A model for programming language semantics.

C. A. R. Hoare.

Draft, December 1975.

This paper presents a model of programming language semantics, which associates a program with a set of uninterpreted traces of program execution. The familiar simple program structures are modelled by regular sets; and more novel constructs such as recovery blocks, guarded commands, and nondeterminacy, can be easily represented. A proof theory for the language can be derived in the form of a "weakest precondition" predicate transformer.

$(p \cup q) \;\&\; p \Rightarrow s \;\&\; q \Rightarrow t$

(2009) early draft of "Some properties of Predicate Transformers"

1. Introduction.

Most constructive programming language definitions specify an interpreter for the language, which takes a program and an initial machine state, and traces a sequence of machine states which result from execution of the successive instructions of the program. This method has the advantage that it models the behaviour of an actual machine obeying actual programs. However, some complexity can be involved in proving properties of the language and its programs; the proofs sometimes depend on global properties of the interpreter, and it is not generally possible to deal with each feature of the language independently.

In this paper, we borrow from the interpretive method the basic idea of defining a programming language by associating with each program a trace of its execution. However, we avoid altogether the introduction of the concept of a machine state by defining the trace as a sequence of elementary actions and assertions; and we associate with each program command the entire set of traces which could result from its execution in any initial machine state. In fact, it is convenient to ~~define this set as a rather large superset of all~~ include in the set a number of impossible traces; feasible traces; this reduces complexity, and makes it easier to check intuitively that all possible traces have been included.

1.1 Traces.

A trace is a "program" consisting just of a sequence of normal assignments, interspersed with assertions. The assertions, like those of Floyd, are intended to express some fact about the program variables when that assertion is reached in execution of the program. A trace is "executed" by execution of its assignments and evaluation of its assertions, in the order given. If an assertion is false, execution of that trace fails. If all assertions evaluate to true, the trace is said to be feasible for the machine state in which it started. A trace which fails for every initial machine state is said to be infeasible.

We shall use the following syntax

<trace> ::= <empty>|<trace>;<element>

<element> ::= <assignment>|<assertion>

where assignments have their usual form, and assertions are propositional formulae of the conventional predicate calculus. We regard ; as a concatenation operator on traces, and take advantage of its associativity to write

$$s;t;u = s;(t;u) = (s;t);u.$$

Also note that s = s;<empty> = <empty>;s

We shall use the letters and forms:

p,q,r	to denote assertions
i,j,k,n	to denote logical variables (that cannot appear in programs); they range over natural numbers 0,1,2,...
s,t,u	to denote traces
x,y,z	to denote program variables
x:=e	to denote assignments.
S,T,U,V	to denote sets of traces
$\underline{F}$,$\underline{T}$	to denote the universally false and true predicates.

Using familiar methods, it is possible to define the weakest precondition under which execution of a trace s is feasible, and its execution would ensure that an assertion r would be true afterwards. The weakest precondition of a trace is itself an assertion about the variables of the program; and corresponds to Dijkstra's wp(s,r).

This precondition is written:

$$s \ \underline{a} \ r \quad \text{(s achieves r)}$$

and satisfies the following axioms, (which may be regarded as a recursive definition of $\underline{a}$):

$$(\text{<empty>} \ \underline{a} \ r) = r \tag{a1}$$

The empty trace has no effect: r is true after iff it is true before.

$$(x:=e \quad \underline{a} \ r) = r^x_e \tag{a2}$$

(where r^x_e is r with all free occurrences of x replaced by free occurrences of e).

An assignment changes the value of x to e. r is true of the value of x after the assignment iff it is true of the value of e before the assignment.

$$(p \underline{a} r) = p \,\&\, r \tag{a3}$$

An assertion has no effect; except to fail if it is false. To avoid failure, p must be true beforehand; and to achieve r, r must also be true beforehand. If both p & r are true, the assertion of p will succeed, and r will still be true afterwards.

$$((s;t) \underline{a} r) = s \underline{a} (t \underline{a} r) \tag{a4}$$

To achieve r after a compound statement (s;t), s must achieve a condition from which t can then achieve r.

The following theorems give the relationship between $\underline{a}$ and the familiar operators of propositional and predicate calculus, and show how the phrase "s $\underline{a}$" distributes through a logical formula. They can be verified in the case where the trace s is empty, a single assignment, or a single assertion; and their truth for concatenation (s;t) follows from the truth for s and t separately. The conclusions follow by induction on the length of the trace.

$$s \underline{a} \underline{F} = \underline{F} \tag{a5}$$

$$s \underline{a} p = (s \underline{a} T) \,\&\, (s \underline{a} p) \tag{a6}$$

$$s \underline{a} \bar{p} = s \underline{a} \underline{T} \,\&\, \overline{s \underline{a} p} \tag{a7}$$

$$s \underline{a} (p \& q) = (s \underline{a} p) \,\&\, (s \underline{a} q) \tag{a8}$$

$$s \underline{a} (p \vee q) = (s \underline{a} p) \vee (s \underline{a} q) \tag{a9}$$

$$s \underline{a} \forall i\, p_i = \forall i (s \underline{a} p_i) \tag{a10}$$

$$s \underline{a} \exists i\, p_i = \exists i (s \underline{a} p_i) \tag{a11}$$

$$s \underline{a} (p \Rightarrow q) = s \underline{a} \underline{T} \,\&\, (s \underline{a} p \Rightarrow s \underline{a} q) \tag{a12}$$

$$s \underline{a} (p \equiv q) = s \underline{a} \underline{T} \,\&\, ((s \underline{a} p) \equiv (s \underline{a} q)) \tag{a13}$$

$$s \underline{a} (p \& \bar{q}) = (s \underline{a} p) \,\&\, (\overline{s \underline{a} q}) \tag{a14}$$

$$s \underline{a} (p \not\equiv q) = ((s \underline{a} p) \not\equiv (s \underline{a} q)) \tag{a15}$$

$$\vdash (p \Rightarrow q) \Rightarrow \vdash (s \underline{a} p \Rightarrow s \underline{a} q)^{*} \tag{a16}$$

* The notation $\vdash p$ stands for the universal closure of the formula p - i.e., a formula in which all free variables of p are universally quantified.

We are also interested in the "weakest liberal precondition" under which a trace s ensures the truth of a condition r; this condition is true if s is infeasible; but if it is feasible, r must be true afterwards. The weakest liberal precondition is written:

$$s \underline{e} r \quad \text{(s ensures r)}$$

and may be defined

$$s \underline{e} r = \overline{s \underline{a} \bar{r}} \tag{ae1}$$

i.e., s ensures r if it can't achieve not-r.

