w_2 \text{ A } \quad w_1\mathfrak{U}w_3]$ by 2, 3 and PR 4

T33. $\vdash [w_1\mathfrak{U}w_3 \vee \quad w_2\mathfrak{U}w_3] \supset (w_1 \vee \quad w_2)\mathfrak{U}w_3$

Proof:

1. $\vdash w_1 \supset (w_1 \vee w_2)$ by IT

2. $\vdash w_1\mathfrak{U}w_3 \supset (w_1 \vee w_2)\mathfrak{U}_3$ by UU

3. I- $w_2 \supset (w_1 \text{ v } w_2)$ by PT

4. $\vdash w_2\mathfrak{U}w_3 \supset (w_1 \vee w_2)\mathfrak{U}_3$ by UU

5. $\vdash [w_1\mathfrak{U}w_3 \vee w_2\mathfrak{U}w_3] \supset (w_1 \vee w_2)\mathfrak{U}w_3$ by 2, 4 and PR ∎

T34. $\vdash (w_1 \supset w_2)\mathfrak{U}w_3 \supset [w_1\mathfrak{U}w_3 \supset w_2\mathfrak{U}w_3]$

Proof:

1. $\vdash (w_1 \supset w_2)\mathcal{U}w_3 \supset \Diamond w_3$ by A10

2. $\vdash [(w_1 \supset w_2)\mathcal{U}w_3 \text{ A } w_1\mathcal{U}w_3] \supset$

$\{w_3 \vee [(w_1 \supset w_2) \wedge \bigcirc((w_1 \supset w_2)\mathcal{U}w_3) \wedge w_1 \wedge \bigcirc(w_1\mathcal{U}w_3)]\}$ by A9 and PR

3. $\vdash [(w_1 \supset w_2)\mathcal{U}w_3 \text{ A } w_1\mathcal{U}w_3] \supset$

$\{w_3 \vee [w_2 \wedge \bigcirc((w_1 \supset w_2)\mathcal{U}w_3) \wedge \bigcirc(w_1\mathcal{U}w_3)]\}$ by PR

4. $\vdash [(w_1 \supset w_2)\mathcal{U}w_3 \text{ A } w_1\mathcal{U}w_3] \supset$

$\{w_3 \vee [w_2 \wedge \bigcirc((w_1 \supset w_2)\mathcal{U}w_3 \wedge w_1\mathcal{U}w_3)]\}$ by T12 and PR

5. $\vdash [(w_1 \supset w_2)\mathcal{U}w_3 \text{ A } w_1\mathcal{U}w_3] \supset w_2\mathcal{U}w_3$ by 1, 4 and RUI, taking w to ho $((w_1 \supset w_2)\mathcal{U}w_3) \wedge (w_1\mathcal{U}w_3)$, u to be w_2, and v to be w_3

6. $\vdash (w_1 \supset w_2)\mathcal{U}w_3 \supset [w_1\mathcal{U}w_3 \supset w_2\mathcal{U}w_3]$ by PR ∎

T35. $\vdash [w_1\, \mathcal{U}w_2 \text{ A } (\sim w_2)\mathcal{U}w_3] \supset w_1\, \mathcal{U}w_3$

Proof:

1 . $\vdash (\sim w_2)\mathcal{U}w_3 \; 3 \; \mathbf{0} \; w_3$ by A10

2. $\vdash [w_1\mathcal{U}w_2 \text{ A } (\sim w_2)\mathcal{U}w_3] \supset \Diamond w_3$ by PR

3. $\vdash w_1\mathcal{U}w_2 \; \mathbf{3} \; \{w_2 \text{ v } [w_1 \text{ A } \bigcirc(w_1\mathcal{U}w_2)]\}$ by A9 and PR

4. $\vdash (\sim w_2)\mathcal{U}w_3 \supset \{w_3 \text{ v } [\sim w_2 \text{ A } \bigcirc((\sim w_2)\mathcal{U}w_3)]\}$ by A9 and PR

5. $\vdash [w_1\mathcal{U}w_2 \text{ A } (\sim w_2)\mathcal{U}w_3] \supset$

$\{w_3 \vee [w_1 \wedge \sim w_2 \text{ A } \bigcirc(w_1\mathcal{U}w_2) \text{ A } \bigcirc((\sim w_2)\mathcal{U}w_3)]\}$ by 3, 4 and PR

6. $\vdash [w_1\mathcal{U}w_2 \text{ A } (\sim w_2)\mathcal{U}w_3] \supset$

$\{w_3 \vee [w_1 \wedge \bigcirc(w_1\mathcal{U}w_2 \wedge (\sim w_2)\mathcal{U}w_3)]\}$ by T12 and PR

7. $\vdash [w_1\mathcal{U}w_2 \text{ A } (\sim w_2)\mathcal{U}w_3] \supset w_1\mathcal{U}w_3$ by 2, 6 and RUI ∎

T36. $\vdash w_1\mathcal{U}(w_2 \text{ A } w_3) \supset (w_1\mathcal{U}w_2)\mathcal{U}w_3$

Proof:

1. $\vdash w_1\mathcal{U}(w_2 \text{ A } w_3) \supset \Diamond(w_2 \text{ A } w_3)$ by A10

2. $\vdash (w_2 \wedge w_3) \supset w_3$ by PT

3. $\vdash \Diamond(w_2 \ \mathrm{A} \ w_3) \supset \Diamond w_3$ by 00

4. I- $w_1 \mathcal{U}(w_2 \ \mathrm{A} \ w_3) \supset \Diamond w_3$ by 1, 3 and PR

5. t- $w_1 \mathcal{U}(w_2 \wedge w_3) \supset \{(w_2 \wedge w_3) \vee [w_1 \wedge \mathrm{O}(w_1 \mathcal{U}(w_2 \wedge w_3))]\}$ by A9 and PR

6. $\vdash (w_2 \ \mathrm{A} \ w_3) \supset w_2$ by PT

7. $\vdash w_1 \mathcal{U}(w_2 \wedge w_3) \supset w_1 \mathcal{U} w_2$ by UU

8. $\vdash w_1 \mathcal{U}(w_2 \ \mathrm{A} \ w_3) \supset \{w_3 \vee [w_1 \mathcal{U} w_2 \ \mathrm{A} \ \mathrm{O}(w_1 \mathcal{U}(w_2 \wedge w_3))]\}$ by 5, 7 and PR

9. $\vdash w_1 \mathcal{U}(w_2 \ \mathrm{A} \ w_3) \ 3 \ (w_1 \mathcal{U} w_2) \mathcal{U} w_3$ by 4, 8 and RUI ∎

The following two theorems are referred to as "collapsing" theorems, since they may be used to derive a consequence of smaller nesting depth from a nested until expression.

T37. $\vdash (w_1 \mathcal{U} w_2) \mathcal{U} w_3 \supset (w_1 \vee w_2) \mathcal{U} w_3$

Proof:

1. $\vdash w_1 \mathcal{U} w_2 \supset [w_2 \vee (w_1 \ \mathrm{A} \ \mathrm{O}(w_1 \mathcal{U} w_2))]$ by A9 and PR

2. $\vdash w_1 \mathcal{U} w_2 \supset (w_1 \vee w_2)$ by PR

3. $\vdash (w_1 \mathcal{U} w_2) \mathcal{U} w_3 \supset (w_1 \vee w_2) \mathcal{U} w_3$ by UU ∎

T38. $\vdash w_1 \mathcal{U}(w_2 \mathcal{U} w_3) \supset (w_1 \vee w_2) \mathcal{U} w_3$

Proof:

1. $\vdash w_1 \mathcal{U}(w_2 \mathcal{U} w_3) \ 3 \ \Diamond(w_2 \mathcal{U} w_3)$ by Al0

2. $\vdash w_2 \mathcal{U} w_3 \supset \Theta \ w_3$ by A10

3. $\vdash w_1 \mathcal{U}(w_2 \mathcal{U} w_3) \supset \Theta \ w_3$ by 1, 2 and ◇C

4. t- $w_1 \mathcal{U}(w_2 \mathcal{U} w_3) \supset \{w_2 \mathcal{U} w_3 \vee [w_1 \ \mathrm{A} \ \mathrm{O}(w_1 \mathcal{U}(w_2 \mathcal{U} w_3))]\}$ by A9 and PR

5. $\vdash w_1 \mathcal{U}(w_2 \mathcal{U} w_3) \supset \{w_3 \vee [w_2 \ \mathrm{A} \ \mathrm{O}(w_2 \mathcal{U} w_3)] \vee [w_1 \ \mathrm{A} \ \mathrm{O}(w_1 \mathcal{U}(w_2 \mathcal{U} w_3))]\}$ by A9 and PR

6. $\vdash w_2 \mathcal{U} w_3 \supset w_1 \mathcal{U}(w_2 \mathcal{U} w_3)$ by A9 and PR

7. I - $[w_2 \ \mathrm{A} \ \mathrm{O}(w_2 \mathcal{U} w_3)] \ 3 \ [(w_1 \vee w_2) \ \mathrm{A} \ \mathrm{O}(w_1 \mathcal{U}(w_2 \mathcal{U} w_3))]$ by 00 and PR

8. t - $[w_1$ A $\bigcirc(w_1\mathcal{U}(w_2\mathcal{U}w_3))] \supset [(w_1 \vee w_2) \wedge \bigcirc(w_1\mathcal{U}(w_2\mathcal{U}w_3))]$ by PR

9 . I - $w_1\mathcal{U}(w_2\mathcal{U}w_3) \supset \{w_3 \vee [(w_1 \vee w_2)$ A $\bigcirc(w_1\mathcal{U}(w_2\mathcal{U}w_3))]\}$ by 5, 7, 8 and PR

10. t - $w_1\mathcal{U}(w_2\mathcal{U}w_3) \supset (w_1 \vee w_2)\mathcal{U}w_3$ by 3, 9, and RUI ∎

A very useful derived operator is the unless operator u $\mathfrak{U}$ v being defined by

$$u\,\mathfrak{U}\,v \;\equiv\; [\square u \;\; \vee \;\; (u\mathcal{U}v)].$$

The unless operator does not insist on the fact that v actually happens but it requires that u holds until such an occurrence. If v never happens u must hold forever. This operator is related to the binary "as long as" operator $p \;\square\; q$, reading "q as long as p," introduced by Lamport in [L2]. The meaning of this construct is that q holds continuously as long as p is continuously maintained. We may express $p\,\square\,q$ by:

$$p\,\square\,q \;\equiv\; q\,\mathfrak{U}(\sim p).$$

Following is a rule for establishing the unless operator.

Unless Introduction — $\mathfrak{U}$I

$$\frac{\vdash u \supset \bigcirc(u \vee v)}{\vdash u \supset (u\,\mathfrak{U}\,v)}$$

Proof:

1. $\vdash$ u $\supset \bigcirc(u$ v $v)$ given

2. $\vdash u \supset [\bigcirc u \vee \bigcirc v]$ by T13

3 . $\vdash \sim(u\mathcal{U}v) \supset \{\sim v$ A $[\sim u \vee \bigcirc\sim(u\mathcal{U}v)]\}$ by A9, T4 and PR

4. I - 0 $\sim(u\mathcal{U}v) \supset$ o - v by 0 0 and PR

5. $\vdash$ [u A $\sim(u\mathcal{U}v)] \supset$ [u A $\bigcirc\sim(u\mathcal{U}v)]$ by 3 and PR

6. $\vdash$ [u A $\sim(u\mathcal{U}v)] \supset$ [u A $\bigcirc\sim(u\mathcal{U}v)$ A $\sim\bigcirc v]$ by 4, 5, A4 and PR

7. $\vdash$ [u A $\sim(u\mathcal{U}v)] \supset [u$ A $\bigcirc u$ A $\bigcirc\sim(u\mathcal{U}v)]$ by 2, 6 and PR

8. $\vdash$ [u A $\sim(u\mathcal{U}v)] \supset [u$ A $\bigcirc(u$ A $\sim(u\mathcal{U}v))]$ by T7 and PR

9. $\vdash$ [u A $\sim(u\mathcal{U}v)] \supset \square$ u by DCI

L0. I- u $\supset (\square$ u v $(u\mathcal{U}v))$ by PR

11. t- u $\supset (u \,\mathfrak{U}\, v)$ by definition of $\mathfrak{U}$ ■

This concludes the description of the propositional section of general temporal logic. The axiomatic system presented for this section of the logic is known to be complete, and the validity problem decidable ([I'S]). Consequently, there exists a procedure that tests each formula in PTL (Propositional Temporal Logic) for validity, and constructs a proof in the presented system if the statement is valid. The procedure given in [PS] takes exponential time in the size of the tested formula.

4. QUANTIFIERS

Since we intend to use terms and predicates in our reasoning we have to extend our system to admit individual variables, terms and quantification. Let us consider additional axioms involving quantifiers and their interaction with the temporal operators.

AXIOMS:

A11. $\vdash$ -32.w $\equiv \forall x. \sim w$

A12. $\vdash (\forall x.w(x)) \supset$ w (t)
where t is any term globally free for x in w

A13. $\vdash$ (V x . $\bigcirc w) \supset (\bigcirc \forall x.w)$

In these axioms, x is any global individual variable. Axioms **Al 1** and A **12** are the usual predicate calculus axioms: *A I 1* defines *3 as* the dual of **V** and A12 is the *instantiation axiom.* Axiom **Al3** is the Barcan formula for the 0 operator; it states that since both operators $\forall$ and *0* have universal characteristics they commute. **We** use the substitution notation $w(x)$ replaced by $w(t)$ to denote the substitution of the term t for all free occurrences of x in w.

A- term t is said to be globally free for x in w if substitution of t for all free occurrences of x in w: (a) does not create new bound occurrences of (global) variables, and (b) does not, create new occurrences of local variables in the scope of a temporal operator. A trivial case: if t is x itself, then t is free for x. Condition (a) is the one stipulated in classical predicate logic. Condition (b) is special to modal and temporal logics with quantification. Condition (b) is essential for A12, because without it we could derive the formula

$$(\forall x.\ \Diamond(x < y)) \supset \Diamond(y < y),$$

which is not valid for a local variable y.

An additional rule of inference is:

INFERENCE RULE:

R4. ∀ Insertion --- $\forall I$

$$\frac{\vdash u \supset v}{\vdash u \supset \forall x.v}$$

where x is not free in u.

DERIVED RULES AND THEOREMS:

From R4 we can obtain the derived rule

Instantiation Rule -- INST

$\vdash w(x)$

- -

I- $w(t)$

where t is any term globally free for x in w.

Proof:

1. $\vdash w(x)$ gi vcn
2. t- $\forall x.w(x)$ by ∀I (taking u to bc ***true***)
3. I - $(\forall x.w(x)) \supset w(t)$ by A12
4. $\vdash w(t)$ by 2, 3 and MI' ∎

The following arc thc duals of A 12 and R4 for thc existential quantifier 3:

T39. $\vdash w(t) \supset \exists x.w(x)$
where t is any term globally free for x in w.

Proof:

1. t- (vx. $\sim w(x)) \supset \sim w(t)$ by A12
2. $\vdash (\sim\exists x.w(x)) \supset \sim w(t)$ by A11 and PR
3. t- w(t) $\supset \exists x.w(x)$ by I'R ∎

Notc again that wc need here thc additional condition (b) ensuring that Lhc substitution of t for x in w does not, create new occurrences of local variables in thc scope of a modal operator.

3 *Insertion* -- ∃I

$$\frac{l\text{-}u \supset v}{t\text{-}\ 3x.u \supset v}$$

where x is not frec in v

Proof:

1. $\vdash u \supset v$ given
2. $\vdash \sim v \supset \sim u$ by PR
3. $\vdash \sim v \supset \forall x. \sim u$ by ∀I
4. $\vdash \sim v \supset \sim \exists x.u$ by A11 and PR
5. $\vdash \exists x.u \supset v$ by PR J

∀∀ *Rules*

(b) (a) $\dfrac{\vdash u \supset v}{\vdash \forall x.u \supset \forall x.v}$ $\dfrac{\vdash u \equiv v}{\vdash \forall x.u \equiv \forall x.v}$

Proof of (a):

1. $\vdash \forall x.u \supset u$ by A12
2. l-u $\supset$ v given
3. I- $\forall x.u \supset$ v by PR
4. I - $\forall x.u \supset \forall x.v$ by VI, since $\forall x.u$ contains no frec occurrences of x.

Rule (b) then follows by propositional reasoning. J

∃∃ *Rules*

(a) $\dfrac{t\text{-}u\ \ 3\ \ v}{\vdash \exists x.u \supset 3x.v}$ (b) $\dfrac{\vdash u \equiv v}{\vdash 3x.u \equiv \exists x.v}$

Proof of (a):

1. l-u $\supset$ v given
2. t- $(\sim v) \supset (\sim u)$ by PR
3. t- $(\forall x. \sim v) \supset (\forall x. \sim$ **u)** by ∀∀
4. $\vdash$ **(4x.2))** $\supset (\sim \exists x.u)$ by A 1 1 and PR

5. I- $\exists x.u \supset$ 3x.v   by PR

Rule (b) then fd1ows by propositional reasoning. ▟

From the axiom Al,

$\vdash \sim\Diamond w \equiv$ o - w ,

we can clearly deduce the formula

$\vdash \sim(w \text{ v } \Box\sim w) \equiv \sim(w \text{ v } \sim\Diamond w)$

by propositional reasoning (PR). However, we cannot deduce by PR the formula

$\Box\Box\sim w \equiv$ c l - o w

or

$\forall x.\ \Box\sim w \equiv \forall x.\ \sim$ o w .

Here, the replacement of $\Box \sim w$ by $\sim$ 0 w is under the scope of the operator $\Box$ and the quantifier $\forall x$, respectively, and thus cannot be justified by propositional reasoning alone. For this reason we need the following equivalence rule.

***Equivalence Rule* ---- ER**

Let w' be the result of replacing an occurrence of a subformula v_1 in w by v_2. Then

$$\frac{\vdash v_1 \equiv v_2}{t\text{-}\ w \equiv w'}$$

Proof:

By induction on the structure of w.

Case: w is v_1. Then w' is v_2 and t- $v_1 \equiv v_2$ implies $\vdash$ w $\equiv$ w'.

Cuse: w is of Lhc form $\sim u$. WC assume that t- $v_1 \equiv v_2$ implies $\vdash$ u $\equiv$ u'. Then by propositional reasoning $\vdash \sim u \equiv \sim u'$, i.e., I- w $\equiv$ w'.

Case: w is of the form $u_1 \vee u_2$. WC assume that if I- $v_1 \equiv v_2$, then t- $u_1 \equiv u'_1$ and t- $u_2 \equiv u'_2$ Then by propositional reasoning $\vdash (u_1 \vee u_2) \equiv (u'_1 \vee u'_2)$, i.e., $\vdash$ w $\equiv$ w'.

. The cases where w is of forms $u_1 \wedge u_2$, $u_1 \supset u_2$, etc. are similar.

Case: w is of the form ❑ lu. We assume that if I- $v_1 \equiv v_2$, lhen $\vdash$ u $\equiv$ u'. By the ❑ Cl-rule, $\vdash$ ❑ u $\equiv \Box$ u', i.c., t- w $\equiv w'$.

Thc cascs in which w is of forms 0 u, 0 u, and $u_1 \mathcal{U} u_2$ are treated similarly, using the 0 ◇-rule, the 0 0-rule, and the UU-rule, respcctivcly.

Case: w is of the form $\forall x.u$. We assume that if t- $v_1 \equiv v_2$, then t- u ≡ u'. Thcn by the ∀∀-rule, t- $\forall x.u \equiv \forall x.u'$, i.e., ⊢ w ≡ w'.

Thc case where w is of form $\exists x.u$ is proved similarly by thc ∃∃-rulc. ■

Deduction Rule -- DED

$$\frac{w_1 \vdash w_2}{\vdash (\Box w_1) \supset w_2}$$

where thc ∀I rule (Rule R4) is never applied to a free variable of w_1 in the derivation of $w_1 \vdash w_2$.

That is, if under the assumption w_1 wc can derive ⊢ w_2, where rule R4 is never applied to a free variable of w_1 , then thcre exists a proof establishing ⊢ (Cl w_1) ⊃ w_2. Wc clearly must also be carcful in using any theorem or derived rulc such as thc ∀∀ or ER rulc Lhat was established using the VI rule.

The additional □ operator in the conclusion is obviously necessary sincc in gcncral w_1 I- w_2 does not imply t- $w_1 \supset w_2$. For examplc, obviously w ⊢ Cl w is truc (an immediate application of rulc R3: t- w by assumption and therefore ⊢ □ w by □I); but w ⊃ ❑ w is not, a theorem.