Similarly,
$$s \underline{a} r = \overline{s \underline{e} \bar{r}} \tag{ae2}$$

$$s \underline{e} r = (s \underline{a} \underline{T} \Rightarrow s \underline{a} r) \tag{ae3}$$

$$s \underline{a} r = \overline{s \underline{e} \underline{F}} \;\&\; s \underline{e} r \tag{ae4}$$

The following theorems follow immediately

$$(\langle\text{empty}\rangle \underline{e} r) = r \tag{e1}$$

$$(x{:=}e \underline{e} r) = r^x_e \tag{e2}$$

$$(p \underline{e} r) = (p \Rightarrow r) \tag{e3}$$

$$((s;t) \underline{e} r) = s \underline{e} (t \underline{e} r) \tag{e4}$$

$$s \underline{e} \underline{T} = \underline{T} \tag{e5}$$

$$s \underline{e} p = (s \underline{e} \underline{F}) \vee (s \underline{e} p) \tag{e6}$$

$$s \underline{e} \bar{p} = (s \underline{e} \underline{F}) \vee \overline{(s \underline{e} p)} \tag{e7}$$

$$s \underline{e} (p \& q) = (s \underline{e} p) \;\&\; (s \underline{e} q) \tag{e8}$$

$$s \underline{e} (p \vee q) = (s \underline{e} p) \vee (s \underline{e} q) \tag{e9}$$

$$s \underline{e} (\forall i\ p_i) = \forall i\ (s \underline{e} p_i) \tag{e10}$$

$$s \underline{e} (\exists i\ p_i) = \exists i\ (s \underline{e} p_i) \tag{e11}$$

$$s \underline{e} (p \Rightarrow q) = (s \underline{e} p \Rightarrow s \underline{e} q) \tag{e12}$$

$$s \underline{e} (p \equiv q) = (s \underline{e} p \equiv s \underline{e} q) \tag{e13}$$

$$s \underline{e} (p \& \bar{q}) = ((s \underline{e} \underline{F}) \vee (s \underline{e} p) \;\&\; \overline{(s \underline{e} q)}) \tag{e14}$$

$$s \underline{e} (p \neq q) = ((s \underline{e} \underline{F}) \vee (s \underline{e} p \neq s \underline{e} q)) \tag{e15}$$

$$\vdash (p \Rightarrow q) \Rightarrow \vdash (s \underline{e} p \Rightarrow s \underline{e} q) \tag{e16}$$

A command of a programming language will be equated with a finite or infinite set of traces. Each trace in the set is necessarily a finite sequence, so the set must be denumerable. A non-terminating program will correspond to a set of traces, all of which are infeasible.

It is possible to define for sets of traces an analogue of the $\underline{e}$ and $\underline{a}$ concepts defined for single traces. If S is a set of traces, we define

$$S \mathbin{\underline{a}} r =_{df} \exists s \in S\ (s \mathbin{\underline{a}} r) \tag{A1}$$

S achieves r if it is possible for a trace of S to achieve r. Thus S must contain at least one terminating computation; but in a nondeterministic language there is nothing to prevent some other terminating computation from achieving $\bar{r}$.

$$S \mathbin{\underline{e}} r =_{df} \forall s \in S\ (s \mathbin{\underline{e}} r) \tag{E1}$$

S ensures r if every terminating computation of S ensures r. The possibility that every trace of S is infeasible is not excluded. The familiar notation of conditional correctness can be defined

$$p\ \{S\}\ r =_{df} \vdash (p \Rightarrow (S \mathbin{\underline{e}} r))$$

The following theorems show the properties of $\underline{a}$ and $\underline{e}$ when applied to sets of traces. The implications are less interesting than the equations; they are included for completeness.

$$S \mathbin{\underline{a}} \underline{F} = \underline{F} \tag{A5}$$

$$S \mathbin{\underline{a}} p = S \mathbin{\underline{a}} \underline{T} \ \&\ S \mathbin{\underline{a}} p \tag{A6}$$

$$S \mathbin{\underline{a}} T \ \&\ \overline{S \mathbin{\underline{a}} p} \Rightarrow S \mathbin{\underline{a}} \bar{p} \tag{A7}$$

$$S \mathbin{\underline{a}} (p \& q) \Rightarrow (S \mathbin{\underline{a}} p)\ \&\ (S \mathbin{\underline{a}} q) \tag{A8}$$

$$S \mathbin{\underline{a}} (p \vee q) = (S \mathbin{\underline{a}} p) \vee (S \mathbin{\underline{a}} q) \tag{A9}$$

$$S \mathbin{\underline{a}} (\forall i\ p_i) \Rightarrow \forall i (S \mathbin{\underline{a}} p_i) \tag{A10}$$

$$S \mathbin{\underline{a}} (\exists i\ p_i) = \exists i (S \mathbin{\underline{a}} p_i) \tag{A11}$$

$$S \mathbin{\underline{a}} T \ \&\ ((S \mathbin{\underline{a}} p) \Rightarrow (S \mathbin{\underline{a}} q)) \Rightarrow (S \mathbin{\underline{a}} (p \Rightarrow q)) \tag{A12}$$

there is no suitable theorem about $\equiv$

$$(S \mathbin{\underline{a}} p)\ \&\ (\overline{S \mathbin{\underline{a}} q}) \Rightarrow S \mathbin{\underline{a}} (p \& \bar{q}) \tag{A14}$$

$$((S \underline{a} p) \neq (S \underline{a} q)) \Rightarrow S \underline{a}(p \neq q) \quad \text{(A15)}$$

$$\vdash(p \Rightarrow q) \Rightarrow \vdash(S \underline{a} p \Rightarrow S \underline{a} q) \quad \text{(A16)}$$

$$S \underline{e} r = \overline{S \underline{a} \bar{r}} \quad \text{(AE1)}$$

$$S \underline{a} r = \overline{S \underline{e} \bar{r}} \quad \text{(AE2)}$$

$$(S \underline{e} r) \Rightarrow ((S \underline{a} T) \Rightarrow (S \underline{a} r)) \quad \text{(AE3)}$$

$$(\overline{S \underline{e} F}) \ \& \ (S \underline{e} r) \Rightarrow (S \underline{a} r) \quad \text{(AE4)}$$

$$S \underline{e} T = T \quad \text{(E5)}$$

$$S \underline{e} p = (S \underline{e} F) \vee (S \underline{e} p) \quad \text{(E6)}$$

$$S \underline{e} \bar{p} \Rightarrow (S \underline{e} F) \vee (\overline{S \underline{e} p}) \quad \text{(E7)}$$

$$S \underline{e}(p \& q) = (S \underline{e} p) \ \& \ (S \underline{e} q) \quad \text{(E8)}$$

$$(S \underline{e} p) \vee (S \underline{e} q) \Rightarrow S \underline{e}(p \vee q) \quad \text{(E9)}$$

$$S \underline{e}(\forall i \ p_i) = \forall i (S \underline{e} p_i) \quad \text{(E10)}$$

$$\exists i (S \underline{e} p_i) \Rightarrow S \underline{e}(\exists i \ p_i) \quad \text{(E11)}$$

$$S \underline{e}(p \Rightarrow q) \Rightarrow ((S \underline{e} p) \Rightarrow (S \underline{e} q)) \quad \text{(E12)}$$

$$S \underline{e}(p \equiv q) \Rightarrow ((S \underline{e} p) \equiv (S \underline{e} q)) \quad \text{(E13)}$$

$$S \underline{e}(p \& \bar{q}) \Rightarrow S \underline{e} F \vee (S \underline{e} p) \ \& \ (\overline{S \underline{e} q}) \quad \text{(E14)}$$

there is no suitable theorem about $\neq$

$$\vdash(p \Rightarrow q) \Rightarrow ((S \underline{e} p) \Rightarrow (S \underline{e} q)) \quad \text{(E16)}$$