Proof:

The proof of thc temporal Deduction Rule follows the same arguments uscd in the proof of thc classical deduction Lhcorcm of Predicate Calculus. By the given $w_1 \vdash w_2$, there exists a proof of the form:

$\vdash u_1$
$\vdash u_2$

.

$\vdash u_m$

such that $u_1 = w_1$ is thc hypothesis on which the proof relics, and $u_m = w_2$ is the conscyucnce of Lhc proof. Wc rcplacc each line $\vdash u_i$ in thc proof of $w_1 \vdash w_2$ by thc line I- ❑ lzol ⊃ u;, and show that Lhis transformation prcscrvcs soundness. That is

given	show
$\vdash u_1$	$\vdash (\Box w_1) \supset u_1$
$\vdash u_2$	$\vdash (\Box w_1) \supset u_2$
.	.
.	.
.	.

t- u_i t- (cl $w_1) \supset u_i$

. .

. .

. .

$\vdash u_m$ $\vdash (\Box w_1) \supset u_m$

i.e., I- w_2 i.e.$\vdash (\Box w_1) \supset w_2$

where each u_i is either the assumption w_1, an axiom, or derived from previous u_j's by some rule of inference.

The proof is by a complete induction on i. We assume that for all $k < i$, $\vdash (\Box w_1) \supset u_k$, and prove that I- $(\Box w_1) \supset u_i$.

Case: u_i is an axiom.

1. $\vdash u_i$ axiom
2. $\vdash (\Box w_1) \supset u_i$ by PR

Note that $\vdash$ w' implies $\vdash$ w $\supset$ w' for any w, by propositional reasoning.

Case: u_i is w_1.

1. $\vdash (\Box w_1) \supset w_1$ by A3

Case: u_i is obtained by rule R1, i.e., u_i is an instance of a tautology.

1. $\vdash u_i$ by PT
2. $\vdash (\Box w_1) \supset u_i$ by PR

Case: u_i is obtained by rule R2 (using previous $\vdash u_k$ and $\vdash u_k$ 3 u_i).

1. $\vdash (\Box w_1) \supset u_k$ induction hypothesis
2. $\vdash (\Box w_1) \supset (u_k \supset u_i)$ induction hypothesis
3. $\vdash (\Box w_1) \supset u_i$ by 1, 2 and PR

Case: u_i is obtained by rule R3 (using previous $\vdash u_k$), i.e., u_i is $\Box u_k$.

1. $\vdash (\Box w_1) \supset u_k$ induction hypothesis
2. t - $(\Box\Box w_1) \supset \Box u_k$ by ❑ ICI
3. ♦📫 $(\Box w_1) \supset$❑●⚐⌽● by T3 and PR
4. $\vdash (\Box w_1) \supset$❑♦& by 2, 3 and PR

Case: u_i is obtained by rulc R4 (using previous $\vdash u \supset v$, i.e. u_k, Lo get $\vdash$ u $\supset \forall x.v$, i.e. u_i, where x is not frcc in ***u*).**

By our deduction rule assumption, we know that x is also not frcc in $\boldsymbol{w_1}$.

1. $\vdash (\Box w_1) \supset (u \supset$ v) — induction hypothesis

2. $\vdash ((\Box w_1)$ **A u)** $\supset$ **v** — by PR

3. **I-** $((\Box\ w_1)$ **A u)** $\supset \forall x.v$ — by R4 (since x is not frcc in u or w_1)

4. $\vdash (\Box w_1) \supset (u \supset \forall x.v)$ — by PR ∎

A different approach to coping with the application of the Cl insertion rule (rule R3) is Lo forbid it altogether. We then get the following restricted deduction rule:

Restricted Deduction Rule **--** RDED

$$\frac{w_1 \vdash w_2}{\text{I-}\ w_1 \supset w_2}$$

wherc □I (rule R3) is ncver applied and VI (rule R4) is ncver applied to a free variable of w_1 in the derivation of $w_1 \vdash w_2$.

Ilerc, wc arc not allowcd to use rule □I or any theorem or derived rulc in whose proof □I was used.

The proof' of RDED follows exactly that of DED cxccpt that Lhc casc in which rule R3 is applied does not arise.

QUANTIFIER THEOREMS:

T40. I- (4x.w) $\equiv$ **(3x.** $\sim w)$

Proof:

1. 1- $(\sim\sim w) \equiv$ w — **by** PT

2. t- $(\forall x. \sim\sim w) \equiv \forall x.w$ — **by** $\forall\forall$

3. $\vdash (\sim\exists x. \sim$**w)** $\equiv$ **vx.w** — **by Al 1** and PR

4. **t-** $\sim\forall x.w \equiv$ **3x.** $\sim w$ — by PR ∎

T41. $\vdash \forall x.(w_1$ A $w_2) \equiv (\forall x.w_1$ A $\forall x.w_2)$

Proof:

1. t- $\forall x.w_1$ 3 w_1 — by A12
2. $\vdash \forall x.w_2$ 3 w_2 — by Al2
3. I - $(\forall x.w_1$ A $\forall x.w_2) \supset (w_1$ A $w_2)$ — by 1, 2 and PR
4. $\vdash (\forall x.w_1$ A $\forall x.w_2) \supset \forall x.(w_1$ A $w_2)$ — by $\forall$I
5. $\vdash (w_1 \wedge w_2) \supset w_1$ — by PT
6. t- $\forall x.(w_1$ A $w_2) \supset \forall x.w_1$ — by $\forall\forall$
7. $\vdash (w_1$ A $w_2) \supset w_2$ — by PT
8. $\vdash \forall x.(w_1$ A $w_2) \supset \forall x.w_2$ — by $\forall\forall$
9. $\vdash \forall x.(w_1$ A $w_2) \supset (\forall x.w_1$ A $\forall x.w_2)$ — by 6, 8 and PR
10. $\vdash \forall x.(w_1$ A $w_2) \equiv (\forall x.w_1$ A $\forall x.w_2)$ — by 4, 9 and PR ∎

T42. $\vdash \exists x.(w_1$ v $w_2) \equiv (\exists x.w_1$ v $\exists x.w_2)$

Proof:

1. $\vdash \forall x.(\sim w_1$ A $\sim w_2) \equiv (\forall x. \sim w_1$ A $\forall x. \sim w_2)$ — by T41
2. I- vx. $\sim (w_1$ V $w_2) \equiv (\forall x. \sim w_1$ A $\forall x. \sim w_2)$ — by ER
3. $\vdash \sim\exists x.(w_1$ V $w_2) \equiv (\sim\exists x.w_1$ A $\sim\exists x.w_2)$ — by Al l and PR
4. I - $\exists x.(w_1$ v $w_2) \equiv (\exists x.w_1$ v $\exists x.w_2)$ — by PR ∎

T43. $\vdash \forall x.(w_1$ V $w_2) \equiv [w_1$ V $\forall x.w_2]$ where x is not free in w_1.

Proof:

1. t- $\forall x.(w_1$ v $w_2) \supset [w_1$ v $w_2]$ — by Al2
2. $\vdash [\forall x.(w_1$ V $w_2)$ A $\sim w_1]$ 3 w_2 — by PR

3 . ⊢ $[\forall x.(w_1$ V $w_2)$ A $\sim w_1] \supset \forall x.w_2$ — by ∀I, since x is not free in $\forall x.(w_1$ V $w_2)$ A $\sim w_1$

4. ⊢ $\forall x.(w_1$ v $w_2) \supset [w_1$ v $\forall x.w_2]$ — by PR

5. ⊢ $w_1 \supset [w_1$ v $w_2]$ — by PT

6. ⊢ $\forall x.w_2 \supset w_2$ — by A12

7. I- $\forall x.w_2 \supset [w_1$ v $w_2]$ — by PR

8. t- $[w_1$ v $\forall x.w_2] \supset [w_1$ v $w_2]$ — by 5, 7 and PR

9. t - $[w_1$ v $\forall x.w_2] \supset \forall x.(w_1$ v $w_2)$ — by ∀I, since x is not free in w_1 V $\forall x.w_2$

10. t - $\forall x.(w_1$ v $w_2) \equiv [w_1$ v $\forall x.w_2]$ — by 4, 9 and PR ┛

T44. ⊢ $\exists x.(w_1$ A $w_2) \equiv [w_1$ A $\exists x.w_2]$ where x is not free in w_1

Proof: By duality on the previous theorem.

The following two theorems show that the 0 operator also commutcs with the quantifiers.

T45. t- (vx. 0 w) ≡ (0 $\forall x.w$)

Proof:

1. ⊢ $(\forall x.$ O$w) \supset$ **(0** $\forall x.w)$ — by A13

2. ⊢ v x . w ⊃ w — by A12

3. ⊢ (O$\forall x.w$) ⊃ o w — by *0 0*

4. ⊢ (0 $\forall x.w$) ⊃ ($\forall x.$ 0 w) — by VI

5. ⊢ ($\forall x.$ 0 w) ≡ (0 $\forall x.w$) — by 1, 4 and PR ┛

1'46. ⊢ (3x. Ow) ≡ (0 $\exists x.w$)

Proof:

1. ⊢ ($\forall x.$ **0** $\sim w$) ≡ **(0 vx. ~ w)** — by T45

2. $\vdash (\forall x. \sim 0\ w) \equiv (\ 0 \text{-}3x.w)$

3. t- $(\text{-}3x.\ \bigcirc w) \equiv (\sim 0\ 3x.w)$

4. I- $(3x.\ \bigcirc w) \equiv (\bigcirc \exists x.w)$

The following two theorems show that each temporal operator commutes with the quantifier that has similar character (universal, or existential).

T47. $\vdash$ (Vx. $\Box\ w) \equiv (\Box \forall x.w)$

Proof:

1. $\vdash \Box w \supset [\ w\ \ \text{A}\ \ \bigcirc \Box w]$ — by T20 and PR

2. $\vdash (\forall x. \Box w) \supset \forall x.(w\ \ \text{A}\ \ \bigcirc \Box w)$ — by VV

3. $\vdash (\forall x.\ \Box\ w) \supset [(\forall x.w)\ \text{A}\ (\forall x.\ \bigcirc\ \Box\ \text{w})]$ — by T41 and PR

4. $\vdash\ \ (\forall x.\ \ \Box\ \ \text{w})\supset[(\forall x.w)\text{A}(\bigcirc \forall x.\Box\ \ \text{w})]$ — by T45 and PR

5. t- $(\forall x.\ \Box\ \text{w}) \supset (\Box \forall x.w)$ — by DCI, taking u to be Vx. Cl w and v lo be $\forall x.w$

6. t- $(\forall x.w)\ \supset\ \text{w}$ — by A12

7. $\vdash (\Box \forall x.w)\ \ 3\ \ \Box w$ — by $\Box\Box$

8. $\vdash (\Box \forall x.w) \supset$ (Vx. Cl w) — by VI

9. $\vdash (\forall x.$ **Et w**$) \equiv (\Box \forall x.w)$ — by 5, 8 and PR ▟

T48. $\vdash$ (3x. Ow) $\equiv (\Diamond \exists x.w)$

Proof:

1. I- $(\forall x.\ \Box\ \sim w) \equiv (\Box$ vx. $\sim$ w) — by T47

2. t- $(\forall x. \sim 0\ \ w) \equiv (\Box\ \text{-}3x.\text{w})$ — by **Al, A** 11 and ER (twice)

3 . t - $(\sim\exists x. \Diamond w) \equiv (\sim \Diamond \exists x.w)$ — by **Al,** A1 1 and PR

4. $\vdash$ (3x. $\Diamond w) \equiv (\Diamond \exists x.w)$ — by PR ▟

Theorem T47 implics thc commutativity of V with Cl: Both have a universal character, with one quantifying over individuals and the other quantifying over states. Similarly, theorem T48 implies the commutativity of **3** with 0. The first two theorems (T45 **and 1'46)** imply the commutativity of V and **3** with 0.

The next two theorems arc consistent with the intcrprctation that the $\mathcal{U}$ operator is universal with respect to its first argument and existential with respect to the second.

T49. $\vdash$ $\forall x.(w_1 \mathcal{U} w_2) \equiv (\forall x.w_1)\mathcal{U} w_2$ where x is not free in w_2

Proof:

1. $\vdash w_1 \mathcal{U} w_2 \supset [w_2$ V $(w_1$ A $O(w_1 \mathcal{U} w_2))]$ — by A9 and PR

2. $\vdash \forall x.(w_1 \mathcal{U} w_2) \supset \forall x.[w_2$ V $(w_1$ A $O(w_1 \mathcal{U} w_2))]$ — by $\forall\forall$

3. $\vdash$ $\forall x.(w_1 \mathcal{U} w_2) \supset [w_2$ V $\forall x.(w_1$ A $O(w_1 \mathcal{U} w_2))]$ — by VI and PR,
since x is not frcc in w_2

4. t- $\forall x.(w_1 \mathcal{U} w_2) \supset [w_2$ V $(\forall x.w_1$ A $\forall x.$ $O(w$ $_1$ $\mathcal{U} w_2))]$ — by T41 and PR

5. t- $\forall x.(w_1 \mathcal{U} w_2) \supset [w_2$ V $(\forall x.w_1$ A **0** $\forall x.(w_1 \mathcal{U} w_2))]$ — by T45 and PR

6. **t-** $\forall x.(w_1 \mathcal{U} w_2) \supset \Diamond w_2$ — by A 12, A10 and PR

7. $\vdash \forall x.(w_1 \mathcal{U} w_2) \supset (\forall x.w_1)\mathcal{U} w_2$ — by 5, 6 and R$\mathcal{U}$I,
taking w to be $\forall x.(w_1 \mathcal{U} w_2)$, u to be $\forall x.w_1$, and v to be w_2

8. $\vdash (\forall x.w_1)$ **3** w_1 — by A12

9. $\vdash (\forall x.w_1)\mathcal{U} w_2 \supset w_1 \mathcal{U} w_2$ — by $\mathcal{U}\mathcal{U}$

10. $\vdash (\forall x.w_1)\mathcal{U} w_2 \supset \forall x.(w_1 \mathcal{U} w_2)$ — by $\forall$I,
since x is not frcc in w_2

11. $\vdash \forall x.(w_1 \mathcal{U} w_2) \equiv (\forall x.w_1)\mathcal{U} w_2$ — by 7, 10 and PR ∎

T50. · $\vdash \exists x.(w_1 \mathcal{U} w_2) \equiv w_1 \mathcal{U}(\exists x.w_2)$ where x is not free in w_1

Proof:

1. $\vdash w_1 \mathcal{U} w_2 \supset \Diamond w_2$ — by A10

2. t- $\exists x.(w_1 \mathcal{U} w_2) \supset (\exists x.\Diamond w_2)$ — by 33

3. t- $\exists x.(w_1 \mathcal{U} w_2) \supset (\Diamond \exists x.w_2)$ — by T48 and PR

4 . $\vdash w_1 \mathcal{U} w_2 \supset [w_2$ V $(w_1$ A $O(w_1 \mathcal{U} w_2))]$ — by **A9** and PR

5. $\vdash \exists x.(w_1 \mathcal{U} w_2) \supset [(\exists x.w_2)$ V $\exists x.(w_1$ A $O(w_1 \mathcal{U} w_2))]$ — by T42, 33 and PR

6. t - $\exists x.(w_1 \mathcal{U} w_2) \supset [(\exists x.w_2) \vee (w_1$ A $\exists x.\ O(w_1 \mathcal{U} w_2))]$ by T44 and PR, since x is not free in w_1

7. $\vdash \exists x.(w_1 \mathcal{U} w_2) \supset$ ((3 $x.w_2) \vee [w_1 \wedge O\, \exists x.(w_1 \mathcal{U} w_2)]\}$ by T46 and PR

8. t- $\exists x.(w_1 \mathcal{U} w_2) \supset w_1 \mathcal{U}(\exists x.w_2)$ by 3, 7, RUI and PR

9 . $\vdash [w_2 \vee (w_1$ A $O(w_1 \mathcal{U} w_2))] \supset w_1 \mathcal{U} w_2$ by A9 and PR

10. t - $\exists x.[w_2 \vee (w_1$ A $O(w_1 \mathcal{U} w_2))]$ 3 $\exists x.(w_1 \mathcal{U} w_2)$ by 33

11. t - $[(\exists x.w_2) \vee \exists x.(w_1$ A $O(w_1 \mathcal{U} w_2))] \supset \exists x.(w_1 \mathcal{U} w_2)$ by T42 and PR

12. $\vdash [(\exists x.w_2) \vee (w_1$ A 3 x . $O(w_1 \mathcal{U} w_2))] \supset \exists x.(w_1 \mathcal{U} w_2)$ by T44 and PR, since 2 is not free in w_1

13. t - $[(\exists x.w_2) \vee (w_1$ A $O\, \exists x.(w_1 \mathcal{U} w_2))] \supset \exists x.(w_1 \mathcal{U} w_2)$ by T46 and PR

14. I- $w_1 \mathcal{U}(\exists x.w_2) \supset \exists x.(w_1 \mathcal{U} w_2)$ by LUI, taking u to be w_1, v to bc $\exists x.w_2$ and w to bc $\exists x.(w_1 \mathcal{U} w_2)$

15. $\vdash \exists x.(w_1 \mathcal{U} w_2) \equiv w_1 \mathcal{U}(\exists x.w_2)$ by 8, 14 and PR ∎

While operators of similar character, i.e., both universal or both existential, commute to yield equivalent formulas, operators of' opposite character usually admit implication in one direction only. Thus we have:

T51. $\vdash \exists x.\Box w \supset \Box$ 3x.w

T52. t- $\Diamond \forall x.w \supset$ Vx. $\Diamond$ w

T53(a). $\vdash \exists x.(w_1 \mathcal{U} w_2) \supset (\exists x.w_1) \mathcal{U} w_2$ where x is not free in w_2

(b). $\vdash w_1\, \mathcal{U}(\forall x.w_2) \supset \forall x.(w_1 \mathcal{U} w_2)$ where x is not free in w_1

Theorems of similar character are:

T54(a). $\vdash \exists x.(u \mathcal{U} v) \supset (\exists x.u) \mathcal{U} (\exists x.v)$

(b). $\vdash (\forall x.u) \mathcal{U} (\forall x.v) \supset \forall x.(u \mathcal{U} v)$

THE NEXT OPERATOR APPLIED TO TERMS:

The use of the next operator 0 applied Lo terms is governed by the axioms:

A14. $\vdash$ 0 $f(t_1, \ldots, t_n) = f(\bigcirc\, t_1, \ldots, 0\, t_n)$
for any function f and terms $t_1, \ldots, t_n$

A15. t- $\bigcirc\, p(t_1, \ldots, t_n) \equiv p(\bigcirc\, t_1, \ldots, \bigcirc\, t_n)$
for any predicatc p and terms $t_1, \ldots, t_n$

These axioms are consistent with the evaluation rules that we gavc which statcd that in order to evaluate an expression 0 $\mathcal{E}(t_1,$ . . . , t,), we can evaluate &(O t_1, . . . , 0 t_n) whether $\mathcal{E}$ is a function or a predicate.

5. EQUALITY

Equality is handled by the following axioms:

AXIOMS:

A16. ***Reflexivity of Equality***

l - t = t for any term t

A17. ***Substitutivity of Equality***

$\vdash (t_1 = t_2) \supset [w(t_1, t_1) \equiv w(t_1, t_2)]$
where t_2 is any term globally free for t_1 in w
and where w does not contain temporal opcrators

A18. $\vdash \bigcirc(t_1 = t_2) \equiv (\bigcirc\, t_1 = \bigcirc\, t_2)$

We use $w(t_1, t_2)$ to indicate that t_2 replaces some of the occurrences of t_1 in w.

Thc axiom A18 is a special casc of Al5 when the predicate p is the equality predicate.

Recall that a tcrm t_2 is said to be ***globally*** free for t_1 in w if substitution of t_2 for all frcc occurrences of t_1 in w: (a) docs not create new bound occurrences of (global) variables, (i.e., t_2 is free *for* t_1 in w), and (b) docs not create new occurrences of local variablcs in the scope of a modal opcrator.

Note that the classical axiom for substitulivity of equality A 17

$$\vdash (t_1 = t_2) \supset [w(t_1, t_1) \equiv w(t_1, t_2)]$$

(where t_2 is free for t_1 in w) is not correct if w contains temporal operators. We could take $w(t_1, t_2)$ tbc□ ●☎⧫● = t_2) and dcducc from Al7

$$\vdash (t_1 = t_2) \supset [\square(t_1 = t_1)\ 3\ \square(t_1 = t_2)],$$

i.e.,

$\vdash (t_1$ ▮ $t_2) \supset$▯☎♦🗀 $= t_2)$,

which is not a valid statement (since $t_1 = t_2$ may contain local variables).