It is interesting to define the <u>unconditional</u> correctness of a program represented by a set of traces S as

$$S \underline{i} r =_{df} S \underline{e} r \ \& \ S \underline{a} r \quad \text{(I1)}$$

S $\underline{i}$ r guarantees that S terminates and always ends in a state satisfying r. The following theorems are immediate

$$S \underline{i} F = F \quad \text{(I5)}$$

$$S \underline{i}(p \& q) = (S \underline{i} p) \ \& \ (S \underline{i} q) \quad \text{(I8)}$$

$$(S \underline{i} p) \vee (S \underline{i} q) \Rightarrow S \underline{i}(p \vee q) \quad \text{(I9)}$$

$$\vdash(p \Rightarrow q) \Rightarrow \vdash((S \underline{i} p) \Rightarrow (S \underline{i} q)) \quad \text{(I16)}$$

These correspond to Dijkstra's first four rules for the healthiness of predicate transformers. However S $\underline{i}$ p does not correspond to Dijkstra's "wp(S,p)", because it does not necessarily satisfy the condition of bounded nondeterminism; nor does it preclude the possibility that execution of S as a program will require backtracking. It is for this reason, for example, that the implication

$$S \ \underline{i} \ (T \ \underline{i} \ r) \Rightarrow [S;T] \ \underline{i} \ r$$

cannot be strengthened to an equivalence.

2. Regular expressions.

In this section we confine attention to commands of a programming language that generate regular sets of traces. We shall use square brackets to bracket commands.

2.1 Atomic commands.

An atomic command is just an assertion or an assignment. It generates a set of traces with only one member, namely the trace consisting of that single element, i.e.

$$[p] = \text{df } \{s \mid s = \langle p \rangle\}$$

$$[x:=e] = \text{df } \{s \mid s = \langle x:=e \rangle\}$$

The following theorems are immediate:

$[x:=e] \ \underline{e} \ q = q^x_e$	(E2)	$[x:=e] \ \underline{a} \ q = q^x_e$	(A2)
$[p] \ \underline{e} \ q \equiv (p \Rightarrow q)$	(E3)	$[p] \ \underline{a} \ q = p \ \& \ q$	(A3)
$[\underline{T}] \ \underline{e} \ q \equiv q$		$[T] \ \underline{a} \ q \equiv q$	
$[\underline{F}] \ \underline{e} \ q \equiv \underline{T}$		$[\underline{F}] \ \underline{a} \ q \equiv \underline{F}$	

The last two theorems suggest a definition of Dijkstra's primitives:

$$\text{skip} =_{\text{df}} [\underline{T}]$$

$$\text{abort} =_{\text{df}} [\underline{F}]$$

2.2 Concatenation.

$$[S;T] = df \{s;t \mid s \in S \ \& \ t \in T\}$$

This definition states simply that each trace of a compound command is a trace of the first command followed by a trace of the second command. Of course many such traces will be always infeasible.

Concatenation of commands is obviously associative, and we write

$$S;T;U = df\ S;[T;U] = [S;T];U \qquad \text{etc.}$$

The following theorems are immediate

$$[S;T]\ \underline{e}\ p \equiv S\ \underline{e}\ (T\ \underline{e}\ p) \qquad (E4) \qquad [S;T]\ \underline{a}\ p = S\ \underline{a}\ (T\ \underline{a}\ p) \qquad (A4)$$

Immediate consequences are:

$$[S;p]\ \underline{e}\ q \equiv S\ \underline{e}\ (p \Rightarrow q) \qquad ([S;p]\ \underline{a}\ q) = S\ \underline{a}(p\&q)$$

$$[p;S]\ \underline{e}\ q \equiv p \Rightarrow (S\ \underline{e}\ q) \qquad ([p;S]\ \underline{a}\ q) = p\ \&(S\ \underline{a}\ q)$$

$$\vdash(p \Rightarrow S\ \underline{e}\ q)\ \&\ \vdash(q \Rightarrow T\ \underline{e}\ r) \Rightarrow \vdash(p \Rightarrow [S;T]\ \underline{e}\ r)$$

$$\vdash(p \Rightarrow S\ \underline{a}\ q)\ \&\ \vdash(q \Rightarrow T\ \underline{a}\ r) \Rightarrow \vdash(p \Rightarrow [S;T]\ \underline{a}\ r)$$

$$(p\{S\}q)\&(q\{T\}r) \Rightarrow p\{S;T\}r$$

2.3 Union.

$$[S\square\ T] = S \cup T$$

Each trace of $S\square\ T$ is either a trace of S or a trace of T. The following theorems are immediate

$$[S\square\ T]\ \underline{e}\ p \equiv S\ \underline{e}\ p\ \&\ T\ \underline{e}\ p \qquad [S\square\ T]\ \underline{a}\ p = (S\ \underline{a}\ p) \vee (T\ \underline{a}\ p)$$

Since $\square$ is associative, we write:

$$S\square\ T\square\ U = df\ S\square[T\square\ U] = [S\square\ T]\square\ U, \text{ etc.}$$

Furthermore, composition distributes through union.

$$[S\square\ T];U = [S;U]\square[T;U]$$

$$S;[T\square\ U] = [S;T]\square[S;U]$$

This motivates a further omission of brackets, under the convention that ; binds tighter than $\square$, i.e.,

$$[S;T\square\ U;V] = [[S;T]\square[U;V]] \text{ etc.}$$

We can define a conventional conditional command:

$\underline{\text{if}}\ p\ \underline{\text{then}}\ S\ \underline{\text{else}}\ T = [p;S \Box \bar{p};T]$

from which it follows that

$[\underline{\text{if}}\ p\ \underline{\text{then}}\ S\ \underline{\text{else}}\ T]\ \underline{e}\ q = (p \Rightarrow (S\ \underline{e}\ q))\ \&\ (\bar{p} \Rightarrow (T\ \underline{e}\ q))$

$(p\ \&\ S\ \underline{a}\ q) \vee (\bar{p}\ \&\ T\ \underline{a}\ q) = [\underline{\text{if}}\ p\ \underline{\text{then}}\ S\ \underline{\text{else}}\ T]\ \underline{a}\ q$

Dijkstra's guarded command can be similarly defined:

$\underline{\text{if}}\ p \rightarrow S \Box q \rightarrow T\ \underline{\text{fi}} = \text{df}\ [p;S \Box q;T]$

with proof rules

$[\underline{\text{if}}\ p \rightarrow S \Box q \rightarrow T\ \underline{\text{fi}}]\ \underline{e}\ r = (p \Rightarrow (S\ \underline{e}\ r))\ \&\ (q \Rightarrow (T\ \underline{e}\ r))$

$[\underline{\text{if}}\ p \rightarrow S \Box q \rightarrow T\ \underline{\text{fi}}]\ \underline{a}\ r = p\ \&\ (S\ \underline{a}\ r) \vee q\ \&\ (T\ \underline{a}\ r)$

Randell's "recovery block" can be defined:

$[\underline{\text{achieve}}\ p\ \underline{\text{by}}\ S\ \underline{\text{or}}\ T] \equiv [S \Box T];p$

with proof rules

$[\underline{\text{achieve}}\ p\ \underline{\text{by}}\ S\ \underline{\text{or}}\ T]\ \underline{e}\ r = S\ \underline{e}\ (p \Rightarrow r)\ \&\ T\ \underline{e}\ (p \Rightarrow r)$

$[\underline{\text{achieve}}\ p\ \underline{\text{by}}\ S\ \underline{\text{or}}\ T]\ \underline{a}\ r = S\ \underline{a}\ (p\ \&\ r) \vee T\ \underline{a}\ (p\ \&\ r)$

2.4 Repetition.

Define $S^0 = \{\langle\text{empty}\rangle\}$

(i.e. the set consisting only of the empty trace)

$S^{n+1} = [S;S^n]$

$S^* = \bigcup_{n=0}^{\infty} S^n$

Each trace of S^n consists of a concatenation of n traces (usually different), each taken from S. Each trace of S^* consists of a trace of S^n for some n. It can be proved that

$$S^* = [\langle\text{empty}\rangle \Box S;S^*]$$

$[S^*]\ \underline{e}\ p \equiv \forall n\ (S^n\ \underline{e}\ p) \qquad [S^*]\ \underline{a}\ p = \exists n\ (S^n\ \underline{a}\ p)$

$\vdash (p \Rightarrow S\ \underline{e}\ p) \Rightarrow \vdash (p \Rightarrow S^*\ \underline{e}\ p)$

$\vdash (p_{n+1} \Rightarrow p_0 \vee S\ \underline{a}\ p_n) \Rightarrow \vdash (p_n \Rightarrow (S^*\ \underline{a}\ p_0))$

The conventional while can be defined

$$\underline{\text{while}}\ p\ \underline{\text{do}}\ S = [p;S]^*;\ \bar{p}$$

and its weakest preconditions are given by

$$[\underline{\text{while}}\ p\ \underline{\text{do}}\ S]\ \underline{e}\ q \equiv \forall n([p;S]^n\ \underline{e}\ (p \vee q))$$

$$[\underline{\text{while}}\ p\ \underline{\text{do}}\ S]\ \underline{a}\ q = \exists n([p;S]^n\ \underline{a}\ (\bar{p}\&q))$$

from which can be derived

$$\vdash(q\&p \Rightarrow S\ \underline{e}\ q) \Rightarrow \vdash(q \Rightarrow [\underline{\text{while}}\ p\ \underline{\text{do}}\ S]\ \underline{e}\ (q\ \&\ \bar{p}))$$

and if $p_o = \bar{p}$

$$\vdash(p_{n+1} \Rightarrow \bar{p} \vee S\ \underline{a}\ p_n) \Rightarrow \vdash(p_n \Rightarrow [\underline{\text{while}}\ p\ \underline{\text{do}}\ S]\ \underline{a}\ \bar{p})$$

Dijkstra's repetitive construct can also be defined, e.g.

$$\underline{\text{do}}\ p \to S \square\ q \to T\ \underline{\text{od}} =_{df} [p;S \square\ q;T]^*;\ \bar{p}\ \&\ \bar{q}$$

Conclusion.

The trace technique introduced in this paper gives a purely syntactic model of programs, traces and assertions; however, the semantics of (predicates) can be explained by the standard interpretation as sets of mappings between free variables and values; and the assignments and assertions can be explained as relations between such mappings; and then all the theorems of this paper can be interpreted as theorems of set and relation theory.

The trace technique is essentially first-order, and does not permit a command to be assigned as a value to a variable; it is essentially less powerful than higher-order techniques, but it has the advantage of dealing happily with negation and quantifiers. In describing a language intended for specifying hardware algorithms as well as software, the restriction to the first order may be realistic.

The trace technique is in no way intended to supplant axiomatic methods of programming language definition, which should be taken as primary. Indeed, the

Appendix

(1) Proofs of theorems stated in this paper, concerning single traces and $\underline{a}$ (achieves).

We are given the following axioms:-

$\langle empty \rangle \ \underline{a}\ r = r$ (a1) $\qquad \langle x:=e \rangle \ \underline{a}\ r = r^x_e$ (a2)

$\langle p \rangle \ \underline{a}\ r = p \ \& \ r$ (a3) $\qquad (s;t) \ \underline{a}\ r = s \ \underline{a}\ (t \ \underline{a}\ r)$ (a4)

First, we prove the following

$$\vdash(p \Rightarrow q) \Rightarrow \vdash(s \ \underline{a}\ p \Rightarrow s \ \underline{a}\ q) \qquad \text{(a16)}$$

Consider the following cases:-

i) $s = \langle empty \rangle$ then obviously(a16) is true, from (a1)

ii) $s = \langle x:=e \rangle$ " " " " " , from (a2)

iii) $s = \langle r \rangle$ " " " " " , from (a3)

iv) $\left.\begin{array}{l}(s;t) \ \underline{a}\ p = s \ \underline{a}\ (t \ \underline{a}\ p) \\ \& \ (s;t) \ \underline{a}\ q = s \ \underline{a}\ (t \ \underline{a}\ q)\end{array}\right\}$ (a4)

but $\vdash(p \Rightarrow q) \Rightarrow \vdash(t \ \underline{a}\ p \Rightarrow t \ \underline{a}\ q)$ by induction hypothesis (i.h.)

and $\vdash(t \ \underline{a}\ p \Rightarrow t \ \underline{a}\ q) \Rightarrow \vdash(s \ \underline{a}\ (t \ \underline{a}\ p) \Rightarrow s \ \underline{a}\ (t \ \underline{a}\ q))$ "

Hence (a16) is true for all traces s as outlined by the induction proof above.

We derive the following corollary of (a16)

$$\vdash(p \equiv q) \Rightarrow \vdash(s \ \underline{a}\ p \equiv s \ \underline{a}\ q)$$

this follows immediately since $(p \equiv q) \equiv ((p \Rightarrow q) \ \& \ (q \Rightarrow p))$

Thus, we are permitted to perform substitution of equivalent predicates on the predicate operand of achieves ($\underline{a}$).

$$s\ \underline{a}\ \underline{F} = \underline{F} \qquad \text{(a5)}$$

i) $s = \langle \text{empty} \rangle$ then (a5) is trivially true, by (a1)

ii) $s = \langle x:=e \rangle$ " " " " " , by (a2)

iii) $s = \langle r \rangle$ " " " " " , by (a3)

iv) $(s;t)\ \underline{a}\ F = s\ \underline{a}\ (t\ \underline{a}\ \underline{F})$ by (a4)

but $t\ \underline{a}\ \underline{F} = \underline{F}$ (i.h.)

$\therefore\ s\ \underline{a}\ (t\ \underline{a}\ \underline{F}) = s\ \underline{a}\ \underline{F}$ by corollary to (a16)

$= \underline{F}$ (i.h.)

The use of the corollary to (a16) will not be made explicit in further proofs as it is felt that this will hardly place a burden on the reader.

We find it convenient to break the order of the theorems by giving the proof for

$$s\ \underline{a}\ (p\ \&\ q) = s\ \underline{a}\ p\ \&\ s\ \underline{a}\ q \qquad \text{(a8)}$$

i) $s = \langle \text{empty} \rangle$ then trivially true by (a1)

ii) $s = \langle x:=e \rangle$ then $s\ \underline{a}\ (p\ \&\ q) = (p\ \&\ q)^x_e$

$= p^x_e\ \&\ q^x_e$

$= s\ \underline{a}\ p\ \&\ s\ \underline{a}\ q$

iii) $s = \langle r \rangle$ then $s\ \underline{a}\ (p\ \&\ q) = r\ \&\ (p\ \&\ q)$

$= (r\ \&\ p)\ \&\ (r\ \&\ q)$

$= s\ \underline{a}\ p\ \&\ s\ \underline{a}\ q$

iv) $(s;t)\ \underline{a}\ (p\ \&\ q) = s\ \underline{a}\ (t\ \underline{a}\ (p\ \&\ q))$ (a4)

but $t\ \underline{a}\ (p\ \&\ q) = t\ \underline{a}\ p\ \&\ t\ \underline{a}\ q$ (i.h.)

$\therefore\ s\ \underline{a}\ (t\ \underline{a}\ (p\ \&\ q)) = s\ \underline{a}\ (t\ \underline{a}\ p\ \&\ t\ \underline{a}\ q)$

$= s\ \underline{a}\ (t\ \underline{a}\ p)\ \&\ s\ \underline{a}\ (t\ \underline{a}\ q)$ (i.h.)

$= (s;t)\ \underline{a}\ p\ \&\ (s;t)\ \underline{a}\ q$ (a4)

Hence (a8) is proved.

$$s\ \underline{a}\ p = s\ \underline{a}\ \underline{T}\ \&\ s\ \underline{a}\ p \qquad (a6)$$

trivially $\quad p \equiv p\ \&\ \underline{T}$

Hence $\quad s\ \underline{a}\ p = s\ \underline{a}\ (p\ \&\ \underline{T}) = s\ \underline{a}\ p\ \&\ s\ \underline{a}\ \underline{T}$ by (a8)

$$s\ \underline{a}\ \bar{p} = s\ \underline{a}\ \underline{T}\ \&\ \overline{s\ \underline{a}\ p} \qquad (a7)$$

i) $s \equiv \langle\text{empty}\rangle$ then $s\ \underline{a}\ \bar{p} = \bar{p} = \underline{T}\ \&\ \bar{p} = s\ \underline{a}\ \underline{T}\ \&\ \overline{s\ \underline{a}\ p}$

ii) $s = \langle x:=e\rangle$ then $s\ \underline{a}\ \bar{p} = (\bar{p})^x_e = \bar{p}^x_e\ \&\ (\underline{T})^x_e = s\ \underline{a}\ \underline{T}\ \&\ \overline{s\ \underline{a}\ p}$

iii) $s = \langle r\rangle$ then $s\ \underline{a}\ \bar{p} = r\ \&\ \bar{p} = r\ \&\ \overline{r\ \&\ p} = s\ \underline{a}\ \underline{T}\ \&\ \overline{s\ \underline{a}\ p}$

iv) $(s;t)\ \underline{a}\ \bar{p} = s\ \underline{a}\ (t\ \underline{a}\ \bar{p}) \qquad (a4)$

$= s\ \underline{a}\ (t\ \underline{a}\ \underline{T}\ \&\ \overline{t\ \underline{a}\ p}) \qquad (i.h.)$

$= s\ \underline{a}\ (t\ \underline{a}\ \underline{T})\ \&\ s\ \underline{a}\ (\overline{t\ \underline{a}\ p}) \qquad (a8)$

$= s\ \underline{a}\ (t\ \underline{a}\ \underline{T})\ \&\ (s\ \underline{a}\ \underline{T}\ \&\ \overline{s\ \underline{a}\ (t\ \underline{a}\ p)}) \qquad (i.h.)$

$= s\ \underline{a}\ (t\ \underline{a}\ \underline{T})\ \&\ \overline{s\ \underline{a}\ (t\ \underline{a}\ p)} \qquad (a6)$

$= (s;t)\ \underline{a}\ \underline{T}\ \&\ \overline{(s;t)\ \underline{a}\ p} \qquad (a4)$

$$s\ \underline{a}\ (p \vee q) = s\ \underline{a}\ p \vee s\ \underline{a}\ q \qquad (a9)$$

i) $s = \langle\text{empty}\rangle$ then (a9) is trivially true by (a1)

ii) $s = \langle x:=e\rangle$ then $s\ \underline{a}\ (p \vee q) = (p \vee q)^x_e = p^x_e \vee q^x_e = s\ \underline{a}\ p \vee s\ \underline{a}\ q$

iii) $s = \langle r\rangle$ then $s\ \underline{a}\ (p \vee q) = r\ \&\ (p \vee q) = (r\ \&\ p) \vee (r\ \&\ q) = s\ \underline{a}\ p \vee s\ \underline{a}\ q$

iv) $(s;t)\ \underline{a}\ (p \vee q) = s\ \underline{a}\ (t\ \underline{a}\ (p \vee q)) \qquad (a4)$

$= s\ \underline{a}\ (t\ \underline{a}\ p \vee t\ \underline{a}\ q) \qquad (i.h.)$

$= s\ \underline{a}\ (t\ \underline{a}\ p) \vee s\ \underline{a}\ (t\ \underline{a}\ q) \qquad (i.h.)$

$= (s;t)\ \underline{a}\ p \vee (s;t)\ \underline{a}\ q \qquad (a4)$

$$s \underline{a} (\forall_i p_i) = \forall_i (s \underline{a} p_i) \qquad (a10)$$

i) $s = \langle \text{empty} \rangle$ then (a10) is trivially true

ii) $s = \langle x := e \rangle$ then, provided no bound variables occur in e, (which can be easily arranged by suitable substitution for bound variables)

$$s \underline{a} (\forall_i p_i) = (\forall_i p_i)^x_e = \forall_i (p_{ie}^x) = \forall_i (s \underline{a} p_i)$$

iii) $s = \langle r \rangle$ then $s \underline{a} (\forall_i p_i) = r \;\&\; (\forall_i p_i)$

$= \forall_i (r \;\&\; p_i)$ provided r does not contain bound variables

$= \forall_i (s \underline{a} p_i)$

iv) $(s;t) \underline{a} (\forall_i p_i) = s \underline{a} (t \underline{a} (\forall_i p_i)$ (a4)

$= s \underline{a} (\forall_i (t \underline{a} p_i))$ (i.h.)