T55. ***Commutativity of Equality***

$$\vdash (t_1 = t_2)\ 3\ (t_2 = t_1)$$

Proof:

1. $\vdash (t_1 = t_2) \supset [(t_1 = t_1) \equiv (t_2 = t_1)]$ by Al7
2. $\vdash t_1 = t_1$ by Al6
3. $\vdash (t_1 = t_2) \supset (t_2 = t_1)$ by 1, 2 and PR ▟

T56. ***Transitivity of Equality***

$$\vdash [(t_1 = t_2)\ A\ (t_2 = t_3)] \supset (t_1 = t_3)$$

Proof:

1. $\vdash (t_1 = t_2) \supset [(t_1 = t_3) \equiv (t_2 = t_3)]$ by A17
2. $\vdash [(t_1 = t_2)\ \mathbf{A}\ (t_2 = t_3)] \supset (t_1 = t_3)$ by PR ▟

T57. ***Term Equality***

☎♋⦶ ⊢▯☎♦● $= t_2) \supset [\tau(t_1, t_1) = \tau(t_1, t_2)]$ for any term τ

(b) $\vdash (t_1 = t_2) \supset [\tau(t_1, t_1) = \tau(t_1, t_2)]$
provided τ does not contain the next operator.

Proof of (a):

By induction on the structure of τ.

Case: $\tau(t_1, t_1) = t_1$ ***and*** $\tau(t_1, t_2) = t_1$. Then

1. $\vdash t_1 = t_1$ by **A16**
2. $\vdash \Box(t_1 = t_2)\ \mathbf{3}\ [\tau(t_1, t_1) = \tau(t_1, t_2)]$
by PR and definition of $\tau(t_1, t_1)$ and $\tau(t_1, t_2)$

Case: $\tau(t_1,t_1)$ = t_1 and $\tau(t_1,t_2)$ = t_2. Then

1. ✋🖆 ❑☎♦● ▮ $t_2)$ ▯ $(t_1$ ▮ $t_2)$ by A3

2. ⊢ ❑ $(t1 = t_2) \supset [\tau(t_1,t_1) = \tau(t_1,t_2)]$
by the definition of $\tau(t_1, t_1)$ and $\tau(t_1, t_2)$

Case: $\tau(t_1, t_1) = f(\tau_1(t_1, t_1), \ldots, \tau_k(t_1, t_1))$ *and* $\tau(t_1, t_2) = f(\tau_1(t_1,t_2), \ldots, \tau_k(t_1,t_2))$. Then

1. I- $\Box(t_1 = t_2) \supset [\tau_i(t_1, t_1) = \tau_i(t_1, t_2)]$, for i = 1, . . . , k
by the induction assumption.

2. $\vdash \bigwedge_{i=1}^{k} [\tau_i(t_1, t_1) = \tau_i(t_1, t_2)] \supset$
$[f\ (\tau_1(t_1,t_1), \ldots, \tau_k(t_1,t_1)) = f(\tau_1(t_1,t_2), \ldots ,\tau_k(t_1,t_2))]$
by repeated application of Al7 and using T56 for transitivity of equality.

A typical step in this repeated application is:

$$\vdash [\tau_i(t_1, t_1) = \tau_i(t_1 t_2)] \supset$$
$$[f\ (\tau_1(t_1,t_2), \ldots, \tau_{i-1}(t_1,t_2), \tau_i(t_1,t_1), \ldots, \tau_k(t_1,t_1)) =$$
$$f(\tau_1(t_1,t_2), \ldots, \tau_{i-1}(t_1, t_2), \tau_i(t_1, t_2), \tau_{i+1}(t_1, t_1), \ldots, \tau_k(t_1,t_1))]$$

justified by Al7 and the fact that $\tau_i(t_1, t_2)$ is free for $\tau_i(t_1, t_1)$ in $f(...)$ since f does not contain any temporal operators.

3. ⊢ ❑ $(t1 = t_2) \supset [\tau(t_1,t_1) = \tau(t_1,t_2)]$
by 1, **2,** PR anti the definition of $\tau(t_1, t_1)$ and $\tau(t_1, t_2)$.

Case: $\tau(t_1, t_1)$ = O $\tau'(t_1, t_1)$ ***and,*** $\tau(t_1, t_2)$ = ***0*** $\tau'(t_1, t_2)$. Then

1. $\vdash \Box(t_1 = t_2) \supset [\tau'(t_1, t_1) = \tau'(t_1, t_2)]$ by the induction hypothesis

2. ✋🖆 ▯ ❑☎♦🗀▮ $t_2) \supset \text{O}[\tau'(t_1,t_1)$ ▮ $\tau'(t_1,t_2)]$ by *0 0*

3. I- $\text{O}[\tau'(t_1, t_1) = \tau'(t_1, t_2)] \supset [\text{O}\ \tau'(t_1, t_1) = \mathbf{0}\ \tau'(t_1, t_2)]$ by A1 8 and PR

4. $\vdash \Box(t_1 = t_2)$ 3 0 $\Box(t_1 = t_2)$ by A7

5. ⊢❑☎♦🗀▮ $t_2) \supset (\text{O}\ \tau'(t_1,t_1)$ ▮ ▯ $\tau'(t_1,t_2))$ by 4, 2, 3 and PR

6. ⊢❑☎♦● ▮ $t_2) \supset [\tau(t_1,t_1)$ ▮ $\tau(t_1,t_2)]$ by the definition of $\tau(t_1, t_1)$, $\tau(t_1, t_2)$.

Proof of (b):

1. I- $(t_1 = t_2) \supset [(\tau(t_1) = \tau(t_2)) \equiv (\tau(t_2) = \tau(t_2))]$ by Al7 (no 0 in τ)

2. **I-** $\tau(t_2) = \tau(t_2)$ by A16

3. $\vdash (t_1 = t_2) \supset (\tau(t_1) = \tau(t_2))$ by 1, 2 and PR ∎

The following theorem generalizes **A17** to arbitrary formulas.

T58. ***Substitutivity of Equality***

$\vdash \Box(t_1 = t_2) \supset [w(t_1, t_1) \equiv w(t_1, t_2)]$ where t_2 is free for t_1 in w.

Proof:

By induction on the structure of w.

Case: w contains no temporal operators. Then

1. $\vdash (t_1 = t_2) \supset [w(t_1, t_1) \equiv w(t_1, t_2)]$ by A17
2. $\vdash \Box(t_1 = t_2) \supset (t_1 = t_2)$ by A3
3. $\vdash \Box(t_1 = t_2) \supset [w(t_1, t_1) \equiv w(t_1, t_2)]$ by MP

Case: $w(t_1, t_2)$ is of the form $\tau_1(t_1, t_2) = \tau_2(t_1, t_2)$. Then

1. $\vdash \Box(t_1 = t_2) \supset [\tau_1(t_1, t_1) = \tau_1(t_1, t_2)]$ by T57
2. $\vdash \Box(t_1 = t_2) \supset [\tau_2(t_1, t_1) = \tau_2(t_1, t_2)]$ by T57
3. I - $[\tau_1(t_1, t_1) = \tau_1(t_1, t_2)] \supset [(\tau_1(t_1, t_1) = \tau_2(t_1, t_1)) \equiv (\tau_1(t_1, t_2) = \tau_2(t_1, t_1))]$
   by AL7 of the form $(\theta_1 = \theta_2) \supset [(\theta_1 = \tau_2(t_1, t_1)) \equiv (\theta_2 = \tau_2(t_1, t_1))]$
   with $\theta_1 = \tau_1(t_1, t_1)$ and $\theta_2 = \tau_1(t_1, t_2)$
4. $\vdash \Box(t_1 = t_2) \supset [(\tau_1(t_1, t_1) = \tau_2(t_1, t_1)) \equiv (\tau_1(t_1, t_2) = \tau_2(t_1, t_1))]$
   by 1, 3 and PR
5. $\vdash \Box(t_1 = t_2) \supset [(\tau_1(t_1, t_2) = \tau_2(t_1, t_1)) \equiv (\tau_1(t_1, t_2) = \tau_2(t_j, t_2))]$
   similarly by A17, using 2
6. $\vdash \Box(t_1 = t_2) \supset [(\tau_1(t_1, t_1) = \tau_2(t_1, t_1)) \equiv (\tau_1(t_1, t_2) = \tau_2(t_1, t_2))]$
   by 4, 5 and PR
7. $\vdash \Box(t_1 = t_2) \supset [w(t_1, t_1) \equiv w(t_1, t_2)]$ by the definition of $w(t_1, t_2)$

Case: w is of the form Cl ***u***. Then

1. $\vdash \Box(t_1 = t_2) \supset [u(t_1, t_1) \equiv u(t_1, t_2)]$ induction hypothesis
2. $\vdash \Box(t_1 = t_2)$ assumption

3. I- $u(t_1, t_1) \equiv u(t_1, t_2)$ by MP

4. ⊢ ❑ ♦☎♦●📪 $t_1) \equiv$ ❑ &,t2) by □□

Thus, ❑ ☎♦● = $t_2)$ I- [$\Box u(t_1, t_1) \equiv$ ❑ u(t1,t2)]

5. ✋📫 ❑ $\Box(t_1 = t_2)$3[$\Box u(t_1, t_1) \equiv$ ❑ u(t1, t_2)] by DED

6. ✋📫 ❑ ☎♦📂 =t_2)⊐[$\Box u(t_1, t_1) \equiv$ ❑ u(t1,t_2)] by T3 and PR

The cases in which w is of the form 0 u, 0 u, $\forall x.u$ and $\exists x.u$ are treated similarly, using the 0 O-rule, the 0 O-rule, the W-rule and the ∃∃-**rule,** respectively.

Case: w is of the form $u \mathcal{U} v$.

1. ✋📫 ❑ (t1 =$t_2) \supset [u(t_1, t_1) \equiv u(t_1, t_2)]$ induction hypothesis

2. ♦📫 ❑ (t1 =$t_2) \supset [v(t_1, t_1) \equiv v(t_1, t_2)]$ induction hypothesis

3. ♦📫 ❑☎♦📪 💾 $t_2)$ assumption

4. I- $u(t_1, t_1) \equiv u(t_1, t_2)$ by 1, 3 and MP

5. ⊢ $v(t_1, t_1) \equiv v(t_1, t_2)$ by 2, 3 and MP

6. ⊢ $(u(t_1, t_1) \mathcal{U} v(t_1, t_1)) \equiv (u(t_1, t_2) \mathcal{U} v(t_1, t_2))$ by 4, 5 and ER

Thus, ❑ ☎♦● = $t_2)$ I - $[(u(t_1, t_1) \mathcal{U} v(t_1, t_1)) \equiv (u(t_1, t_2) \mathcal{U} v(t_1, t_2))]$

7. ♦📫 □ ❑ (t1=$t_2) \supset [(u(t_1, t_1) \mathcal{U} v(t_1, t_1)) \equiv (u(t_1, t_2) \mathcal{U} v(t_1, t_2))]$ by DED

8. ⊢❑☎♦📪 💾 $t_2) \supset [(u(t_1, t_1) \mathcal{U} v(t_1, t_1)) \equiv (u(t_1, t_2) \mathcal{U} v(t_1, t_2))]$
by T3 and PR ▟

6. FRAME AXIOMS AND RULES

In this section we consider the consequences of the partition of the set of all variables into local and global variables. By thc semantic definition, global variables are given their value by the global- assignment a, and these values do not vary from slate to state. Consequcntly, for a global variablc u it must be universally true that u = 0 u, i.e., the value of u al any state is idcntical lo its valuc in thc next statc (see A19 bclow). The following axioms arc called frame axioms in reference to the "frame axiom" in Iloarc's deductive system for program vcrilication ([ILL]).

Recall that we split the set of our symbols into two subsets: global and local symbols. The logical consequence of this convention is thc following frame axiom:

A19. *Frame Axiom*

⊢ $x = \bigcirc x$ for cvcry global variable x

WC can therefore provc by induction on the structure of the term t and thc formula w the following frame ***theorems:***

T59. For a term t and formula w

(a) $\vdash t = Ot$
where t is global, i.e., does not contain local symbols

(b) $\vdash w \equiv \Box w$
whcrc w is global, i.e., does not contain local symbols.

(c) $\vdash w(\bigcirc y_1,$● $.\,.\,, \bigcirc y_n) \equiv \bigcirc\, w(y_1,$● ..,Y?a)
where $y_1, \ldots, y_n$ are all the local variablcs in w.

We present several frame theorems that facilitate moving global formulas in and out of the scope of tcmporal operators.

T60. ⊢ ❑ (Wl v w_2) ≡ (w_1 v ❑ w2)
whcre w_1 is global, i.e., contains no local symbols.

Proof:

1. $\vdash \sim w_1 \supset \Box\sim w_1$ — by T59b

2. ⊢[□(w_1vw_2)✌ ❑ ~w_1] ▤ ❑ ((Wl v w_2)A~w_1) — by T7 and PR

3. **I-** [(w_1 **v** w_2) **A** ~w_1] ⊃ w_2 — by PT

4. ⊢ [□(w_1 V w_2) A □~w_1] ⊃ □w_2 — by 2, 3, ❑ □ and I'R

5. ⊢ [□(w_1 V w_2) A ~w_1] ⊃ □w_2 — by 1, 4 and PR

6. ⊢ ❑ ☎♦🗁 ❖ w_2)⊃(w_1□w_2) — by PR

7. ⊢ w_1 ⊃❑ ●⏀● — by T59b

8. ⊢ (w_1 ❖ ❑ w_2)⊃ (□w_1 ❖ ❑♦♑✆ — by PR

⊢ (□w_1 ❖ □w_2)⊃❑☎⏀● ❖ w_2) — by T9

10. ♦📁 (w_1 ❖ ❑♦▤✆ ⊃❑☎⏀● ❖ w_2) — by 8, 9 and PR

11. ⊢ ❑ ☎♦🗁 ❖ w_2)≡(w_1v□w_2) — by 6, 10 and PR ▟

T61. $\vdash \Diamond(w_1$ A $w_2) \equiv (w_1$ A *0* $w_2)$ wherc w_1 is global.

Proof: The proof follows from T60 by duality.

A derived frame rule that we will bc using is

Frame Rule — FR

$$\frac{\vdash u \supset \Diamond v}{\text{I-}\ (w\ A\ u) \supset \Diamond(w\ A\ v)}$$

where w is global

Proof:

1. $\vdash u \supset \Diamond v$ — given

2. $\vdash (w\ A\ u) \supset (w\ A\ \Diamond v)$ — by PR

3 . $\vdash (w \wedge \Diamond v) \supset \Diamond(w \wedge v)$ — by T61 and PR

4. $\vdash (w\ A\ u) \supset \Diamond(w\ A\ v)$ — by 2, 3 and PR ∎

C. DOMAIN PART

The next part of the system contains domain axioms that specify the necessary properties of the domain of intcrcsl. Thus, to rcason about programs manipulating natural numbers, we nced the set of Pcano Axioms, and to rcason about trees we need a set of axioms giving the basic properties of trees and thc basic operations defined on them.

7. INDUCTION AXIOMS AND RULES

An essential axiom schema for many domains is the ***induction axiom schema.*** This (and all other schemas) should be formula14 to admit temporal instances as subformulas. Thus the induction principle for natural numbers can be stated as follows:

A20. *Induction Axiom*

$$\vdash \{R(0) \mathrm{A}\ \forall n[R(n) \supset R(n + 1)]\} \supset R(k)$$

for any statement R.

One instance of this axiom, which will be used later, is obtained by taking R(n) to bc □ l(Q(n) ⊃ $\diamond \psi$):

T62. ***Induction Theorem:***

$$\vdash \{\square(Q(0) \supset \diamond \psi) \wedge \forall n[\square(Q(n) \supset \diamond \psi) \supset$$ [illegible] $$1) \supset \diamond \psi)]\}$$

$$\supset \square(Q(k) \supset \diamond \psi).$$

Using this induction theorem wc can derive the following useful induction rule:

$\diamond$ ***Induction Rule*** — $\diamond$IND

$$\vdash Q(0) \supset \diamond \psi$$

$$\vdash Q(n + 1) \supset [\diamond \psi \vee \diamond Q(n)]$$

$$\vdash \mathrm{Q(k)} \supset 0 \psi$$

$\diamond$IND is useful for proving convergence of a loop: show that Q(0) guarantees 0 ψ and that for each n, either $Q(n + 1)$ implies Q(n) across the loop or it already establishes 0 ψ and no further execution is ncccssary. Then for any k, $Q(k)$ ensures that 0 ψ is established.

Proof:

1. $\vdash Q(0) \supset \diamond \psi$ given

2. [illegible] $\supset \square\, \psi)$ by □[

3. t- $Q(n + 1) \supset$ (0 $\psi \vee$ 0 Q(n)) given

4. $\vdash \Box(Q(n) \supset \Diamond\psi) \supset (\Diamond Q(n) \supset \Diamond\psi)$ by T6, T4 and PR

5. ⊢ [Q(n 🗏 1) ∧❑☎✈☎■✆ ⊃ ◇ψ)] ⊃ ◇ψ by 3, 4 and PR

6. ⊢❑☎✈☎▤ ⊃ ◇ψ) ⊃ $(Q(n+1) \supset \Diamond\psi)$ by PR

7. ⊢ ❑ lO(Q(n)⊃ ◇ψ) ⊃ ❑ (Q(n+l)⊃ ◇ψ) by ❑ lCl

✋📪 ❑ (Q(n)⊃ ◇ψ)⊃ ❑ l(Q(n+l)⊃ ◇ψ) by T3 and PR

⊢ ∀n[□(Q(n) ⊃ 0ψ) ⊃ ❑ (Q(n+1)⊃0ψ)] by ∀I

1 0 . $\vdash \Box(Q(k) \supset \Diamond\psi)$ by 2, 9 and T62

11. $\vdash Q(k) \supset \Diamond\psi$ by A3 and MP ▮

While induction over thc natural numbers is usually sufficient in order to prove properties of sequential programs, we nccd induction over morc gcneral orderings in order to rcason about concurrent prograrns ([LPS]). Thus we havc to formulate a more general induction principle over arbitrary well-founded orderings.

Lct (A, $\prec$) be a partially ordered set. We call the ordering $\prec$ ***a well-founded ordering*** if there cxists no infinitely decreasing sequence of elements in A:

$$\alpha_1 \succ \alpha_2 \succ \alpha_3 \succ \ldots$$

For each well-founded ordering (A, $\prec$), the following is a valid induction rule:

R5. *Well-Founded Induction Rule* — WIND

$$\frac{\vdash \forall\beta[(\beta \prec \alpha) \supset w(\beta)] \supset w(\alpha)}{\vdash w(\alpha)}$$

This rule should hold for an arbitrary temporal formula w(a) dependent on a global variable $\alpha \in$ A, and we adopt it as a primitive inference rule.

To justify the rule semantically we may arguc as follows:

Assume that the premise Lo the rule is true but the conclusion is not. Then there must exist a modcl M and an α_1 such that $w(\alpha_1)$ is false undcr M. By thc premisc thcrc must exist somc α_2 such Lhat $\alpha_2 \prec \alpha_1$ and $w(\alpha_2)$ is falsc undcr M. Arguing in a similar way wc obtain an infinitcly decreasing sequence:

$$\alpha_1 \succ \alpha_2 \succ \alpha_3 \succ \ldots$$

such that for each i, $w(\alpha_i)$ is false under M. This of course contradicts the well foundcdncss of $(A, \prec)$.

Note that the induction axiom and rules can be derivcd from WIND by taking (A, $\prec$) Lo be $(N, <)$.

In order to use thc WIND rule, one has to establish that thc ordering $\prec$ is indeed a well-founded ordering. Several specific orderings are known to be well-founded (such as lexicographic ordering over tuples of integers, multiscts, etc.), and may bc frccly used. I lowcvcr the gcncral statcmcnt that an ordering '$\prec$' is well-founded is a second order statement which may rcquire sccond ordcr reasoning for its establishment.

By substitution of a special form of a temporal formula we can obtain thc following induction principle for 0 formulas:

Well-Founded 0 Induction Rule -- OWIND

$$\frac{\vdash w(\alpha) \supset \Diamond\big(\psi \vee \exists\beta[(\beta \prec \alpha) \wedge w(\beta)]\big)}{\vdash w(\alpha) \supset 0\ \psi}$$

WC show that $\Diamond$WIND follows from WIND.

Proof:

1. $\vdash w(\alpha) \supset \Diamond\big(\psi \vee \exists\beta[(\beta \prec \alpha) \wedge w(\beta)]\big)$ given

2. $\vdash w(\alpha) \supset \big(\Diamond\psi \vee \Diamond\ \exists\beta[(\beta \prec \alpha) \wedge w(\beta)]\big)$ by T8 and PR

3. $\vdash \Box\big(\exists\beta[(\beta \prec \alpha) \wedge w(\beta)] \supset \Diamond\psi\big) \supset$
$\big(\Diamond\exists\beta[(\beta \prec \alpha) \wedge w(\beta)] \supset \Diamond\psi\big)$ by T6, T4 and PR

4. $\vdash \{w(\alpha) \wedge$❑☎♎⚐☯☎⚐ $\prec \alpha) \wedge w(\beta)] \supset \Diamond\ \psi)\} \supset \Diamond\psi$ by 2, 3 and PR

5. $\vdash \Box\big(\exists\beta[(\beta \prec \alpha) \wedge w(\beta)] \supset \Diamond\psi\big) \supset \big(w(\alpha) \supset \Diamond\psi\big)$ by PR

6. $\vdash \big(\exists\beta[(\beta \prec$ a) A $w(\beta)] \supset \Diamond\psi\big) \equiv \big(\sim\exists\beta[(\beta \prec \alpha)$ A $w(\beta)] \vee \Diamond\psi\big)$ by PT

7. $\vdash \big(\sim\exists\beta[(\beta \prec \alpha)$ A $w(\beta)] \vee \Diamond\ \psi\big) \equiv \big(\forall\beta[\sim(\beta \prec \alpha) \vee \sim w(\beta)] \vee \Diamond\psi\big)$
by A11, ER and PR

8. I- $\big(\forall\beta[\sim(\beta \prec \alpha) \vee \sim w(\beta)]$ v 0 $\psi\big) \equiv \forall\beta[(\beta \prec \alpha) \supset \big(w(\beta) \supset \Diamond\psi\big)]$
by T43, PR and ER, since 0 ψ docs not depend on β

9. $\vdash \big(\exists\beta[(\beta \prec \alpha) \wedge w(\beta)] \supset \Diamond\psi\big) \equiv \forall\beta[(\beta \prec \alpha) \supset \big(w(\beta) \supset \Diamond\psi\big)]$
by 6, 7, 8 and PR

10. $\vdash \Box\forall\beta[(\beta \prec \alpha) \supset \big(w(\beta) \supset \Diamond\psi\big)] \supset \big(w(\alpha) \supset \Diamond\psi\big)$ by 9, 5 and ER

11. t- ❑ lVp[(/I $\prec \alpha) \supset \big(w(\beta) \supset \Diamond\psi\big)] \supset$ ❑ (w(a) $\supset \Diamond\psi\big)$ by T3, $\Box\Box$ and PR

12. ⧫✆ $\forall\beta$❑ ☯☎⚐ $\prec \alpha) \supset \big(w(\beta) \supset \Diamond\psi\big)] \supset$❑☎⧫☎▯▯✆ $\supset \Diamond\psi\big)$ by T47 and PR

13. I- $\forall\beta[(\beta \prec \alpha) \supset \Box\big(w(\beta) \supset \Diamond\psi\big)] \supset$ ❑ (w(cy) $\supset \Diamond\psi\big)$
by T60, ER and PR, since $(\beta \prec$ a) is global

14. $\vdash$ ❑ (w(cr) $\supset \Diamond\psi\big)$ by WIND, taking w(a) to bc $\Box\big(w(\alpha) \supset 0\ \psi\big)$

15. $\vdash$ w(a) 3 $\Diamond\psi$ by A3 and PR ∎

D. PROGRAM PART

Our proof system must be augmcntcd by additional axioms that reflect thc structure of thc program under consideration. Thc additional axioms constrain the state sequences to be exactly the set of cxccution scqucnccs of the program under study. This relieves us from the need to include program text explicitly in the system; all the necessary information is captured by the additional axioms.

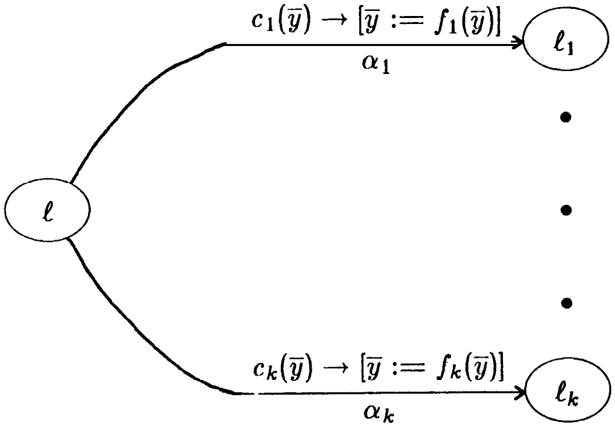
8. PROGRAMS AND COMPUTATIONS

In our model a concurrent program consists of m parallel processes:

$$P: \quad \bar{y} := g(\bar{x});\ [P_1 \| \ldots \| P_m].$$

Each process P_i is represented as a transition graph with locations (nodes) $L_i = \{\ell_0^i, \ldots, \ell_t^i\}$. The cdgcs in the graph are labelled by guarded commands of the form $c(\bar{y}) \rightarrow [\bar{y} := f(\bar{y})]$ whose meaning is that if $c(\bar{y})$ is true the edgc may bc traversed while replacing $\bar{y}$ by $f(\bar{y})$.

Let $\ell, \ell_1, \ell_2, \ldots, \ell_k \in L_i$ be locations in process P_i:



$c_1(\bar{y}) \rightarrow [\bar{y} := f_1(\bar{y})]$
α_1
ℓ_1
ℓ
$c_k(\bar{y}) \rightarrow [\bar{y} := f_k(\bar{y})]$
α_k
ℓ_k

The variables $\bar{y} = (y_1, \ldots, y_n)$ are shared by all processes. We define $E_\ell(\bar{y}) = c_1(\bar{y}) \vee \ldots \vee c_k(\bar{y})$ to bc thc ***exit condition*** at nodc ℓ. We do not requirc that thc conditions c_i be cithcr cxclusivc or cx haustivc.

The advantage of the transition graph rcprcscntation is that programs arc rcprcscntcd in a uniform way and that we have only to deal with one type of instruction. Wc show first that programs rcprcscntcd in a linear text form can easily bc translated into graph form.

Assumc that a linear text program allows the following types of instructions:

Assignment: $\bar{Y} := f(\bar{y})$

Conditional Branch: ***if*** $p(\overline{y})$ ***then go to*** ℓ_1 ***else go to*** ℓ_2

Halt: ***halt***

Waiting loop: ***loop until*** $p(\overline{y})$

loop w hile $p(\overline{y})$

and the semaphore instructions

Request: ***request(y)***

Release: release(y)

A linear text program for each of the processes has the following form:

$\ell_0 : I_0$
$\ell_1 : I_1$

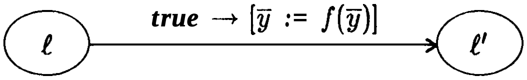
ℓ_t : ***halt*** or ***go to*** ℓ_j

where $\ell_0, \ell_1, \ldots, \ell_t$ are labels and $I_0, I_1, \ldots$ are instructions from the list above.

The graph representation of such a program for process P_i will be a labelled graph with $L_i = \{\ell_0, \ldots, \ell_t\}$ as the set of nodes. For each instruction I at label $\ell \in L_i$ we construct edges as follows:

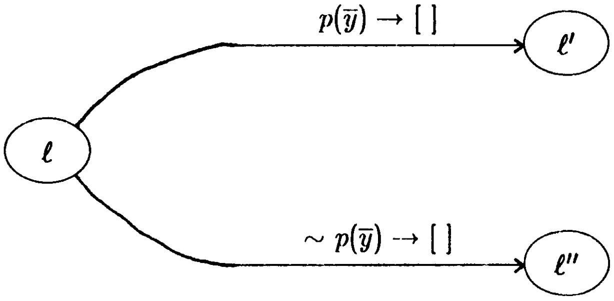
- for the instruction
$\ell : \overline{y} := f(\overline{y})$
$\ell' :$

construct



- for the instruction
ℓ : ***if*** $p(\overline{y})$ ***then go to*** ℓ' ***else go to*** ℓ''
ℓ' :

construct

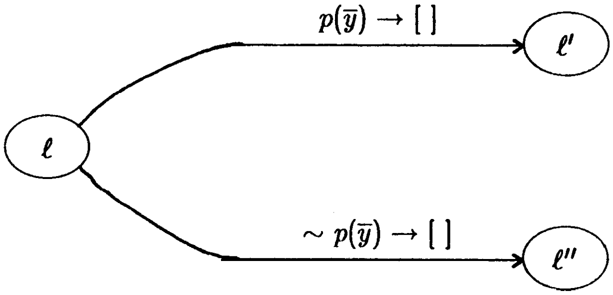


$p(\overline{y}) \rightarrow [\ ]$
ℓ'
ℓ
$\sim p(\overline{y}) \rightarrow [\ ]$
ℓ''

▶ for the instruction

ℓ : ***if*** $p(\overline{y})$ ***then go to*** ℓ'
ℓ'' :

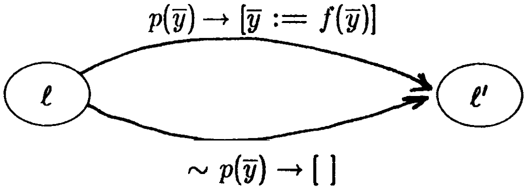
construct


$p(\overline{y}) \rightarrow [\,]$
ℓ'
ℓ
$\sim p(\overline{y}) \rightarrow [\,]$
ℓ''

▶ for the instruction

ℓ : ***if*** $p(\overline{y})$ ***then*** $\overline{y} := f(\overline{y})$
ℓ' :

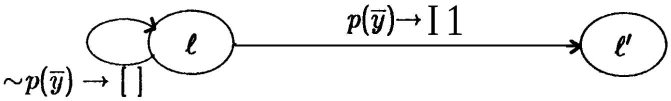
construct


$p(\overline{y}) \rightarrow [\overline{y} := f(\overline{y})]$
ℓ
ℓ'
$\sim p(\overline{y}) \rightarrow [\,]$

▶ for the instruction

ℓ : ***loop until*** $p(\overline{y})$
ℓ' :

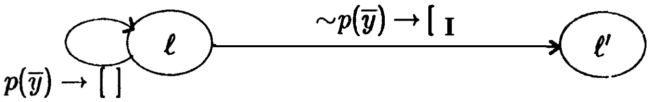
construct



▶ for the instruction

ℓ : ***loop*** *while* **p(y)**
ℓ' :

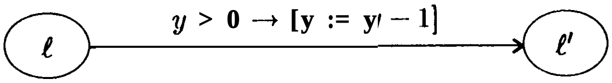
construct


$\sim p(\overline{y}) \rightarrow$ [I
ℓ
ℓ'
$p(\overline{y}) \rightarrow [\,]$

▶ for the instruction

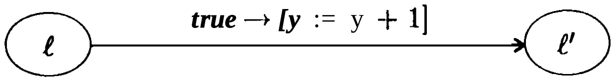
ℓ : ***request(y)***
ℓ' :

construct



▶ for the instruction
ℓ : ***release(y)***
ℓ' :

construct



For ***halt*** at labcl ℓ we construct no edgcs out of ℓ.

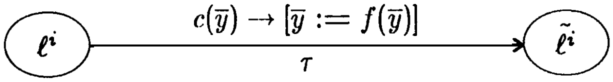
The actual translation into graph form need not bc carricd out explicitly. Rather, the gcncral axiomatic description of transition diagrams can be easily translated to axioms for each of the types of instructions in the linear text form.

A statc of the prograrn P is a tuple of the form $s = \langle \bar{\ell}; \bar{\eta} \rangle$ with $\bar{\ell} \in L_1 \times \ldots \times L_m$ and $\bar{\eta} \in D^n$, where D is the domain over which the program variables $y_1, \ldots, y_n$ range. The vector $\bar{\ell}$ = (a', . . . ,ℓ^m) is thc set of current locations which are next to bc executed in each of the processes. The vector $\bar{\eta}$ is thc set of current values assumed by thc prograrn variables $\bar{y}$ at statc s.

Lct $s = \langle \ell^1, \ldots, \ell^i, \ldots, \ell^m; \bar{\eta} \rangle$ be a state. We say that proccss P_i is ***enabled*** on s if $E_{\ell^i}(\bar{\eta})$ = ***true.*** This implies that if we lct P_i run at this point, there is at least onc condition c_j among thc edgcs departing from ℓ^i that is true. Othcrwise, wc say that P_i is ***disabled*** on s. An cxamplc of a disabled process is the casc where ℓ^i labels an instruction ***request(y)*** and y = 0. Another cxample is that of ℓ^i labeling a ***halt*** statement. A statc is defined to be ***terminal*** if no P_i is enabled on it.

Given a program P wc define the notion of a ***computation step*** of P.

Lct $s = \langle \ell^1, \ldots, \ell^m; \bar{\eta} \rangle$ and $\tilde{s} = \langle \tilde{\ell}^1, \ldots, \tilde{\ell}^m; \tilde{\bar{\eta}} \rangle$ be two states of P. Let τ be a transition in P_i of the form:



such that $c(\bar{\eta})$ = ***true,*** $\tilde{\bar{\eta}} = f(\bar{\eta})$, and for every j $\neq i$, $\tilde{\ell}^j = \ell^j$. Thcn we say that $\tilde{s}$ can be obtained from s by a ***Pi-step*** (a single computation stcp), and write

$$s \xrightarrow{P_i} \tilde{s}.$$

An ***initialized admissible computation*** of a program P for an input $\bar{x} = \bar{\xi}$ is a labcllcd maximal sequence of states of P:

$$\sigma : s_0 \xrightarrow{P_{i_1}} s_1 \xrightarrow{P_{i_2}} s_2 \xrightarrow{P_{i_3}} s_3 \longrightarrow \ldots$$

which satisfies the following three conditions. (The sequence σ is considered *maximal* if it cannot be extended, i.e., it is either infinite or ends with a state s_k which is terminal.)

A. Initialization:

The first state s_0 has the form:

$$s_0 = \langle \bar{\ell}_0;\ g(\bar{\xi})\rangle$$

where $\bar{\ell}_0 = (\ell_0^1, \ldots, \ell_0^m)$ is the vector of initial locations. The values $g(\bar{\xi})$ are the initial values assigned to the $\bar{y}$ variables for the input ξ.

B. State to State Sequencing:

Every step in the computation $s \xrightarrow{P_i} \tilde{s}$, is justified by a Pi-step.

C. Fairness:

Every P_i which is enabled on infinitely many states in σ rnust be activated infinitely many times in σ, i.e., there must be an infinite number of P_i-steps in σ.

We define an ***admissible computation*** of P for input $\bar{\xi}$ to be either an initialized admissible computation or a suffix of an initialized admissible computation.

Thus the class of admissible computations is closed under the operation of taking the suffix. This is needed in order to ensure soundness of the inference rule □I (123). We denote the class of all $\bar{\xi}$-admissible computations of a program P by $\mathcal{A}(P, \bar{\xi})$.

An admissible computation is said to be ***convergent*** if it is finite:

$$\sigma:\ s_0 \xrightarrow{P_{i_1}} s_1 \longrightarrow \ldots \xrightarrow{P_{i_f}} s_f\ .$$

If the terminal state s_f in a convergent computation is of the form $s_f = \langle \ell_t^1, \ldots, \ell_t^m; \bar{\eta}\rangle$, where each ℓ_t^i labels a halt instruction, we say that the ***computation has terminated.*** Otherwise, we say that the ***computation has blocked*** or ***is deadlocked.***

In order to describe properties of states we introduce a vector of ***locution variables*** $\bar{\pi} = (\pi_1, \ldots, \pi_m)$. Each π_i ranges over L_i, and assumes the location value ℓ^i in a state

$$s = \langle \ell^1, \ldots, \ell^i, \ldots, \ell^m; \bar{\eta}\rangle.$$

Thus we may describe a state $s = \langle \bar{\ell};\ \bar{\eta}\rangle$ by saying that in this state $\bar{\pi} = \bar{\ell}$ and $\bar{y} = \bar{\eta}$.

A ***state formula*** $Q = Q(\bar{\pi}; \bar{y})$ is any formula which contains no temporal operators. It is built up of terms and predicates over the location and program variables $(\bar{\pi};\ \bar{y})$ and may also refer to global variables.

We frequently abbreviate the statement $\pi_i = \ell$ to $at\,\ell$. Since the L_i's are disjoint, there is no difficulty in identifying the particular π_i which assumes the value ℓ.

Let us consider a program P over a domain $\boldsymbol{D}$ with fixed interpretation $\boldsymbol{I}$ for all the predicate, function and individual constant symbols. A model M is said to be ***admissible*** for P if it has the form:

$$\boldsymbol{M = (I, \alpha, \hat{\sigma})}$$

where α and $\hat{\sigma}$ satisfy the following condition:

There exists an $\alpha[\bar{x}]$-admissiblc comnutation $\sigma \in \mathcal{A}(P, \alpha[\bar{x}])$ such that

either

σ is infinite: $\sigma = s_0 \xrightarrow{P_{i_1}} s_1 \xrightarrow{P_{i_2}}$ s2 $\longrightarrow s_3 \ldots$

and

$\hat{\sigma} =$ so, s_1, 52, . . .

or

σ is finite: $\sigma = s_0 \xrightarrow{P_{i_1}} s_1 \xrightarrow{P_{i_2}} s_2 \longrightarrow \ldots \xrightarrow{P_{i_f}} s_f$

and then

$\hat{\sigma} = s_0, s_1,$ 5 2 , $\ldots, s_f, s_f,$. '*'

Thus we force $\hat{\sigma}$ to be always infinite by indefinitely repeating the last state of σ if it is finite. This corresponds to our intuition that while the computation may have terminated, time still marches on, but no furl her change in the program will ever occur.

Let us denote the class of all admissible models for a program P by C(P). Note that this class, differently from A(P, $\bar{\xi}$), contains computations corresponding to different inputs.

Wc define the state formula stating that a process P_i is enabled as follows:

$$Enabled(P_i;\bar{\pi};\ \bar{y}) = \bigwedge_{\ell \in L_i} [(\pi_i = \ell) \supset E_\ell(\bar{y})].$$

For the complete program P we defined

$$Enabled(P;\ \bar{\pi};\ \bar{y}) = \bigvee_{i=1}^{m} Enabled(P_i;\ \bar{\pi};\ \bar{y}).$$

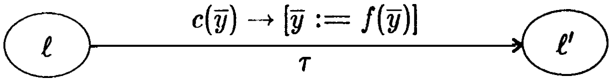
Thus a state $s = \langle \bar{\ell};\ \bar{\eta} \rangle$ is terrninal iff

$$Enabled(\ P;\ \bar{\ell};\ \bar{\eta}) = \textbf{\textit{false}}$$

and we may define

$$\textbf{\textit{Terminal}}\ (\bar{\pi};\ \bar{y}) \equiv\ \sim\! Enabled(P;\bar{\pi};\bar{y}).$$

Let the following be a transition τ in process P_i:


$c(\overline{y}) \rightarrow [\overline{y} := f(\overline{y})]$
ℓ
τ
ℓ'

We define the transformation associated with the transition τ by:

$$r_\tau(\overline{\pi};\, \overline{y})\, .\, \left(\overline{\pi}[\ell'/\pi_i];\, f(\overline{y})\right).$$

The transformation is obtained by replacing the current value ℓ of π_i by ℓ' and the values of $\overline{y}$ by f(Y)*

Let $\varphi(\overline{\pi};\, \overline{y})$ and $\psi(\overline{\pi};\overline{y})$ be two state formulas. Wc say:

- ***The transition*** τ ***leads from*** φ ***to*** ψ if the following implication is valid:
  $$[\varphi(\overline{\pi};\, \overline{y}) \text{ A } at\,\ell \text{ A } c(\overline{y})] \supset \psi\left(r_\tau(\overline{\pi};\, \overline{y})\right).$$
- ***The*** process P_i ***feuds from*** φ ***to*** ψ if every transition τ in P_i leads from φ to ψ.
- ***The program P feuds from*** φ ***to*** ψ if every P_i leads from φ to ψ.