$= \forall_i (s \underline{a} (t \underline{a} p_i))$ (i.h.)

$= \forall_i ((s;t) \underline{a} p_i)$ (a4)

$$s \underline{a} (\exists_i p_i) = \exists_i (s \underline{a} p_i) \qquad (a11)$$

$\exists_i p_i \equiv \overline{\forall_i \bar{p}_i}$ and applying (a7), (a10), (a7)

$$s \underline{a} (p \Rightarrow q) = s \underline{a} \underline{T} \;\&\; (s \underline{a} p \Rightarrow s \underline{a} q) \qquad (a12)$$

Use $(p \Rightarrow q) \equiv (\bar{p} \vee q)$

and apply (a9) (a7) & (a6)

$$s \underline{a} (p \equiv q) = s \underline{a} \underline{T} \;\&\; (s \underline{a} p \equiv s \underline{a} q) \qquad (a13)$$

Use $(p \equiv q) \equiv (p \Rightarrow q) \;\&\; (q \Rightarrow p)$

and apply (a8) (a12)

$s \underline{a} (p \,\&\, \bar{q}) = (s \underline{a} p \,\&\, \overline{s \underline{a} q})$ (a14)

apply (a8), (a7) & (a6)

$s \underline{a} (p \neq q) = (s \underline{a} p \neq s \underline{a} q)$ (a15)

$(p \neq q) \equiv (p \,\&\, \bar{q}) \vee (q \,\&\, \bar{p})$

apply (a9) and (a14)

---

(2) The following are the proofs of the theorems concerning single traces and $\underline{e}$ (ensures).

We are given the following definition $s \underline{e} r =_{df} \overline{s \underline{a} \bar{r}}$ (ael)

$$s \underline{a} r = \overline{s \underline{e} \bar{r}} \quad \text{(ae2)}$$

$$s \underline{e} \bar{r} =_{df} \overline{s \underline{a} r}$$

$$\therefore \quad \overline{s \underline{e} \bar{r}} = s \underline{a} r$$

$$s \underline{e} r = (s \underline{a} T \Rightarrow s \underline{a} r) \quad \text{(ae3)}$$

$$s \underline{e} r = \overline{s \underline{a} \bar{r}} \quad \text{(ael)}$$

$$= \overline{s \underline{a} T \;\&\; \overline{s \underline{a} r}} \quad \text{(a7)}$$

$$= s \underline{a} T \Rightarrow s \underline{a} r$$

$$s \underline{a} r = \overline{s \underline{e} F} \;\&\; s \underline{e} r \quad \text{(ae4)}$$

$$s \underline{a} r = s \underline{a} T \;\&\; \overline{s \underline{a} \bar{r}} \quad \text{(a7)}$$

$$= \overline{s \underline{e} F} \;\&\; s \underline{e} r \quad \text{(ae2) \& (ael)}$$

First we prove the following,

$$\vdash (p \Rightarrow q) \Rightarrow \vdash (s \underline{e} p \Rightarrow s \underline{e} q) \quad \text{(el6)}$$

$$(p \Rightarrow q) \Rightarrow (\bar{q} \Rightarrow \bar{p}) \Rightarrow (s \underline{a} \bar{q} \Rightarrow s \underline{a} \bar{p}) \quad \text{by (al6)}$$

$$\Rightarrow (\overline{s \underline{a} \bar{p}} \Rightarrow \overline{s \underline{a} \bar{q}}) = (s \underline{e} p \Rightarrow s \underline{e} q) \text{ by (ael)}$$

The corollary

$$\vdash (p \equiv q) \Rightarrow \vdash (s \underline{e} p \equiv s \underline{e} q)$$

follows immediately and so again the substitution of equivalent predicates is valid.

The proofs of theorems (e1) - (e15) follow directly from the definition of $\underline{e}$ and appropriate use of theorems for $\underline{a}$.

We give an outline of each proof.

(e1)	$\langle \text{empty} \rangle \mathbin{\underline{e}} r = r$	apply (ae1) & (a1)
(e2)	$\langle x := e \rangle \mathbin{\underline{e}} r = r^x_e$	apply (ae1) & (a2)
(e3)	$\langle p \rangle \mathbin{\underline{e}} r = (p \Rightarrow r)$	apply (ae1) & (a3)
(e4)	$(s;t) \mathbin{\underline{e}} r = s \mathbin{\underline{e}} (t \mathbin{\underline{e}} r)$	apply (ae1) & (a4) & (ae2) twice
(e5)	$s \mathbin{\underline{e}} \underline{T} = \underline{T}$	apply (ae1) & (a5)
(e6)	$s \mathbin{\underline{e}} p = s \mathbin{\underline{e}} \underline{F} \vee s \mathbin{\underline{e}} p$	apply (ae1) & (a6)
(e7)	$s \mathbin{\underline{e}} \bar{p} = s \mathbin{\underline{e}} \underline{F} \vee \overline{s \mathbin{\underline{e}} p}$	apply (ae1) & (a7)
(e8)	$s \mathbin{\underline{e}} (p \,\&\, q) = s \mathbin{\underline{e}} p \,\&\, s \mathbin{\underline{e}} q$	apply (ae1) & (a9)
(e9)	$s \mathbin{\underline{e}} (p \vee q) = s \mathbin{\underline{e}} p \vee s \mathbin{\underline{e}} q$	apply (ae1) & (a8)
(e10)	$s \mathbin{\underline{e}} (\forall_i p_i) = \forall_i (s \mathbin{\underline{e}} p_i)$	apply (ae1) & (a11)
(e11)	$s \mathbin{\underline{e}} (\exists_i p_i) = \exists_i (s \mathbin{\underline{e}} p_i)$	apply (ae1) & (a10)
(e12)	$s \mathbin{\underline{e}} (p \Rrightarrow q) = (s \mathbin{\underline{e}} p \Rightarrow s \mathbin{\underline{e}} q)$	apply (e9), (e7) & (e6)
(e13)	$s \mathbin{\underline{e}} (p \equiv q) = (s \mathbin{\underline{e}} p \equiv s \mathbin{\underline{e}} q)$	apply (e8) & (e12)
(e14)	$s \mathbin{\underline{e}} (p \,\&\, \bar{q}) = s \mathbin{\underline{e}} \underline{F} \vee (s \mathbin{\underline{e}} p \,\&\, \overline{s \mathbin{\underline{e}} q})$	apply (e8), (e7) & (e8)
(e15)	$s \mathbin{\underline{e}} (p \not\equiv q) = s \mathbin{\underline{e}} \underline{F} \vee (s \mathbin{\underline{e}} p \not\equiv s \mathbin{\underline{e}} q)$	apply (e7) & (e13)

(3) Proofs of theorems concerning sets of traces and $\underline{a}$ (achieves).