We are ready now to give a temporal axiornatization for the notion of computation under the program ***P.***

9. AXIOMS AND RULES FOR CONCURRENT PROGRAMS

The first axiom states that the location variable π_i may only assumc values in L_i.

A21. ***Location Axiom*** -- LOC

$$\vdash \pi_i \in L_i \quad \text{for } i = 1, \ldots, m.$$

This is an abbreviation for:

$$\vdash (\pi_i = \ell_0^i) \vee (\pi_i = \ell_1^i) \vee \ldots \vee (\pi_i = \ell_t^i).$$

Since all the locations are disjoint, it also follows from thc equality axioms that π_i may be equal to at most one ℓ_j^i at a time.

For each of thc three requirements defining an admissible computation we have a corresponding inference rule schcme:

R6. ***Initialization*** -- INIT

For an arbitrary temporal formula w:

$$\frac{\vdash [at\,\overline{\ell}_0 \text{ A } \overline{y} = \text{y(z)}] \supset \Box \text{ w}}{\vdash \text{clw}}$$

For let us assume that the premise to this rule holds. This implies that Cl w is true for all initialized computations. By the semantic definition of ❑ , this implies that w is true for every suffix of an initialized computation, i.e., for every admissible computation. Thus, w is C(P)-valid, and by generalization (□I) so is ❑ lzu.

R7. ***Transition*** -- TRNS

Let $\varphi(\overline{\pi};\ \overline{y})$ and $\psi(\overline{\pi};\overline{y})$ be two state formulas.

$\vdash P$ leads from φ to ψ

$\vdash [\varphi(\overline{\pi};\overline{y}) \wedge \mathit{Terminal}(\overline{\pi};\ \overline{y})] \supset \psi(\overline{\pi};\ \overline{y})$

$\vdash \varphi \supset \bigcirc\psi$

Indeed let s be a state in the sequence $\hat{\sigma}$ corresponding to an admissible computation σ, and let s' be its successor in $\hat{\sigma}$. Assume that $\varphi(s)$ is true. There are two cases to bc considered. In the first case, s' is derived from s by a Pi-step for some i = 1, . . . ,m. But then, by the first premise, P_i leads from φ to ψ and therefore ψ must be true for s'. In the other case, s is terminal and s' = s the repetition of the terminal state of a finite computation. But then s is terminal and satisfies the antecedent of the second premise, leading to $\psi(s) = \psi(s') = \mathit{true}$. Hence, in both cases $\psi(s')$ must hold and the conclusion of the rule follows.

Note that the first premise lo this rule requires establishing rnany conditions involving the individual transitions of each of the proccsscs. However, by examining the definitions of “leading from φ to ψ” wc see that they are all expressible as classical statements involving no temporal operators. Therefore this premise should bc provable from the domain axioms plus the usual predicate calculus proof system. The second premise is also classical, and ensures the consequence after the sequence has reached a terminal state.

R8. ***Fairness*** -- FAIR

Let $\varphi(\overline{\pi};\ \overline{y})$ and $\psi(\overline{\pi};\ \overline{y})$ be two state formulas and P_k be one of the processes.

A. I- P leads from φ to φ V ψ

B. t- P_k leads frorn φ to ψ

$\vdash [\varphi$ A □ ◇ $\mathit{Enabled}(P_k)] \supset \varphi\mathcal{U}\psi$

To give a semantic justification of this rule, consider a computation such that φ is true initially. By A, φ will hold until ψ is realized, if ever. By B, once P_k will bc activated in a state satisfying φ it will achieve ψ in one step. Consider now a sequence σ such that φ A □ ***0*** $\mathit{Enabled}(P_k)$ is true on σ. This means that φ is initially true and P_k is enabled infinitely many times in σ. By fairness, P_k will eventually be activated, which, if ψ has not been realized before, will achieve ψ in one step.

Since $(\varphi\mathcal{U}\psi) \supset$ 0 ψ, wc often use the FAIR rule in order to derive the consequence

$[\varphi$ A □ 0 $\mathit{Enabled}(P_k)] \supset$ 0 ψ.

There arc several derived rules that can bc obtained from the above axiomatization,

Invariance Rule — INV

$$\frac{\vdash P \text{ leads from } \varphi \text{ to } \varphi}{\vdash \varphi \supset \Box\varphi}$$

Proof:

1.	I- P leads from φ to φ	given
2.	$\vdash [\varphi \wedge \mathit{Terminal}] \supset \varphi$	by PT
3.	$\vdash \varphi \supset \bigcirc\varphi$	by TRNS
4.	$\vdash \varphi \supset \Box\varphi$	by CI ▟

Initialized Invariance Rule -- IINV

Let φ be a state formula

-- $\quad$ ***t-*** $[\mathit{at}\,\bar{\ell}_0$ ***A*** $\bar{y}$ ***= g(Z)]*** $\supset \varphi$

$$\frac{\vdash P \text{ leads from } \varphi \text{ to } \varphi}{\vdash \Box\varphi}$$

Proof:

1.	$\vdash$ ***[at*** $\bar{\ell}_0$ ***A*** $\bar{y}$ ***= g(F)]*** $\supset \varphi$	given
2.	t- P leads from φ to φ	given
3.	$\vdash \varphi \supset \Box\varphi$	by 2 and INV
4.	$\vdash [\mathit{at}\,\ell_0$ ***A*** $\bar{y}$ ***= g(Z)]*** $\supset$ ❑ ***lp***	by 1, 3 and PR
5.	$\vdash \Box\varphi$	by INIT ▟

The IINV rule is the rule most often used in order to establish invariance properties of programs.

Unless Establishment Rule -- UER

Let φ be a state formula

$$\frac{\vdash P \text{ leads from } \varphi \text{ to } \varphi \vee \psi}{\vdash \varphi \supset (\varphi\,\mathcal{U}\,\psi)}$$

Proof:

1.	I- P leads from φ to φ v ψ	given

2. $\vdash \varphi \supset (\varphi \vee \psi)$ by PT

3. $\vdash [\varphi \wedge \mathit{Terminal}] \supset (\varphi \vee \psi)$ by PR

4. $\vdash \varphi \supset \bigcirc(\varphi \vee \psi)$ by 1, 3 and TRNS

5. $\vdash \varphi \supset (\varphi \mathfrak{U} \psi)$ by $\mathfrak{U}$I ▟

The following rule is a consequence of the FAIR rule.

***Eventuality Rule* ----** EVNT

Let $\varphi(\bar{\pi}; \bar{y})$ and $\psi(\bar{\pi}; \bar{y})$ be two state formulas and P_k one of the processes.

A. I- ***P*** leads from φ to $\varphi \vee \psi$

B. $\vdash P_k$ leads from φ to ψ

C **.** $\vdash \varphi \supset \Diamond(\psi$ **v** $\mathit{Enabled}(P_k))$

$$\overline{\hphantom{\vdash \varphi \supset \Diamond(\psi \vee Enabled(P_k))}}$$

$\vdash \varphi \supset \varphi \mathfrak{U} \psi$

Proof:

1. I- P leads from φ to φ v ψ given

2. t- P_k leads from φ to ψ given

3. ***t -*** $\varphi \supset$ ***0*** $(\psi$ ***v*** $\mathit{Enabled}(P_k))$ given

4. t- $[\varphi$ A ❑]Obhu$\mathit{bled}(P_k)] \supset \varphi \mathfrak{U} \psi$ by 1, 2 and FAIR

5. $\vdash \varphi \supset (\Box\varphi \vee \varphi \mathfrak{U} \psi)$ by 1 and CINV

6 . I - $[\varphi$ A $\Box{\sim}\psi] \supset \Diamond \mathit{Enabled}(P_k)$ by 3, T8, A1 and PR

7. ♦📬 ❑ (pA$\Box{\sim}\psi)\supset$Cl$\mathbf{0}\mathit{Enabled}(P_k)$ by □□

8. $\vdash [\Box\varphi$ A $\Box \sim\psi] \supset \Box \Diamond \mathit{Enabled}(P_k)$ by T3, T7 and PR

9. I- $[\Box \varphi$ A $\sim$ Cl $\Diamond \mathit{Enabled}(P_k)] \supset \Diamond \psi$ by A1 and PR

10. $\vdash \Box\varphi \supset \Diamond\psi$ by 4, 9, A3, A10 and PR

11. $\vdash \Box\varphi \supset \varphi \mathfrak{U} \psi$ by 10, T24 and PR

12. $\vdash \varphi \supset \varphi \mathfrak{U} \psi$ by 5, 11 and PR ▟

In contrast with earlier rules, premise C of EVNT is not purely classical since it contains the temporal operator 0. Since C has a form similar to the conclusion of the EVNT rule, it is Lo be expected lhat its derivation will require once more the application of the EVNT rule. This seems to imply circular reasoning. However, note that at each nested application of the EVNT rule, another P_k is taken out of consideration. This is because in trying to establish ***0*** $Enabled(P_k)$ ***we*** need not consider any P_k-steps at all, since when they are possible, P_k is already enabled.

A useful special case of C that frequently suffices for the application of the EVNT rule is:

$$C' : \ \vdash \varphi \ 3 \quad [\psi \lor \quad Enabled(P_k)].$$

Note that the EVNT rule can also be used to establish properties of the form

$$\varphi \supset \Diamond \psi,$$

since $\varphi \mathcal{U} \psi \supset \Diamond \psi$.

The EVNT rule is the one most often used in order to establish both eventuality (livencss) properties and precedence properties.

E. EXAMPLES

In this section we present several examples of proofs of properties of programs using the proof system described above.

10. EXAMPLE 1: DISTRIBUTED GCD

Let us consider the following example of a program computing the greatest common divisor of two positive integers in a distributed manner.

```
                         $(y_1, y_2) := (x_1, x_2)$

$\ell_0$ : if $y_1 > y_2$ then $y_1 := y_1 - y_2$        $m_0$ : if $y_1 < y_2$ then $y_2 := y_2 - y_1$
$\ell_1$ : if $y_1 \neq y_2$ then go to $\ell_0$           $m_1$ : if $y_1 \neq y_2$ then go to $m_0$
$\ell_2$ : halt                                  $m_2$ : halt

      $- P_1 -$                                    $- P_2 -$
```

We wish to prove total correctness for this program, i.e.,

Theorem:

$$\vdash [at(\ell_0, m_0) \wedge (y_1, y_2) = (x_1, x_2)] \supset \Diamond[at(\ell_2, m_2) \wedge y_1 = gcd(x_1, x_2)]$$

We will split lhe proof into two parts, proving separately invariance and termination.

Lemma A:

$$\vdash \Box[gcd(y_1, y_2) = gcd(x_1, x_2)]$$

Proof of Lemma A:

Let us denote $gcd(y_1, y_2) = gcd(x_1, x_2)$ by $\tilde{\varphi}(x_1, x_2, y_1, y_2)$.

It is easy to check that every transition in P leads from $\tilde{\varphi}$ to $\tilde{\varphi}$. Also

$$\vdash [(y_1, y_2) = (x_1, x_2)] \supset \tilde{\varphi}(x_1, x_2, y_1, y_2).$$

Thus we have the two premises to the IINV rule, which yields the desired result. ∎

Lemma B:

$$\vdash [at\ell_{0,1} \wedge atm_{0,1} \wedge (y_1,y_2) > 0 \text{ A } (y_1 + y_2) \leq \text{n + 1) A } y_1 \neq y_2]$$
$$\supset \Diamond[at\ell_{0,1} \wedge atm_{0,1} \text{ A } (y_1,y_2) > 0 \text{ A } (y_1 + y_2 \leq n)]$$

Here we use $at\ \ell_{0,1}$ as an abbreviation for $at\ell_0$ v $at\ell_1$, $atm_{0,1}$ for $atm_0 \vee atm_1$ and $(y_1,y_2) > 0$ for $(y_1 > 0) \wedge (y_2 > 0)$.

Proof of Lemma B:

Let us define

$$\varphi(y_1,y_2,n): \quad at\ell_{0,1} \wedge atm_{0,1} \wedge (y_1,y_2) > 0 \text{ A } (y_1 + y_2 \leq n).$$

Thus we have to prove: _

$$\vdash [\varphi(y_1, y_2, n + 1) \wedge (y_1 \neq y_2)] \supset \Diamond\ \varphi(y_1,y_2,n).$$

. We will split the proof into two cases:

B1. $\vdash [\varphi(y_1,y_2,n+1) \wedge (y_1 > y_2)] \supset \Diamond \varphi(y_1,y_2,n)$

B2. $\vdash [\varphi(y_1, y_2, n + 1) \wedge (y_1 < y_2)] \supset \Diamond\ \varphi(y_1, y_2, n)$

The lemma obviously follows from these two statements.

To prove B1 we first observe that by PR:

1. $\vdash \varphi(y_1,y_2,n+1) \supset (at\ell_0 \vee at\ell_1)$

Consider therefore first the case that P_1 is at ℓ_0. We take

$$\varphi': \quad \varphi(y_1,y_2,n + 1) \wedge (y_1 > y_2) \wedge at\ell_0$$
$$\psi': \quad \varphi(y_1,y_2,n).$$

We claim that φ' and ψ' satisfy the premises of EVNT with $P_k = P_1$.

To see this, consider requirement A of EVNT that states that every transition in P leads from φ' to φ' v ψ'.

Consider transitions in P_2. The only relevant ones are $m_0 \rightarrow m_1$ and transitions leading out of m_1. The transition mu $\rightarrow m_1$ under $y_1 > y_2$ leaves φ' invariant. Again, under $y_1 > y_2$ the only transition out of m_1 goes to m_0 leaving φ' invariant.

The only transition enabled in P_1 *is* $\ell_0 \rightarrow \ell_1$ which rcplaccs (y_1,y_2) by $(y_1 - y_2, y_2)$. If $y_1 + y_2 \le n + 1$ and $y_1 > 0$, $y_2 > 0$ then certainly $(y_1 - y_2) + y_2 \le$ n and $(y_1 - y_2) > 0, y_2 > 0$. Thus $\ell_0 \rightarrow \ell_1$ leads from φ' to ψ'. This also establishes requirement B with $P_k = P_1$.

Since E_{ℓ_0} = *true*, condition C is trivially fulfilled. Consequently wc conclude by the EVNT rulc that $\vdash \varphi' \supset 0\ \psi'$, i.e.,

2. $\vdash [\varphi(y_1,y_2,n+1) \wedge (y_1 > y_2) \wedge at\ell_0] \supset \Diamond\ \varphi(y_1,y_2,n)$.

Consider next the case where P_1 is at ℓ_1. By taking

φ'' : $\varphi(y_1,y_2,n + 1) \wedge (y_1 > y_2)$ A $at\ell_1$

$\psi'' = \varphi'$: $\varphi(y_1,y_2,n + 1)$ A $(y_1 > y_2)$ A $at\ell_0$.

Wc can show that the premises of the EVNT rule are satisfied with rcspccl to φ'', ψ''. Consequently wc have $\vdash \varphi'' \supset 0\ \psi''$, i.e.,

3. $\vdash [\varphi(y_1,y_2,n + 1) \wedge (y_1 > y_2) \wedge at\ell_1] \supset$
 $\Diamond[\varphi(y_1,y_2,n + 1) \wedge (y_1 > y_2) \wedge at\ell_0]$

4. $\vdash [\varphi(y_1,y_2,n + 1) \wedge (y_1 > y_2) \wedge at\ell_1] \supset \Diamond\varphi(y_1,y_2,n)$ by 2, 3 and OC

5. $\vdash [\varphi(y_1,y_2,n + 1) \wedge (y_1 > y_2)] \supset \Diamond\varphi(y_1,y_2,n)$ by 1, 2, 4 and PR

This establishes B1.

By a symmetric argument wc can establish B2. By propositional reasoning B1 and 132 lead to Lemma B. ▟

Proof of theorem:

We will now proceed with the proof of the main theorem.

6. $\vdash [\varphi(y_1,y_2,n + 1) \wedge (y_1 \neq y_2)] \supset \Diamond\varphi(y_1,y_2,n)$ Lemma B

7. $\vdash \varphi(y_1,y_2,n + 1) \supset [(y_1 = y_2) \vee \Diamond\varphi(y_1,y_2,n)]$ by PR

8. $\vdash \varphi(y_1,y_2,n + 1) \supset [\Diamond(y_1 = y_2) \vee \Diamond\varphi(y_1,y_2,n)]$ by T1 and PR

9. $\vdash \sim\varphi(y_1,y_2,0)$ by PR,
using the domain property that the conjunction
$(y_1 > 0) \wedge (y_2 > 0)$ A $(y_1 + y_2 \le 0)$ is impossible

10. $\vdash \varphi(y_1,y_2,0) \supset \Diamond(y_1 = y_2)$ by PR

11. $\vdash \varphi(y_1,y_2,n) \supset \Diamond(y_1 = y_2)$ by 8, 10 and OIND

12. $\vdash \exists n.\varphi(y_1,y_2,n) \supset \Diamond(y_1 = y_2)$ by $\exists$I

13. $\vdash [at(\ell_0,m_0) \wedge (y_1,$ Y2) = $(x_1,x_2) > 0]$ 3 $\exists n.\varphi(y_1,y_2,n)$

by taking $n = x_1 + x_2 > 0$.

By considering the different locations of P_1 and P_2 under the assumption that $y_1 = y_2$ it is easy (though long if carried out in full detail) to establish

14. $\vdash (y_1 = y_2) \supset \Diamond[at(\ell_2, m_2)\ \text{A}\ (y_1 = y_2)]$.

By combining 12, 13 and 14 using OC we obtain:

15. $\vdash [at(\ell_0, m_0) \wedge (y_1, y_2) = (x_1, x_2) > 0] \supset \Diamond[at(\ell_2, m_2)\ \text{A}\ (y_1 = y_2)]$.

Togelher with lemma **A** and T10 this gives

16. $\vdash [at(\ell_0, m_0) \wedge (y_1, y_2) = (x_1, x_2) > 0] \supset \Diamond[at(\ell_2, m_2) \wedge y_1 = gcd(x_1, x_2)]$
since $(y_1 = y_2) \supset$ Y I $=$ $gcd(y_1, y_2)$

Note that theorcm T10 enables us to infer frorn a previously established invariant I- Cl $\tilde{\varphi}$ and an implication $\vdash w_1 \supset \Diamond\, w_2$ the implication $\vdash w_1$ 3 $\Diamond(w_2 \wedge \tilde{\varphi})$. ■

II. EXAMPLE 2: SEMAPHORES

For our next example we will present a very simple program with semaphores:

```
                         Y := 1

      ℓ0 : request(y)                  m0 : request(y)
      ℓ1 : release(y)                  m1 : release(y)
      ℓ2 : go to ℓ0                    m2 : go to m0

          – P1 –                           – P2 –
```

This example models a solution to the mutual exclusion problem using semaphores.

There are two properties that we wish to prove for this program. The first is that of mutual exclusion, namely:

Lemma A:

$$\vdash \Box[(\sim at\,\ell_1) \vee (\sim at\,m_1)]$$

Proof: .

Take

$$\varphi(\pi_1, \pi_2; y): \quad (at\,\ell_1 + at\,m_1 + y = 1) \wedge (y \geq 0).$$

In expressions such as the abovc wc interpret propositions as having the numerical value 1 when true and 0 otherwise.

We can easily show that φ is preserved under every transition. For examplc, consider the transition $\ell_0 \rightarrow \ell_1$. When it is enabled, wc havc y > 0, and the transition assigns to the variable y the valuc y − 1 which is nonnegative. Considering thc value of the sum

$$at\ell_1 + atm_1 + \text{y},$$

$at\ell_1$ changes from 0 to 1 on this transition but y is decrcmented by 1. Consequently the value of thc sum remains invariant.

Initially, $at\ell_1$ + atm_1 + y = 0 + 0 + 1 = 1 and y = $1 \geq 0$.

Hcnce φ satisfics the two premises of the IINV rule, from which wc conclude

$$I_1 : \quad \text{t-}\ \Box[at\ell_1 + atm_1 + \text{y} = 1)\ \text{A}\ (\text{y} \geq \text{o})].$$

This implies

⊢▯☯♦♦📖🗏 $atm_1 \leq 1]$

which is equivalent to Lemma A. ▟

The second property is that of accessibility. It states that each process will eventually be admitted to its critical scction. This is cstablished by:

Lemma B:

$$\vdash at\ell_0 \supset \Diamond\ at\ell_1$$

and

$$\vdash atm_0 \supset \boldsymbol{0}\ atm_1$$

Proof:

Lct us define

$$\varphi_1 : \quad at\ell_0 \wedge atm_1\ \text{A}\ \ \text{y} = 0$$

$$\psi_1 : \quad y > 0$$

We show that φ_1 and ψ_1 satisfy thc conditions of the EVNT rule with $\boldsymbol{k = 2}$.

. In fact the only cnablcd transition is $m_1 \rightarrow m_2$ which does lead from φ_1 to ψ_1. While at m_1, P_2 is always enabled. Thus we conclude:

1. $\vdash [at\ell_0\ \text{A}\ atm_1\ \text{A}\ \text{y} = 0] \supset \Diamond(y > 0)$ by EVNT with $\boldsymbol{k} = 2$

2. I- $[at\ell_0$ A $atm_1] \supset \Diamond(y > 0)$ by I_1 above, 1 and PR

3. $\vdash [at\ell_0 \wedge atm_{2,3}] \supset (y > 0)$ also by I_1 and PR

4. $\vdash at\ell_0 \supset \Diamond(y > 0)$ by T1, 2, 3, LOC and PR

Take now

$\varphi_2 : \ at\ell_0$

$\psi_2 : \ at\ell_1$

We check premises A to C in the EVNT rule with respect to the pair $\{\varphi_2, \psi_2\}$ taking $k = 1$. Clearly P always leads from φ_2 to $\varphi_2 \vee \psi_2$. The process P_1 always leads (when enabled) from φ_2 to ψ_2. Condition C is guaranteed by 4 above. We therefore conclude

5. $\vdash at\ell_0 \supset 0 \ at\ell_1$.

By a completely symmetric argument we can show that:

$\vdash atm_0 \supset 0 \ atm_1$. ∎

12. EXAMPLE 3: MUTUAL EXCLUSION

As a third example we consider a program that solves the mutual exclusion problem without semaphores:

```
                    $(y_1, y_2, t) := (false, false, 1)$

    $\ell_0$ : Noncritical Section                    $m_0$ : Noncritical Section
    $\ell_1 : y_1 :=$ true                            $m_1 : y_2 :=$ true
    $\ell_2 : t := 1$                                 $m_2 : t := 2$
    $\ell_3$ : if $y_2 = false$ then go to $\ell_5$   $m_3$ : if $y_1 = false$ then go to $m_5$
    $\ell_4$ : if $t = 1$ then go to $\ell_3$         $m_4$ : if $t = 2$ then go to $m_3$
    $\ell_5$ : Critical Section                       $m_5$ : Critical Section
    $\ell_6 : y_1 :=$ false                           $m_6 : y_2 := false$
    $\ell_7$: go to $\ell_0$                          $m_7$ : go to $m_0$

              $- P_1 -$                                       $- P_2 -$
```

For convenience we will abbreviate formulas $at\ell_i$ to ℓ_i.

The principle of operation of this program is that each process P_i has a variable y;, $i = 1, 2,$ which expresses the process's wish to enter its critical section. The variable y_i is set to *true* at ℓ_1 and m_1 and reset to *false* at ℓ_6 and m_6, respectively. In addition, each process leaves a signature in the common variable t. The process P_1 sets it to 1 at ℓ_2 and P_2 sets it to 2 at m_2. A process P_i may entcr its critical section only if either y_j = *false* (meaning that thc othcr process is not interested) or if t = j, for j $\neq i$. The latter case corresponds to both processes being interested in entering the critical section but P_j being thc *last* to pass through the signing instructions at (ℓ_2, m_2).

To formally prove that this program is correct we first prove sevcral invariance properties.

Lemma A:

$$\vdash y_1 \equiv \ell_{2..6}$$

Here $\ell_{2..6}$ stands for *at* $\ell_{2..6}$. Thus the lemma states that

$$y_1 = \textit{true} \quad \text{if and only if} \quad \pi_1 \in \{\ell_2, \ell_3, \ell_4, \ell_5, \ell_6\}.$$

Proof:

To prove the Lemrna we take

$$\varphi_1 : \ (y_1 \equiv \ell_{2..6})$$

and show that it is invariant under every transition, i.e., every transition leads from φ_1 to φ_1.

The only transitions that can affect the truth of φ_1 arc $\ell_1 \rightarrow \ell_2$ and $\ell_6 \rightarrow \ell_7$.

In $\ell_1 \rightarrow \ell_2$ both y_1 and *at* $\ell_{2..6}$ bccomc simultaneously true. Similarly in $\ell_6 \rightarrow \ell_7$ both y_1 and *at* $\ell_{2..6}$ become simultaneously false. Thus

1. $\vdash (y_1 \equiv \ell_{2..6}) \supset \bigcirc(y_1 \equiv \ell_{2..6})$ by TRNS
2. $\vdash \{at(\ell_0, m_0) \wedge [(y_1, y_2, t) = (\textit{false}, \textit{false}, 1)]\} \supset (y_1 \equiv \ell_{2..6})$
3. $\vdash \Box\ (y_t \equiv \ell_{2..6})$ by 1, 2 and TINV ∎

Lemma B:

$$\vdash y_2 \equiv m_{2..6}$$

The lcmma is proved by a symmetric argument.

Lemma C:

$$\vdash (t = 1) \vee (t = 2)$$

This lemma states that the only possible values of the variable t are **1** or 2.

Proof:

The Lemma is clearly provable by the IINV principle. Obviously, it is true initially since $t = 1$. The only transitions that modify the value of t set it either to 1 or to 2. Thus P always leads to a state satisfying $(t = 1) \vee (t = 2)$. ,

Lemma D:

$$\vdash \ell_{5,6} \supset [(\sim y_2) \vee (t = 2) \vee m_2]$$

Proof:

Let φ_2 stand for $\ell_{5,6} \supset [(\sim y_2) \vee (t = 2) \vee m_2]$.

It *is* clearly true ***initially since*** $\vdash \ell_0 \supset \sim\ell_{5,6}$. To show that every transition leads from φ_2 to φ_2, consider the only transitions that may falsify φ_2, i.e., that may possibly lead from φ_2 to $\sim\varphi_2$. Potentially they arc:

- $\ell_3 \rightarrow \ell_5$. This transition is possible only under $\sim y_2$ which makes

  $(\sim y_2) \vee (t = 2) \vee m_2$

  true.

- $\ell_4 \rightarrow \ell_5$. This is possible only when $t \neq 1$ which by Lemma C makes

  $(\sim y_2) \vee (t = 2) \vee m_2$

  again true.

The other transitions we should consider are transitions of P_2 while P_1 is already at $\ell_{5,6}$. The only ones to bc considered arc those which affect any of the variables in $\sim y_2 \vee (t = 2) \vee m_2$.

- $m_1 \rightarrow m_2$. Causes m_2 to become true.
- $m_2 \rightarrow m_3$. Causes t to be set to 2.
- $m_6 \rightarrow m_7$. Sets y_2 to *false*, making $\sim y_2$ true.

The lemma follows by the IINV principle. ▟

Lemma E:

$$\vdash m_{5,6} \supset [(\sim y_1) \vee (t = 1) \vee \ell_2]$$

The lemma is proved by a completely symmetric argument.

Theorem:

$$\vdash (\sim\ell_{5,6}) \vee (\sim m_{5,6})$$

This theorem proves the mutual exclusion of the processes.

Proof:

1. $\vdash (\ell_{5,6} \wedge m_{5,6}) \supset [((\sim y_2) \vee (t = 2) \vee m_2) \wedge ((\sim y_1) \vee (t = 1) \vee \ell_2)]$ by lemmas C, D and PR
2. $\vdash (\ell_{5,6} \wedge m_{5,6}) \supset [y_1 \wedge y_2 \wedge \sim\ell_2\ \mathrm{A}\ \sim m_2]$ by lemmas A, B, LOC and PR
3. $\vdash (\ell_{5,6}\ \mathrm{A}\ m_{5,6}) \supset [(\mathrm{t} = 1)\ \mathrm{A}\ (t = 2)]$ by 1, 2 and PR
4. $\vdash \sim(\ell_{5,6} \wedge m_{5,6})$ by the equality axiorns and PR, using the domain fact that $1 \neq 2$
5. $\vdash (\sim\ell_{5,6}) \vee (\sim m_{5,6})$ by PR ∎

Next we will prove accessibility. We will only prove:

Theorem:

$$\vdash at\,\ell_1 \supset \Diamond\, at\,\ell_5$$

The result for P_2 is completely symmetric.

Proof:

The proof will proceed by a sequence of statements most of which are proved by the EVNT rule in the version whose conclusion is $\varphi \supset 0\ \psi$. Simple passages justified by propositional temporal reasoning will not be fully presented and their omission is denoted by mentioning PTR in the justification clause.

1. $\vdash (\ell_4\ \mathrm{A}\ m_{3,4}\ \mathrm{A}\ t = 2) \supset 0\ \ell_5$ by EVNT with $k = 1$, using lemma A
2. $\vdash (\ell_3\ \mathrm{A}\ m_{3,4}\ \mathrm{A}\ t = 2) \supset \Diamond(\ell_4\ \mathrm{A}\ m_{3,4}\ \mathrm{A}\ t = 2)$ by EVNT with $k = 2$, using lemmas A, B
3. $\vdash (\ell_3\ \mathrm{A}\ m_{3,4}\ \mathrm{A}\ t = 2) \supset 0\ \ell_5$ by 2, 1 and *OC*
4. $\vdash (\ell_{3,4}\ \mathrm{A}\ m_{3,4}\ \mathrm{A}\ t = 2) \supset 0\ \ell_5$ by 1, 3 and PR
5. $\vdash (\ell_{3,4}\ \mathrm{A}\ m_2) \supset \Diamond[\ell_5 \vee (\ell_{3,4}\ \mathrm{A}\ m_{3,4}\ \mathrm{A}\ t = 2)]$ by EVNT with $k = 2$

6. $\vdash (\ell_{3,4} \wedge m_2) \supset \Diamond \ell_5$ by 4, 5 and PTR

7. $\vdash (\ell_{3,4} \wedge m_1) \supset \Diamond[\ell_5 \vee (\ell_{3,4} \wedge m_2)]$ by EVNT with $k = 2$

8. $\vdash (\ell_{3,4} \wedge m_1) \supset \Diamond \ell_5$ by 7, 6 and PTR

9. $\vdash (\ell_3 \wedge m_0) \supset \Diamond[\ell_5 \vee (\ell_{3,4} \wedge m_1)]$ by EVNT with $k = 1$

10. t- $(\ell_3$ A $m_0) \supset$ 0 ℓ_5 by 9, 8 and PTR

11. $\vdash (\ell_4 \wedge m_0) \supset \Diamond[\ell_5 \vee (\ell_{3,4} \wedge m_1) \vee (\ell_3 \wedge m_0)]$ by EVNT with $k = 1$

12. I- $(\ell_4$ A $m_0) \supset$ 0 ℓ_5 by 11, 8, 10 and PTR

13. $\vdash (\ell_{3,4} \wedge m_0) \supset \Diamond \ell_5$ by 10, 12 and PR

14. I - $(\ell_{3,4} \wedge m_7) \supset \Diamond[\ell_5 \vee (\ell_{3,4} \wedge m_0)]$ by EVNT with $k = 2$

15. $\vdash (\ell_{3,4} \wedge m_7) \supset \Diamond \ell_5$ by 14, 13 and PTR

16. $\vdash (\ell_{3,4} \wedge m_6) \supset \Diamond(\ell_{3,4} \wedge m_7)$ by EVNT with $k = 2$ and lemma E

17. $\vdash (\ell_{3,4} \wedge m_6) \supset \Diamond \ell_5$ by 16, 15 and PTR

18. $\vdash (\ell_{3,4} \wedge m_5) \supset \Diamond(\ell_{3,4} \wedge m_6)$ by EVNT with $k = 2$ and lemma E

19. $\vdash (\ell_{3,4} \wedge m_5) \supset \Diamond \ell_5$ by 18, 17 and PTR

20. $\vdash (\ell_{3,4}$ A m_4 A $t = 1) \supset \Diamond(\ell_{3,4}$ A $m_5)$ by EVNT with $k = 2$ and lemma A

21. $\vdash (\ell_{3,4}$ A m_4 A $t = 1) \supset$ 0 ℓ_5 by 20, 19 and PTR

22. $\vdash (\ell_{3,4}$ A m_3 A $t = 1) \supset \Diamond(\ell_{3,4}$ A m_4 A $t = 1)$
by EVNT with $k = 2$ and lemma A

23. I- $(\ell_{3,4}$ A m_3 A $t = 1) \supset$ 0 ℓ_5 by 22, 21 and PTR

24. $\vdash (\ell_{3,4}$ A $m_{3,4}$ A $t = 1) \supset$ 0 ℓ_5 by 21, 23 and PR

25. $\vdash (\ell_{3,4} \wedge m_{3,4}) \supset \Diamond \ell_5$ by 4, 24, lemma C and PR

We may summarize now as follows:

26. $\vdash \ell_{3,4} \supset [\ell_{3,4} \wedge (m_0 \vee m_1 \vee m_2 \vee m_3 \vee m_4 \vee m_5 \vee m_6 \vee m_7)]$
by LOC

27. I- $\ell_{3,4} \supset$ 0 ℓ_5 by 26, 13, 8, 6, 25, 19, 17, 15 and PTR

28. $\vdash \ell_2 \supset \Diamond \ell_{3,4}$ by EVNT with $k = 1$

29. I- $\ell_2 \supset$ 0 ℓ_5 by 27, 28 and OC

30. I- $\ell_1 \supset \Diamond \ell_2$ by EVNT with $k = 1$

31. $\vdash \ell_1 \supset$ 0 ℓ_5 by 29, 30 and $\Diamond$C

F. COMPACT PROOF PRINCIPLES

In the preceding sections we introduced a comprehensive proof system for proving arbitrary temporal properties of concurrent programs. However, as demonstrated in the last examples a fully formal proof tends to be rather lengthy and sometimes tedious to follow. Consequently we will next discuss shorter and more compact representations of proofs and corresponding compact proof principles. All Lhcsc principles can be derived in the basic proof system presented above. Consequently, a proof according Lo these principles can always be mechanically expanded into a more detailed proof using just thc basic axioms. We will discuss the threc main classes of properties one may wish to prove about programs, namely: invariance, liveness and precedence properties.

13. THE INVARIANCE PRINCIPLE

The IINV principle does not significantly simplify formal proofs. Most of the needed work in applying the IINV principle is in establishing the premise that the program P leads from φ to φ. Several heuristics or meta-rulcs can bc suggested in order to rcducc the number of transitions that have to be checked, which in the worst case is proportional to the size of the program. For example:

a) Only transitions that modify variables on which φ depends should bc checked.

b) Assume that φ has the form $\varphi = \varphi_1$ V φ_2 (similarly for implication), and that some variables $y_1, \ldots, y_m$ appear only in φ_1. Then, in checking transitions that only modify Lhcsc variables, it is sufficient to check transitions that may falsify φ_1 and one may assume in checking them Lhat φ_2 = *false*.

c) Assume that an invariance χ has already been established beforc. Let

$$[\varphi \wedge \chi] \supset (\sim at\,\ell)$$

for some location ℓ. Then no transitions of the form $\ell \rightarrow \ell'$ need ever be considered in showing that P leads from φ to ψ.

A simple generalization of the IINV rule is given by:

Generalized ***Invariance Rule*** -- GINV

A . $\vdash \varphi \supset \psi$

B. I- [at& A $\overline{y}$ = g(C)] $\supset \varphi$

C. I- P leads from φ to φ

$$\vdash \Box \psi$$

Certainly premises B and C establish $\vdash \Box\ \varphi$ according to IINV, from which by premise A and the $\Box\ \Box$ rulc, t- $\Box\ \psi$ follows.

The advantage of the GINV principle is that no additional temporal reasoning is required and the rule can be proved complete by itself. By this we mean that, given a program P, any state property ψ which is invariant for all executions of P can bc proven invariant by a single application of the GTNV rule and no additional temporal reasoning.

Theorem:

The GINV rulc is complete for proving invariance propcrties.

Proof:

Let $\psi = \psi(\bar{x}; \bar{\pi}; \bar{y})$ be a state property, possibly dependent on the input variables $\bar{x}$. We define a state $s = \langle\bar{\ell}; \bar{\eta}\rangle$ to be $\bar{\xi}$-accessible in P if there exists a segment of some computation initialized with $\bar{x} = \bar{\xi}$ that reaches s, i.e.,

$$\langle\bar{\ell}_0;\ g(\bar{\xi})\rangle \rightarrow \ . \ . \ . \rightarrow \langle\bar{\ell};\ \bar{\eta}\rangle.$$

Define the predicate $\varphi = \varphi(\bar{x}; \bar{\pi}; \bar{y})$ by:

$$\varphi(\bar{\xi}; \bar{\ell}; \bar{\eta}) = \textbf{\textit{true}} \quad \Leftrightarrow \quad \langle\bar{\ell};\ \bar{\eta}\rangle \text{ is T-accessible.}$$

Thus, φ characterizes all the states that are $\bar{x}$-acccssiblc. We will show that the predicate φ so defined satisfies, togcthcr with ψ, all thc prcrniscs required by the rule GJNV.

Consider prcmisc A. Since ψ is invariantly truc in all computations of P it must be true for every accessible state $\langle\bar{\ell}; \bar{\eta}\rangle$. Consequently

$$\varphi(\bar{\xi};\bar{\ell};\bar{\eta}) \ \supset \ \psi(\xi;\ell;\bar{\eta});$$

when generalized to arbitrary $\bar{\xi}$, $\bar{\ell}$ and $\bar{\eta}$ Lhis implies

$$\models \varphi \ \supset \ \psi.$$

Since we assume that thc underlying domain theory is adequate for proving all classically sound formulas this implies

$$\vdash \varphi \ \supset \ \psi.$$

Consider now premise B. Since every initial state is by definition accessible we certainly have

$$\models \varphi\ (\bar{x};\ \bar{\ell}_0;\ \mathrm{g(z)}).$$

Again by completeness of our domain part with respect to classical formulas, this leads Lo

$$\vdash [at\,\bar{\ell}_0 \ \mathrm{A} \quad y = \quad g(\bar{x})] \supset \varphi(\bar{x};\ \bar{\pi};\ \bar{y}).$$

Finally, considcr premise C. Clearly every transition in P leads from an z-accessible statc to another $\bar{x}$-acccssiblc stale. Consequcntly

$$\models P \text{ leads from } \varphi \text{ to } \varphi.$$

From this prcmisc C follows by completeness of the domain part. ∎

In the preceding theorem we have only shown the existence of an appropriate state predicate φ. Wc have not discussed the question of the exact formal language in which such a predicate can be expressed. However, assuming that our domain contains thc integers or some isomorphic structure, and using a first-order language, it is not difficult to show that thc statement:

> "There exists a finite computation of P leading from $\langle \bar{\ell}_0; g(\bar{\xi})\rangle$ to $\langle \bar{\ell}; \bar{\eta}\rangle$"

can bc Gödel-encoded into a first-order statement ovcr the integers.

14. LIVENESS PRINCIPLES

As a typical example of a detailed proof of liveness properties wc may rccxaminc the proof of accessibility for the mutual exclusion program (Example 3). The structure of such a proof proceeds through a chain of events characterized by statc assertions. Lct the eventuality to be provcd bc $\varphi \supset \Diamond \psi$ where both φ and ψ arc state properties. We may rcgard $\psi = \varphi_0$ as being Lhc last assertion in the chain. Then we identify an assertion φ_1 such that by a single application of the EVNT principle we can prove

$$\vdash \varphi_1 \supset \Diamond \psi.$$

In the example considered we have

$$\psi : \ell_5$$

$$\varphi_1 : \ell_4\ A\ m_{3,4}\ A\ (t = 2).$$

Next, we idcntify an assertion φ_2 such that by a single application of thc EVNT principle we can prove

$$\vdash \varphi_2 \supset \Diamond(\varphi_1 \vee \psi).$$

In the general stcp, wc identify an assertion φ_i such that by a single application of thc EVNT principle wc can provc

$$\vdash \varphi_i \supset \Diamond\big(\bigvee_{j<i} \varphi_j\big).$$

Finally wc havc to prove $\varphi \supset (\bigvee_{i=0}^{r} \varphi_i)$ where $\varphi_1, \ldots, \varphi_r$ is the chain of assertions constructcd. Wc may summarize this proof pattern by Lhc following proof principle:

The huin Reasoning Proof Principle --- CHAIN

Let $\varphi_0, \varphi_1, \ldots, \varphi_r$ be a sequence of state properties satisfying the following rcqui rements:

A. $\vdash P$ leads from φ_i Lo $\bigvee_{j \le i} \varphi_j$ for $i = 1, \ldots, r$.

B. For every $i > 0$ there exists a $k = k_i$ such that:

$$\vdash P_k \text{ leads from } \varphi_i \text{ to } \bigvee_{j<i} \varphi_j$$

C. For $i > 0$ and $k = k_i$ as above:

$$\vdash \varphi_i \supset 0 \;[(\bigvee_{j<i} \varphi_j) \vee \mathit{Enabled}(P_k)]$$

$$\vdash (\bigvee_{i=o}^{r} \varphi_i) \supset (\bigvee_{i=1}^{r} \varphi_i) \mathcal{U} \varphi_0$$

Proof:

To justify this principle we will prove by induction on n, $n = 0, 1, \ldots, r$, that

$$\vdash (\bigvee_{i=o}^{n} \varphi_i) \supset (\bigvee_{i=1}^{n} \varphi_i) \mathcal{U} \varphi_0.$$

For n = 0 we have $\vdash \varphi_0 \supset \varphi_0$ from which trivially follows by axiom Λ9

$$\vdash \varphi_0 \supset (\mathit{false}\; \mathcal{U}\; \varphi_0).$$

Note that wc interpret an empty disjunction as false.