We are given the definition of

$$S \underline{a} p =_{df} \exists_{s \in S} (s \underline{a} p) \qquad (A1)$$

(A16) $\vdash (p \Rightarrow q) \Rightarrow \vdash (S \underline{a} p \Rightarrow S \underline{a} q)$ prove using (A1) & (a16)

(A5) $S \underline{a} F = F$ prove using (A1) & (a5)

(A6) $S \underline{a} p = S \underline{a} T \ \& \ S \underline{a} p$ prove using (A1) & (a6) for forward and reverse implication.

(A7) $S \underline{a} T \ \& \ \overline{S \underline{a} p} \Rightarrow S \underline{a} \bar{p}$ prove using (A1) & (a7)

(A8) $S \underline{a} (p \ \& \ q) \Rightarrow S \underline{a} p \ \& \ S \underline{a} q$ prove using (A1) & (a8)

(A9) $S \underline{a} (p \vee q) = S \underline{a} p \vee S \underline{a} q$ prove using (A1) & (a9) for forward and reverse implication.

(A10) $S \underline{a} (\forall_i p_i) \Rightarrow \forall_i (S \underline{a} p_i)$ prove using (A1) & (a10)

(A11) $S \underline{a} (\exists_i p_i) = \exists_i (S \underline{a} p_i)$ prove using (A1) & (a11)

(A12) $S \underline{a} T \ \& \ (S \underline{a} p \Rightarrow S \underline{a} q) \Rightarrow (S \underline{a} (p \Rightarrow q))$ prove using (A7), (A6) & (A9)

(A14) $S \underline{a} p \ \& \ \overline{S \underline{a} q} \Rightarrow S \underline{a} (p \ \& \ \bar{q})$ prove using (A1) & (a14)

(A15) $(S \underline{a} p \neq S \underline{a} q) \Rightarrow S \underline{a} (p \neq q)$ prove using (A14) & (A9)

(4) Proofs of theorems concerning sets of traces and $\underline{e}$ (ensures).

Given the definition $S \underline{e} r =_{df} \forall s \in S\ (s \underline{e} r)$ (E1)

(AE1) $S \underline{e} r = \overline{S \underline{a} \bar{r}}$

proof $S \underline{e} r = \forall s \in S\ (s \underline{e} r) = \forall s \in S\ \overline{(s \underline{a} \bar{r})}$ (ael)

$= \overline{\exists s \in S\ (s \underline{a} \bar{r})}$

$= \overline{S \underline{a} \bar{r}}$ (A1)

(AE2) $S \underline{a} r = \overline{S \underline{e} \bar{r}}$ apply (AE1)

(AE3) $S \underline{e} r \Rightarrow (S \underline{a} \underline{T} \Rightarrow S \underline{a} r)$ apply (E1) & (ae3)

(AE4) $\overline{S \underline{e} \underline{F}} \ \& \ S \underline{e} r \Rightarrow S \underline{a} r$ apply (AE2) & (AE3)

(E16) $\vdash (p \Rightarrow q) \Rightarrow \vdash (S \underline{e} p \Rightarrow S \underline{e} q)$ apply (el6) & (E1)

(E5) $S \underline{e} \underline{T} = \underline{T}$ apply (E1) & (e5)

(E6) $S \underline{e} p = S \underline{e} \underline{F} \vee S \underline{e} p$ apply (AE1) & (A6)

(E7) $S \underline{e} \bar{p} \Rightarrow S \underline{e} \underline{F} \vee \overline{S \underline{e} p}$

$S \underline{e} \bar{p} = \overline{S \underline{a} p}$ (AE2)

$\Rightarrow \overline{S \underline{a} \underline{T} \ \& \ \overline{S \underline{a} \bar{p}}}$ (A7)

$= S \underline{e} \underline{F} \vee \overline{S \underline{e} p}$ (AE1)

(E8) $S \underline{e} (p \,\&\, q) = S \underline{e} p \,\&\, S \underline{e} q$ apply (E1) & (e8)

(E9) $S \underline{e} p \vee S \underline{e} q \Rightarrow S \underline{e} (p \vee q)$ apply (E1) & (e9)

(E10) $S \underline{e} (\forall_i p_i) = \forall_i (S \underline{e} p_i)$ apply (E1) & (e10)

(E11) $\exists_i (S \underline{e} p_i) \Rightarrow S \underline{e} (\exists_i p_i)$ apply (E1) & (e11)

(E12) $S \underline{e} (p \Rightarrow q) \Rightarrow (S \underline{e} p \Rightarrow S \underline{e} q)$

$S \underline{e} (p \Rightarrow q) = (S \underline{e} (\bar{p} \vee q)$

$= \overline{S \underline{a} (p \,\&\, \bar{q})}$ (AE1)

$\Rightarrow \overline{S \underline{a} p \,\&\, \overline{S \underline{a} q}}$ (A14)

$= S \underline{a} p \Rightarrow S \underline{a} q$

(E13) $S \underline{e} (p \equiv q) \Rightarrow (S \underline{e} p \equiv S \underline{e} q)$ apply (E8) & (E12)

(E14) $S \underline{e} (p \,\&\, \bar{q}) \Rightarrow S \underline{e} F \vee (S \underline{e} p \,\&\, \overline{S \underline{e} q})$ apply (E8) (E6) & (E7)

---

(5) Proof of theorems concerning sets of traces and $\underline{i}$.

Given the definition $S\ \underline{i}\ p = S\ \underline{e}\ p\ \&\ S\ \underline{a}\ p$ (I1)

then

(I1) $S\ \underline{i}\ \underline{F} = \underline{F}$

$S\ \underline{i}\ \underline{F} = S\ \underline{e}\ \underline{F}\ \&\ S\ \underline{a}\ \underline{F}$ (I1)

$= S\ \underline{e}\ \underline{F}\ \&\ \underline{F}$ (A5)

$= \underline{F}$

(I8) $S\ \underline{i}\ (p\ \&\ q) = S\ \underline{i}\ p\ \&\ S\ \underline{i}\ q$

forward implication by (I1) & (A8) & (E8)

reverse implication

$S\ \underline{i}\ p\ \&\ S\ \underline{i}\ q = S\ \underline{e}\ p\ \&\ S\ \underline{e}\ q\ \&\ S\ \underline{a}\ p\ \&\ S\ \underline{a}\ q$ (I1)

$= S\ \underline{e}\ (p\ \&\ q)\ \&\ S\ \underline{a}\ p\ \&\ S\ \underline{a}\ q$ (E8)

but $S\ \underline{e}\ p\ \&\ S\ \underline{a}\ q \Rightarrow S\ \underline{a}\ (p\ \&\ q)$ by (AE1) & (A14)

$\therefore S\ \underline{e}\ (p\ \&\ q)\ \&\ S\ \underline{a}\ p \Rightarrow S\ \underline{a}\ (p\ \&\ q)$

Hence $S\ \underline{i}\ p\ \&\ S\ \underline{i}\ q \Rightarrow S\ \underline{e}\ (p\ \&\ q)\ \&\ S\ \underline{a}\ (p\ \&\ q)$

$= S\ \underline{i}\ (p\ \&\ q)$

(I9) $S\ \underline{i}\ (p \vee q) \Rightarrow S\ \underline{i}\ p \vee S\ \underline{i}\ q$ apply (I1) & (E9) & (A9)

(I16) $\vdash (p \Rightarrow q) \Rightarrow \vdash (S\ \underline{i}\ p \Rightarrow S\ \underline{i}\ q)$ apply (A16) & (E16)

(6) Atomic Commands

The theorems concerning the atomic commands shown are all proved by a direct application of the axioms and theorems concerning the corresponding single trace element.