We assume that the statement above has been proved for certain n and we attempt to prove it for $n + 1$.

Consider the EVNT rule with $\varphi = \varphi_{n+1}$, $\psi = (\bigvee_{i=o}^{n} \varphi_i)$. By premise Λ of CHAIN we obtain that P leads from $\varphi_{n+1} = \varphi$ to

$$(\bigvee_{j \le n+1} \varphi_j) = (\varphi_{n+1} \vee (\bigvee_{j \le n} \varphi_j)) = (\varphi \vee \psi).$$

This provides premise Λ of EVNT. Let $k = k_{n+1}$. Then by premise B of CHAIN, P_k leads from $\varphi_{n+1} = \varphi$ to $(\bigvee_{j<n+1} \varphi_j) = \psi$.Similarly, premise C of CHAIN yields that

1. $\vdash \varphi \supset \Diamond(\psi \vee \mathit{Enabled}(P_k))$.

By the EVNT rule it follows that

$$2.\quad \vdash \varphi \supset \varphi\mathcal{U}\psi$$

or

$$3.\ \vdash \varphi_{n+1} \supset \varphi_{n+1}\mathcal{U}(\bigvee_{i=0}^{n} \varphi_i).$$

By the induction hypothesis and the UU rule this yields

$$4.\ \vdash \varphi_{n+1} \supset \varphi_{n+1}\mathcal{U}((\bigvee_{i=1}^{n} \varphi_i)\mathcal{U}\varphi_0).$$

Again by the induction hypothesis using part of A9, $w_2 \supset w_1\mathcal{U}w_2$, wc can obtain

$$5.\quad \vdash (\bigvee_{i=0}^{n} \varphi_i) \supset \varphi_{n+1}\mathcal{U}((\bigvee_{i=1}^{n} \varphi_i)\mathcal{U}\varphi_0).$$

Combining this with 4 above yields

$$6. \vdash (\bigvee_{i=0}^{n+1} \varphi_i) \supset \varphi_{n+1}\ \mathcal{U}((\bigvee_{i=1}^{n} \varphi_i)\mathcal{U}\varphi_0).$$

By T38, $p\mathcal{U}(q\mathcal{U}r) \supset (p \vee q)\mathcal{U}r$, wc can reduce the nesting depth of the U operator to get:

$$7. \vdash (\bigvee_{i=0}^{n+1} \text{Pi}) \supset ((\bigvee_{i=1}^{n+1} \varphi_i)\mathcal{U}\varphi_0)$$

as needed.

Taking $n = r$ concludes the proof of the principle. ∎

In presenting a proof according to the chain-reasoning principle it is usually sufficient to identify $\varphi_0, \varphi_1, \ldots, \varphi_r$ and for each i to point out the "helpful" process $P_k = P_{k_i}$. It can be left to the reader to verify that premises A to C arc satisfied for each $i = 1, 2, \ldots, r$.

We prefer to present such proofs in the form of a diagram. Consider a diagram consisting of nodes that correspond Lo the assertions $\varphi_0, \varphi_1, \ldots, \varphi_r$. For each transition affected by some process P_j, that leads frorn a state s satisfying φ_i Lo a state s' satisfying φ_ℓ, $\ell < i$, wc draw an edge from the node φ_i to the node φ_ℓ and label it by P_j, the name of the responsible process. All edges corresponding to the helpful process $P_k = P_{k_i}$ are drawn as double arrows. We do not explicitly draw edges corresponding to transitions frorn φ_i back to itself. However it is assumed that such edges may exist for all but the helpful process for φ_i.

As an example wc present a diagram form of the proof of accessibility for the Mutual Exclusion program. Tt is given in Fig. 1. in constructing such a proof wc rnay freely use any invariants previously derived.

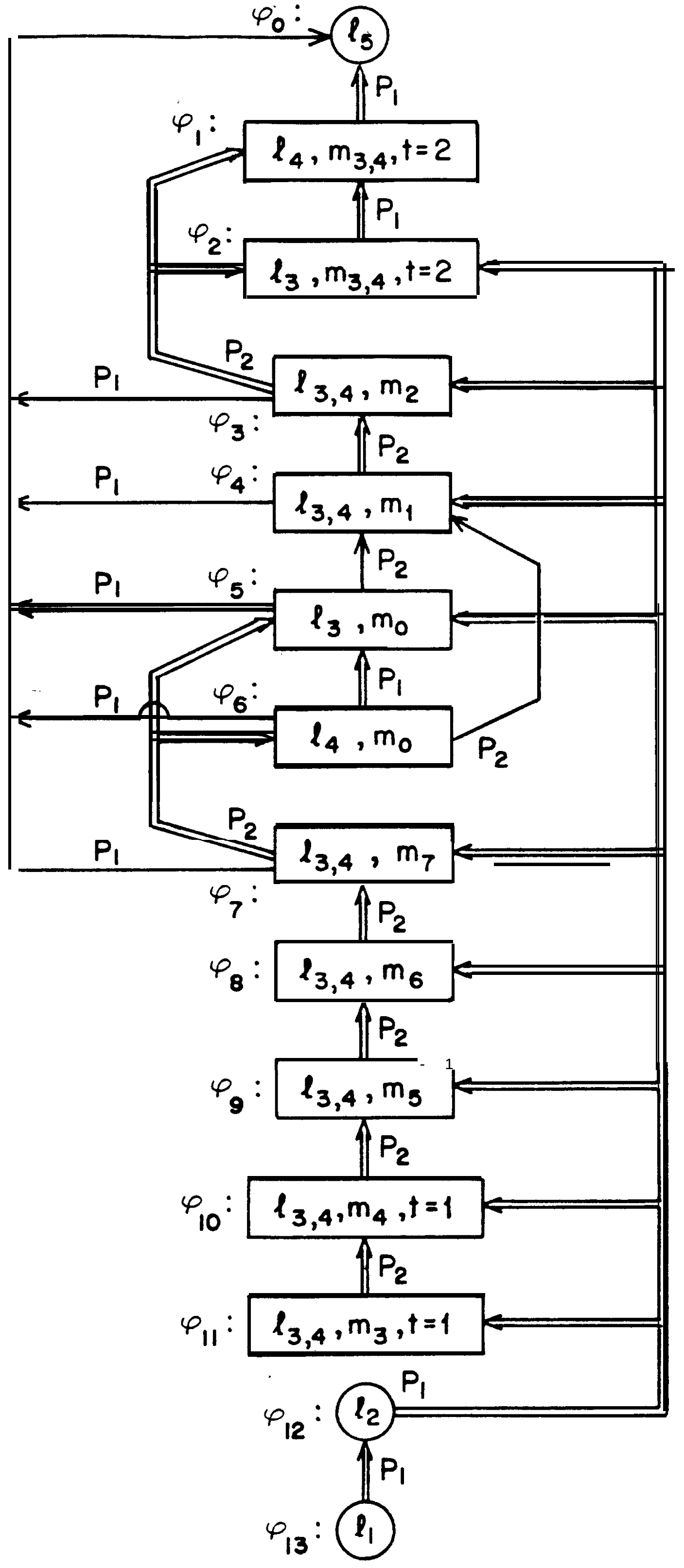
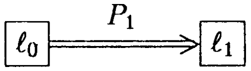

φ_0: ℓ_5
P_1
φ_1: $\ell_4, m_{3,4}, t=2$
P_1
φ_2: $\ell_3, m_{3,4}, t=2$
P_2
P_1
$\ell_{3,4}, m_2$
φ_3:
P_2
P_1
φ_4: $\ell_{3,4}, m_1$
P_2
P_1
φ_5: ℓ_3, m_0
P_1
P_1
φ_6: ℓ_4, m_0
P_2
P_2
P_1
$\ell_{3,4}, m_7$
φ_7:
P_2
φ_8: $\ell_{3,4}, m_6$
P_2
φ_9: $\ell_{3,4}, m_5$
P_2
φ_{10}: $\ell_{3,4}, m_4, t=1$
P_2
φ_{11}: $\ell_{3,4}, m_3, t=1$
P_1
φ_{12}: ℓ_2
P_1
φ_{13}: ℓ_1

Fig. 1. Proof Diagram for the Mutual Exclusion Program

In this program, and typically in all non-terminating programs that have no semaphore instructions, wc do not have to check premisc C of thc CIIAIN or EVNT rule. This is bccause in non-terminating programs without semaphores every process is continuously enabled and therefore condition C is automatically satisfied.

In contrast lct us consider thc proof of acccssibility for example 2 – a program with semaphores. IIere wc want to prove $\ell_0 \supset \Diamond\ \ell_1$. Thc main diagram here is vcry simple:

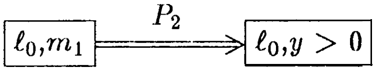


It denote; a singlc application of the EVNT rulc with $\varphi : at\,\ell_0$ and $\psi : at\,\ell_1$ with $P_k = P_1$ being thc helpful process.

IIowever, in ordcr to justify premisc C, which is not trivial in this case, we have to prove

$$\vdash \ell_0 \supset \Diamond(\ell_1\ v\ y > 0).$$

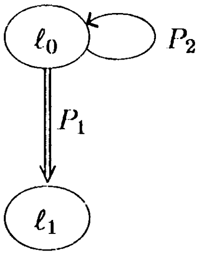
For this we havc to consider P_2's position. If P_2 is at m_0 or m_2 then $y = 1$ by thc invariant I_1 proved above, Thc only othcr casc is when P_2 is at m_1 where by a single application of thc EVNT rulc it will cventually move to m_2 producing a positive value of y. This may bc represented by a sccondary diagram:



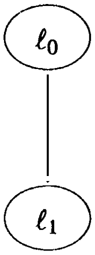
Thc diagram rcprcsentation of a proof according to the CHAIN principle is very similar to thc proof latticcs introduccd in [OL] as a concise presentation of a proof of a liveness property. A superficial diffcrcncc is that thcy ***choose*** to rcprcsent as cdgcs the ***consequences*** of the EVNT rulc, while in our rcprcsentation cdgcs stand for the premises of thc EVNT rulc which arc also the prcmiscs Lo thc CIIAIN rule. To illustrate this diffcrcncc, consider the following trivial program:

$\ell_0 :\ \ y := y$	$m_0 :$ ***go to*** m_0
$\ell_1 :$	
– P_1 –	– P_2 –

The livcness properly to bc proved is $\ell_0 \supset$ 0 L$_1$. Bclow are diagram representations of the CIIAIN principlc and a proof lattice according to [OL].



CT LAIN Diagram



Proof Lattice

As wc scc, thc CIIAIN diagram contains a sclf-cdgc, labcllcd by P_2 (this timc drawn cxplicitly) and a helpful cdgc labcllcd by P_1. The process P_1 is guaranteed to get us to ℓ_1. As a conscqucncc of this, by the EVNT rule, $\ell_0 \supset 0\ \ell_1$. This conclusion is represented in the proof lattice by a single edge from ℓ_0 to ℓ_1. Thus, the different choices of representation lead to the following minor syntactical differences between CIIAIN diagrams and proof lattices:

(a) I'roof lattices are acyclic, whereas CIIAIN diagrams are only weakly acyclic, i.e., may contain self-loops.

(b) In CIIAIN diagrams, edges arc labelled by the processes responsible for the transition. Special identification is provided for edges traversed by the helpful process. In proof lattices, we no longer care about the identities of the processes since progress along the lattice has already been established.

However these differences are minor and a simple procedure for translation between CIIAIN diagrams and proof lattices exists. The important part in both is the identification of the intermediate assertions that are represented as nodes. In constructing a proof, this is usually the creative and most demanding process. Both graph presentations provide a natural and intuitive rcprcscntation of these assertions and the precedence relations between them.

The chain-reasoning principle assumed a finite number of links in the chain. It is quite adequate for finite-state programs, i.e., programs whose variables range over finite domains. However, once WC consider programs over the integers it is no longer sufficient Lo consider only finitely many assertions. In fact, sets of--assertions of quite high cardinality are needed. The obvious generalization of a finite set of assertions $\{\varphi_i \mid i = 0, \ldots, r\}$ is to consider a single assertion $\varphi(\alpha)$, parametrized by a parameter α taken from a well-founded ordered set (A, $\prec$). Obviously, the most important property of our chain of assertion is that program transitions eventually lead from φ_i to.φ_j with $j < i$. This property can also be stated for an arbitrary well-founded ordering. Thus a natural generalization of the chain reasoning rule is the following:

The Well Founded *Liveness* ***Principle*** — WELL

Let ***(A,*** $\prec$) be a ***well-founded set.*** Let $\varphi(\alpha) = \varphi(\alpha; \overline{x}; \overline{\pi}; \overline{y})$ be a parametrized state formula.
Let $h : A \rightarrow [1 \,.\,.\, k]$ be a helpfulness function identifying for each $\alpha \in A$ the helpful process $P_{h(\alpha)}$ for states in $\varphi(\alpha)$.

A. $\vdash P$ lends' from $\varphi(\alpha)$ to $\psi \vee (\exists\beta \preceq$ a . $\varphi(\beta))$

B. $\vdash P_{h(\alpha)}$ leads from $\varphi(\alpha)$ to $\psi \vee (\exists\beta \prec \alpha\ .\ \varphi(\beta))$

c . $\vdash$ p (a) $\supset \Diamond[\psi$ v $(\exists\beta \prec \alpha\ .\ \varphi(\beta))$ v $Enabled(P_{h(\alpha)})]$

$\vdash (\exists\alpha\ .\ \varphi(\alpha)) \supset (\exists\alpha\ .\ \varphi(\alpha))\,\mathcal{U}\psi$

A justification of this rule can again be conducted, based on induction. Now, however, induction over arbitrary well-founded sets is required.

15. EXAMPLE 4: BINOMIAL COEFFICIENT

As an example for the application of the WELL principle, we consider the following program that computes the binomial coefficient $\binom{n}{k}$ for inputs $0 \le k \le n$.

```
                    $(y_1, y_2, y_3, y_4) := (n, 0, 1, 1)$

$\ell_7$ : if $y_1 = (n - k)$ then go to $\ell_1$
$\ell_6$ : request($y_4$)
$\ell_5$ : $t_1 := y_3 \cdot y_1$
$\ell_4$ : $y_3 := t_1$
$\ell_3$ : release($y_4$)
$\ell_2$ : $y_1 := y_1 - 1$
$\ell_8$ : go to $\ell_7$
$\ell_1$ : halt

                    – $P_1$ –

$m_3$ : if $y_2 = k$ then go to $m_1$
$m_2$ : $y_2 := y_2 + 1$
$m_9$ : loop until $y_1 + y_2 \le n$
$m_8$ : request($y_4$)
$m_7$ : $t_2 := y_3 / y_2$
$m_6$ : $y_3 := t_2$
$m_5$ : release($y_4$)
$m_4$ : go to $m_3$
$m_1$ : halt

                    – $P_2$ –
```

The labelling scheme of the program has been constructed in a way that simplifies the expression of the assertion $\varphi(\alpha)$.

The computation of this program is based on the formula:

$$\binom{n}{k} = \frac{n \cdot (n-1) \cdots (n-k+1)}{1 \cdot 2 \cdots k}.$$

The values of y_1, i.e., $n, n-1, \ldots, n-k+1$, are used to compute the numerator in P_1, and the values of y_2, i.e., $1, 2, \ldots, k$, are used to compute tho denominator. The process P_1 multiplies $n \cdot (n-1) \cdots (n-k+1)$ into y_3 while P_2 divides y_3 by $1 \cdot 2 \cdots k$.

The instruction

$$m_9 : \textbf{loop until } y_1 + y_2 \le n$$

guarantees even divisibility of y_3 by y_2. It synchronizes P_2's operation with that of P_1 to ensure that y_3 is divided by i only after $(n - i + 1)$ has already been multiplied into it. We rely here on the mathematical theorem that the product of i consecutive integers $n \cdot (n-1) \cdots (n-i+1)$ is always divisible by $i!$ (the quotient actually being the integer $\binom{n}{i}$).

The critical sections $\ell_{3..5}$ and $m_{5..7}$ are mutually protected by the semaphore variable y_4. This protection ensures that y_3 is not updated by P_2 between, say, the computation of $y_3 \cdot y_1$ and the assignment of this value to y_3. Without this protection, the updated value might, have been overwritten by P_1.

Wc start by establishing somc invariant properties of this program.

$$I_1: \vdash (at\ell_{3..5} + atm_{5..7} + y_4 = 1) \wedge (y_4 \geq 0).$$

'This is the usual semaphore invariant. It can bc provcn by observing that initially this sum equals 1, and then by considering all possible transitions. For example, the $\ell_6 \rightarrow \ell_5$ transition changcs $at\ell_{3..5}$ from 0 (*false*) to 1 (*true*), and also decrements y_4 by 1, leaving however the sum constant. From I_1 we can deduce mutual exclusion of the critical scctions, i.e.,

$$\vdash (\sim\ell_{3..5}) \vee (\sim m_{5..7}).$$

As a consequcncc of this we can establish:

$$I_2: \vdash (\ell_4 \supset t_1 = y_3 \cdot y_1) \wedge (m_6 \supset t_2 = y_3/y_2).$$

This holds due to the impossibility of interference by P_2 while P_1 is at ℓ_4.

$$I_3: \vdash (n - k + at\ell_{2..6}) \leq y_1 \leq n.$$

This invariance states that y_1 always lics betwccn n- k and n. When P_1 is at $\ell_{2..6}$, $y_1 > n - k$, whereas P_1 is at other locations, $y_1 \geq n - k$. To verify 4 we need only consider the transitions:

- $\ell_7 \rightarrow \ell_6$ which maintains $n - k < y_1 \leq n$, assuming it was previously known that $n - k \leq y_1 \leq n$.
- $\ell_2 \rightarrow \ell_8$ which results in $n - k \leq y_1 - 1 \leq n$ from $n - k < y_1 \leq n$.

$$I_4: \vdash 0 \leq y_2 \leq (k - atm_2).$$

This invariance bounds Lhe rangc of y_2. We nccd consider the transitions $m_3 \rightarrow m_2$ and $m_2 \rightarrow m_4$ which can be shown to maintain I_4.

$$I_5: \vdash atm_{7..8} \supset (y_1 + y_2) \leq n.$$

Hcre we should consider two transitions:

- $m_9 \rightarrow m_8$ which is possible only if currently $y_1 + y_2 \leq n$.
- $\ell_2 \rightarrow \ell_8$ is the only transition modifying y_1. Howcvcr since it decrements y_1 it ccrtainly preserves $y_1 + y_2 \leq$ n.

Lct us define the following virtual variables:

- $y_1^* = \textit{if } at\ell_{2,3} \textit{ then } y_1 - 1 \textit{ else } y_1$

  $y_2^* = \textit{if } atm_{6..9} \textit{ then } y_2 - 1 \textit{ else } y_2$

These variables are roughly equal to y_1 and y_2 respectively and differ from them by 1 in certain ranges.

$$I_6 : \quad \text{I-} \quad y_3 = [n \,.\, (n - 1) \ldots (y_1^* + 1)]/[1 \,.\, 2 \cdot . \cdot y_2^*].$$

To verify this invariant wc have to check the transitions $\ell_4 \rightarrow \ell_3$, $m_6 \rightarrow m_5$. Making use of I_2, they can be shown Lo maintain I_6.

$$I_7 : \quad \text{I-} \quad [at\ell_1 \supset y_1 = (n - k)] \ \text{A} \ [atm_1 \supset (y_2 = k)].$$

Using I_6, I_7 and the definition of y_1^*, y_2^* we obtain partial correctness of this program, namely

$$\vdash (at\ell_1 \wedge atm_1) \supset [y_3 = \tbinom{n}{k}].$$

To prove termination we will use the WELL rule in order lo establish $\vdash$ *0(* $at\ell_1$ *A* atm_1). *As* the well-founded domain we take

$$(\text{A}, \prec) = (\text{N x } N \text{ x N}, \prec_{lex}).$$

That is, the set of triplets of nonnegative integers ordered by lexicographic ordering. This ordering defines $(m_1, m_2,$ m3$)$ $\prec$ (n_1, n_2, n_3) iff for the lowest i, $i = 1,2,3$ such that $m_i \neq$ n;, *m; < n;.*

For our goal assertion we take ψ : $at\ell_1$ A atm_1. The parameterized assertion is given by:

$$\varphi(\alpha; \ell_i, m_j; y_1, y_2) : (y_1 + k - y_2, j, i) = \alpha.$$

The helpfulness function is given by:

$$h(a) = h(r, j, i) = (\textit{if } i = 1 \textit{ then } 2 \textit{ else } 1).$$

Thus as long as the first process P_1 has not terminated wc rely on P_1 to bc the helpful process. Once it has terminated, we take P_2 to be the helpful process.

We have to show that all the three premises of the WELL rule arc satisfied.

Consider first premise A. Wc have lo show that every transition of P leads to $\varphi(\beta)$ with $\beta \preceq \alpha$ if ψ is not already satisfied. By simple inspection of all the possible transitions wc find that they all lead from $\langle \ell_i, m_j \rangle$ to $\langle \ell_{i'}, m_{j'} \rangle$ such that either $i' < i$ or $j' < j$ except for the following transitions:

- $\ell_2 \rightarrow \ell_8$. But this transition decrements y_1 producing a strict decrease in y_1 + *k-* y_2 which is the first component in α.
- $m_2 \rightarrow m_9$. In a similar way Lhis transition increments y_2, leading to a decrease in $y_1 + k - y_2$.
- $m_9 \rightarrow m_9$. This transition leaves α at the same value.

. Consider now premise B. As we have shown above, all transitions provide a strict decrease in α. The only exception is $m_9 \rightarrow m_9$. However this is a &transition which is considered helpful only when P_1 is at ℓ_1. By I_7, at this point y_1 = (n − *k)* so that in view of I_4, $y_1 + y_2 \leq k$ and hence the only transition possible from m_9 is $m_9 \rightarrow m_8$.

To show premise C we have to prove that P_h is always eventually enabled. Consider first the case that h = 1. The only location in which it is not immediately enabled is when P_1 is at ℓ_6 while P_2 is at $m_{5..7}$ (in view of I_1). However by simple chain reasoning it is obvious that in such a case, P_2 will certainly reach m_4 in which y_4 becomes positive and P_1 enabled.

The case h = 2 is even simpler because it is only considered when P_1 is at ℓ_1. Consequently, even when P_2 is at m_8, which may potentially raise some problems, we have in view of I_1 and $at\ell_1$ that $y_4 > 0$ and P_2 is enabled.

Thus we conclude that $\psi : at\ell_1 \wedge atm_1$ must eventually be realized and therefore the program must terminate.

16. PRECEDENCE PROPERTIES

The next class of properties we will consider and provide proof principles for is that of precedence properties. These are properties, usually needing the $\mathcal{U}$ operator for their expression, which ensure that some event precedes another event, or that a certain event will not happen until another event happens first. In view of the fact that the basic FAIR and EVNT rules did actually provide a conclusion containing the $\mathcal{U}$ operator, they may be naturally utilized to form precedence proof principles which are generalizations of the corresponding liveness principles.

In the following we will often consider nested ***until*** expressions in which the nesting always occurs in the second argument. We therefore adopt the convention of representing the nested formula:

$$\varphi_n \,\mathcal{U} \left(\varphi_{n-1} \,\mathcal{U} \left(\ldots (\varphi_1 \,\mathcal{U}\, \varphi_0) \ldots \right)\right)$$

by:

$$\varphi_n \,\mathcal{U}\, \varphi_{n-1} \,\mathcal{U} \ldots \varphi_1 \,\mathcal{U}\, \varphi_0.$$

The semantic meaning of this formula is that, starting from the present there is going to be a period in which φ_n continuously holds, followed by another period in which φ_{n-1} continuously holds, . . . , followed by a period in which φ_1 continuously holds, until finally φ_0 occurs. Any of these periods may be empty, but the occurrence of φ_0 is guaranteed.

Let us consider first the proper generalization of the CHAIN rule in which we assume a ***finite*** chain of assertions $\varphi_r, \varphi_{r-1}, \ldots, \varphi_1$ leading to the goal $\psi = \varphi_0$.

Let $0 < p_1 < p_2 < \ldots < p_s = r$ be a partition of the index range into s contiguous segments. Then we may formulate the following chain principle for precedence properties:

***The Chain Rule for Precedence Properties* — P-CHAIN**

Let $\varphi_0, \varphi_1, \ldots, \varphi_r$ be a sequence of state assertions, and $0 = p_0 < p_1 < p_2 < \ldots < p_s = r$ a partition of $[1 \,.\,.\, r]$.

A. I- P leads from φ_i to $(\bigvee_{j \leq i} \varphi_j)$ for $i = 1, \ldots, r$.

B. For every $i > 0$ there exists a $k = k_i$ such that:

$$\vdash P_k \text{ leads from } \varphi_i \text{ to } (\bigvee_{j<i} \varphi_j)$$

C. For i > 0 and $k = k_i$ as above:

$$\vdash \varphi \supset 0 \left[(\bigvee_{j<i} \varphi_j) \vee Enabled(P_k) \right]$$

$$\vdash (\bigvee_{i=0}^{r} \varphi_i) \supset (\psi_s \, \mathcal{U} \, \psi_{s-1} \ldots \psi_1 \, \mathcal{U} \, \varphi_0)$$

where

$$\psi_\ell \text{ is } \bigvee_{p_{\ell-1} < j \leq p_\ell} \varphi_j \quad \text{for } \ell = 1, \ldots, s.$$

The conclusion states that starting at a state that satisfies one of the φ_i, $i = 0, \ldots. r,$ *we* are guaranteed to have a period in which $(\bigvee_{j=p_{s-1}+1}^{p_s} \varphi_j)$continuously holds, followed by a period in which $(\bigvee_{j=p_{s-2}+1}^{p_{s-1}} \varphi_j)$ continuously holds, etc., until φ_0 is finally realized. Any of these periods may be empty.

Proof:

To justify the soundness of this conclusion we will first prove it for the most refined partition possible, namely:

$$(\bigvee_{i=0} \varphi_i) \supset (\varphi_r \, \mathcal{U} \, \varphi_{r-1} \, \mathcal{U} \, \varphi_{r-2} \, \mathcal{U} \cdot \ldots \varphi_1 \, \mathcal{U} \, \varphi_0).$$

This is proved in a way similar to the justification of the corresponding liveness principle. We show, by induction on n, $n = 0, 1, \ldots, r$; that

$$\vdash (\bigvee_{i=0}^{n} \text{Pi}) \supset (\varphi_n \, \mathcal{U} \, \varphi_{n-1} \, \mathcal{U} \ldots \varphi_1 \, \mathcal{U} \, \varphi_0).$$

For n = 0 WC have I- $\varphi_0 \supset \varphi_0$ which is the induction statement for n = 0.

Assumc that the statement above has been proved for a certain n and consider its proof for n+ 1.

Consider the EVNT rule with $\varphi = \varphi_{n+1}$, $\psi = (\bigvee_{i=0}^{n} \varphi_i)$. As shown in the proof of the livcness case, all the premises of the EVNT rule are satisficd. Consequently we may conclude:

$$\vdash \varphi_{n+1} \supset \varphi_{n+1} \,\mathcal{U}\, (\bigvee_{i=0}^{n} \varphi_i).$$

By the induction hypothesis and the $\mathcal{U}\mathcal{U}$ rule this yields

$$\vdash \varphi_{n+1} \supset \varphi_{n+1} \,\mathcal{U}\, (\varphi_n \,\mathcal{U}\, \ldots \varphi_1 \,\mathcal{U}\, \varphi_0).$$

Due to $\vdash v \supset (u \mathcal{U} v)$ which is a consequence of axiom A9, the induction hypothesis can also be written as

$$\vdash (\bigvee_{i=0}^{n} \varphi_i) \supset \varphi_{n+1} \,\mathcal{U}\, (\varphi_n \,\mathcal{U}\, \cdots \varphi_1 \,\mathcal{U}\, \varphi_0).$$

Taking the disjunction of the last two gives

$$\vdash (\bigvee_{i=0}^{n+1} \varphi_i) \supset \varphi_{n+1} \,\mathcal{U}\, (\varphi_n \,\mathcal{U}\, \ldots \varphi_1 \mathcal{U} \varphi_0),$$

which is the required staternent for n + 1.

Consider now a coarser partition:

$$0 = p_0 < p_1 < p_2 < \ldots < p_s = r.$$

By consecutively merging any Lwo contiguous assertions that fall into the same partition cell, using theorem T38:

$$\vdash (\varphi_{i+1} \,\mathcal{U}\, (\mathrm{Pi} \,\mathcal{U}\, \varphi)) \supset ((\varphi_{i+1} \vee \mathrm{Pi}) \,\mathcal{U}\, \varphi),$$

wc obtain the coarser conclusion:

$$\vdash (\bigvee_{i=0}^{n+1} \varphi_i) \supset ((\bigvee_{p_{s-1}<j\leq p_s} \varphi_j) \,\mathcal{U}\, (\bigvee_{p_{s-2}<j\leq p_{s-1}} \varphi_j) \,\mathcal{U} \ldots (\bigvee_{0<j\leq p_1} \varphi_j) \,\mathcal{U}\, \varphi_0)). \quad ■$$

Examples:

As our first example, let us consider the Mutual Exclusion program analyzed above. We have already proven that mutual exclusion is maintained by this program. Wc have also proven the livcness property that if P_1 wishes Lo enter its critical section it will eventually gain access Lo it. A more discriminating question is that, of how fair is our algorithm. That is, if P_1 wishes to enter its critical section, how many times will P_2 be able to enter its own critical section before P_1? Is that, number bounded? Wc refer to this question as the problem of bounded overtaking. Namely, how many times can P_2 overtake P_1 before P_1 enters his critical section.

Our first analysis makes use of Fig. 1 without any modifications. Wc only read from it Lhe stronger conclusion according to the stronger P-CIIAIN rule. As a partition we choosc $p_1 = 7$, $p_2 = 9$, $p_3 = r = 11$. Consequently, from Lhe diagram of Fig. 1 we conclude by the P-CHAIN rule:

$$\vdash \left(\bigvee_{i=1}^{11}\varphi_i\right) \supset \left(\left(\bigvee_{i=10}^{11}\varphi_i\right) \mathcal{U} \left(\bigvee_{i=8}^{9}\varphi_i\right) \mathcal{U} \left(\bigvee_{i=1}^{7}\varphi_i\right) \mathcal{U}\ \varphi_0\right).$$

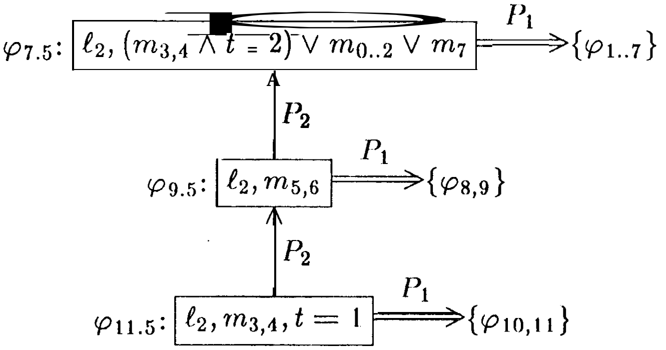
Replacing each of the right hand side disjunctions by a weaker property and the left hand side disjunction by a stronger statement we obtain:

$$\vdash \ell_{3,4} \supset \left((\sim m_{5,6})\ \mathcal{U}\ m_{5,6}\ \mathcal{U}\ (\sim m_{5,6})\ \mathcal{U}\ \ell_5\right).$$

This implies that if P_1 is at the wailing loop in $\ell_{3,4}$, there will ho a period in which P_2 is not in the critical section $m_{5,6}$, followed by a period in which P_2 is inside the critical section $m_{5,6}$ followed by a period in which P_2 is outside the critical section which terminates by P_1 entering his critical section. Since any of these periods may be empty this is a worst-case analysis. But it certainly assures 1-bounded overtaking, i.e., once P_1 is in $\ell_{3,4}$, P_2 may overtake it at most once.

Having successfully analyzed the situation from $\ell_{3,4}$ on we may attempt to obtain a similar analysis from the moment that P_1 enters ℓ_2.

This analysis calls for a refinement of the diagram of Fig. 1. The following is a subdiagrarn that should replace the node corresponding to φ_{12} in Fig. 1. It consists of three nodes labelled respectively $\varphi_{7.5}$, $\varphi_{9.5}$ and $\varphi_{11.5}$. The fractional indexing indicates that $\varphi_{7.5}$ should be inserted betwccn φ_7 and φ_8 in Fig. 1. The edges out of φ_{13} should enter one of these three nodes. Edges out of $\varphi_{7.5}$ lead Lo some of $\varphi_1, \ldots, \varphi_7$.


$\varphi_{7.5}$: $\ell_2, (m_{3,4} \wedge t = 2) \vee m_{0..2} \vee m_7$
P_1
$\{\varphi_{1..7}\}$
P_2
$\varphi_{9.5}$: $\ell_2, m_{5,6}$
P_1
$\{\varphi_{8,9}\}$
P_2
$\varphi_{11.5}$: $\ell_2, m_{3,4}, t = 1$
P_1
$\{\varphi_{10,11}\}$

Similarly for edges out of $\varphi_{9.5}$ and $\varphi_{11.5}$. Considering the updated diagram composed of Fig. 1 and the above subdiagram wc obtain the following conclusion:

$$\vdash \ell_{2..4} \supset \left(\left(\bigvee_{i=10}^{11.5}\varphi_i\right) \mathcal{U} \left(\bigvee_{i=8}^{9.5}\varphi_i\right) \mathcal{U} \left(\bigvee_{i=1}^{7.5}\varphi_i\right) \mathcal{U}\ \varphi_0\right).$$

This again leads to

$$\vdash\ \ell_{2..4} \supset \big((\sim m_{5,6})\ \mathcal{U}\ m_{5,6}\ \mathcal{U}\ (\sim m_{5,6})\ \mathcal{U}\ \ell_5\big),$$

which ensures 1--bounded overtaking even frorn ℓ_2. Encouraged by this, we may next ask whether a similar result can bc obtained from ℓ_1. Unfortunatcly this is not the case. P_2 may cntcr its critical section an arbitrary numbcr of timcs while P_1 is at ℓ_1. This is obvious since while being at ℓ_1, P_1 has not yet modified any variable in a way that will show that it is not still in ℓ_0. And naturally while P_1 is at ℓ_0, P_2 may enter the critical section any number of Limes if the algorithm is correct.

THE WELL-FOUNDED PRINCIPLE FOR PRECEDENCE PROPERTIES

A natural extension of the P-CHAIN rule Lo programs that require infinite chains of assertions again uses well founded ordcred sets.

Let $(A, \prec)$ be a well founded ordered sct. Wc require however that the ordering is total (or linear). That is, for every two distinct elements $\alpha_1, \alpha_2 \in$ A cither $\alpha_1 \prec \alpha_2$ or $\alpha_2 \prec \alpha_1$.

Well Founded Precedence Rule --- P-WELL

Let $\varphi(\alpha) = \varphi(\alpha;\overline{\pi};\overline{y})$ be a parametrized state assertion with a ∈ A.
Let h : A → $[1\ ..\ k]$ be a helpfulness function.
Let $\alpha_1 \prec \alpha_2 \prec \ldots \prec \alpha_s$ be a sequence of elements of A.

t- P leads from $\varphi(\alpha)$ to $\psi \vee (\exists\beta \preceq \alpha\ .\ \varphi(\beta))$

$\vdash P_{h(\alpha)}$ leads from $\varphi(\alpha)$ to $\psi \vee (\exists\beta \prec \alpha\ .\ \varphi(\beta))$

t- $\varphi(\alpha) \supset \Diamond[\psi \vee (\exists\beta \prec \alpha\ .\ \varphi(\beta)) \vee \mathit{Enabled}(P_{h(\alpha)})]$

$$\vdash (\exists\alpha \preceq \alpha_s\ .\ \varphi(\alpha)) \supset (\psi_s\ \mathcal{U}\ \psi_{s-1}\ \mathcal{U} \ldots \psi_1\ \mathcal{U}\ \psi)$$

where

ψ_ℓ is $\exists\beta(\alpha_{\ell-1} \prec \beta \preceq \alpha_\ell)\ .\ \varphi(\beta)$ for $\ell = 2, \ldots s,$ and

ψ_1 is $\exists\beta(\beta \preceq \alpha_1)\ .\ \varphi(\beta)$

Note Lhat whilc the range of thc paramctcr in thc asscrtions is infinitc, thc partition is still finite.

Acknowledgement:

We thankfully acknowlcdgc the help extended to us by Yoni Malachi, Bcn Moszkowski, Stuart Russcll, and Fran k Ycllin in reading thc manuscript. Special thanks are due Lo Evclyn Eldridgc-Diaz for TEXing the manuscript and Lo Carol Wcintraub for typing its first draft.

REFERENCES

[II] Hoare, C.A.R., "Communicating Sequential Processes," CACM 21 (1978) pp. 666-677.

[ILL] Igarashi, S., London, R.L., Luckham, D.C., "Automatic Program Verification I: A Logical Basis and Its Implementation," *Acta* ***Informatica,*** Vol. 4, No. 2 (1975), pp. 145-182.

[KR] Kuiper, R. and de Roever, W.P. "Fairness Assumptions for CSP in a Temporal Logic Framework," TC2 Working Conference on the Formal Description of Programming Concepts, Garmisch (June 1982).

[L1] Lamport, L., "Proving the Correctness of Multiprocess Programs," IEEE Trans. Soft. Eng. SE-3, 2 (Mar. 1977), pp. 125-143.

[L2] Lamport, L . , " 'Sometime' is Sometimes 'Not Never': On the Temporal Logic of Programs," 7th Annual ACM Symposium on Principles of Programming Languages (1980), pp. 174- 185.

[LPS] Lehmann, D., A. Pnueli, and J. S tavi, "Impartiality, justice and fairness: the ethics of concurrent termination," in *Automata* ***Languages and Programming,*** Lecture Notes in Computer Science 115, Springer Verlag (198 I), pp. 264-277.

[M] Manna, Z., "Verification of Sequential Programs: Temporal Axiomatization," Theoretical Foundations of Programming Methodology (M. Rroy and G. Schmidt, cds.), NATO Scientific Series, D. Reidel Pub. Co., Holland (1982), pp. 53-102.

[MP1] Manna, Z. and A. Pnucli, "Verification of Concurren t Programs: The Temporal Frame work," in ***The Correctness Problem*** *in* ***Computer Science*** (R.S. Boyer and J S. Moore, cds.), International Lecture Series in Computer Science, Academic Press, London (1982), pp. 215-273.

[MP2] Manna, Z. and A. Pnueli, "Verification of Concurrent Programs: Temporal Proof Principles," Proc. of the Workshop on Logic of Programs (D. Kozen, ed.), Yorktown-Heights, N.Y. (198 L). Springer- Verlag Lecture Notes in Computer Science 131, pp. 200-252.

[MP3] Manna, Z. and A. Pnucli, "Verification of Concurrent Programs: Proving Eventualities by Well-Pounded Ranking," TOPLAS (1983, to appear).

[MP4] Manna, Z. and A. Pnueli, "How to Cook a Temporal Proof System for Your Pet Language," in the Proc. of the Symposium on Principles of Programming Languages, Austin, Texas (Jan. 1983).

[OL] Owicki, S. 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.

[Pe] Peterson, G.L., "Myths about, the Mutual Exclusion Problem," Information Processi ng Letters, Vol. 12, No. 3 (June 1981), pp. 115-116.

[I'S] Pnueli, A. and R. Sherman, "Semantic Tableau for Temporal Logic," Technical Report,, CS81-21, The Weizmann Institute (Sept. 81).