---

(7) CONCATENATION

We must first state the following lemmas:-

If r contains no free variable assigned in s or S then

$r \Rightarrow (s \underline{e} p) = s \underline{e} (r \Rightarrow p)$ (e17)

$r \Rightarrow (S \underline{e} p) = S \underline{e} (r \Rightarrow p)$ (E17)

$r \,\&\, (s \underline{a} p) = s \underline{a} (r \,\&\, p)$ (a17)

$r \,\&\, (S \underline{a} p) = S \underline{a} (r \,\&\, p)$ (A17)

Then proof of $[S;T] \underline{e} p \equiv S \underline{e} (T \underline{e} p)$ (E4) follows,

$[S;T] \underline{e} p \equiv \forall u \in [S;T] (u \underline{e} p)$ (E1)

$\equiv \forall s,t (s \in S \Rightarrow (t \in T \Rightarrow (s;t) \underline{e} p))$ definition [S;T]

$\equiv \forall s,t (s \in S \Rightarrow (t \in T \Rightarrow s \underline{e} (t \underline{e} p))$ (e4)

$\equiv \forall s,t (s \in S \Rightarrow s \underline{e} (t \in T \Rightarrow t \underline{e} p)$ (e17)*

$\equiv \forall s \in S (s \underline{e} (\forall t \in T (t \underline{e} p)))$ (e10)

$\equiv S \underline{e} (T \underline{e} p)$

Footnote *In the proposition $t \in [T]$, where T is a description of a set of traces, the variable names occurring in T are not considered to be free and are therefore not subject to substitution by assignment i.e.

$$[t \in T]^{x}_{e} = t \in T$$

(this represents the fact that an assignment cannot modify a piece of program).

$[S;T] \; a \; p \equiv S \underline{a} (T \underline{a} p)$ (A4)

proof is similar to the above proof (E4).

$S \underline{i} (T \underline{i} p) \Rightarrow [S;T] \underline{i} p$ (I4)

$$
\begin{aligned}
S\ \underline{i}(T\ \underline{i}\ p) &= S\ \underline{e}(T\ \underline{i}\ p)\ \&\ S\ \underline{a}(T\ \underline{i}\ p) && \text{(I1)}\\
&= S\ \underline{e}(T\ \underline{e}\ p\ \&\ T\ \underline{a}\ p)\ \&\ S\ \underline{a}(T\ \underline{e}\ p\ \&\ T\ \underline{a}\ p) && \text{(I1)}\\
&\Rightarrow S\ \underline{e}(T\ \underline{e}\ p)\ \&\ S\ \underline{a}(T\ \underline{a}\ p) && \text{(E8) \& (A8)}\\
&= [S;T]\ \underline{e}\ p\ \&\ [S;T]\ \underline{a}\ p
\end{aligned}
$$

$[S;p]\ \underline{e}\ q \equiv S\ \underline{e}\ (p \Rightarrow q)$ (E4) & (E3)

$[p;S]\ \underline{e}\ q \equiv (p \Rightarrow (S\ \underline{e}\ q))$ (E4) & (E3)

$[S;p]\ \underline{a}\ q \equiv S\ \underline{a}\ (p\ \&\ q)$ (A4) & (A3)

$[p;S]\ \underline{a}\ q \equiv p\ \&\ (S\ \underline{a}\ q)$ (A4) & (A3)

$\vdash (p \Rightarrow S\ \underline{e}\ q)\ \&\ \vdash (q \Rightarrow T\ \underline{e}\ r) \Rightarrow \vdash (p \Rightarrow [S;T]\ \underline{e}\ r)$ (E16) & (E4)

$\vdash (p \Rightarrow S\ \underline{a}\ q)\ \&\ \vdash (q \Rightarrow T\ \underline{a}\ r) \Rightarrow \vdash (p \Rightarrow [S;T]\ \underline{a}\ r)$ (A16) & (A4)

---

(8) Union

$[S \square T]\ \underline{e}\ p \equiv \forall x \in (S \cup T)(x\ \underline{e}\ p)$ (E1)

$\equiv \forall s \in S\ (s\ \underline{e}\ p)\ \&\ \forall t \in T\ (t\ \underline{e}\ p)$

$\equiv S\ \underline{e}\ p\ \&\ T\ \underline{e}\ p$ (E1)

$[S \square T]\ \underline{a}\ p \equiv \exists x \in (S \cup T)(x\ \underline{a}\ p)$ (A1)

$\equiv \exists s \in S\ (s\ \underline{a}\ p) \vee \exists t \in T\ (t\ \underline{a}\ p)$

$\equiv S\ \underline{a}\ p \vee T\ \underline{a}\ p$ (A1)

---

(9) Repetition

$[S^*]\ \underline{e}\ p \equiv \forall n\ (S^n\ \underline{e}\ p)$ include 1st section overleaf

$[S^*]\ \underline{a}\ p \equiv \exists n\ (S^n\ \underline{a}\ p)$ include 2nd section overleaf

$\vdash (p \Rightarrow S\ \underline{e}\ p) \Rightarrow \vdash (p \Rightarrow S^*\ \underline{e}\ p)$

from the antecedent

$\vdash (S\ \underline{e}\ p \Rightarrow S^2\ \underline{e}\ p)$ (E16)

$\therefore \vdash (p \Rightarrow S^2\ \underline{e}\ p)$ also $(p \Rightarrow \{\text{<empty>}\}\ \underline{e}\ p)$

and by induction $\vdash (p \Rightarrow S^n\ \underline{e}\ p)$

hence $\vdash (p \Rightarrow \forall n\ (S^n\ \underline{e}\ p))$ since n does not occur in p

---

$$S^* \underline{e}\ p = \forall s\ s \in S^*(s \underline{e}\ p)$$

$$= \forall s((\exists n(s \in S^n)) \Rightarrow s \underline{e}\ p)$$

$$= \forall s \forall n(s \in S^n \Rightarrow s \underline{e}\ p)$$

$$= \forall n(\forall s \in S^n(s \underline{e}\ p)$$

$$= \forall n(S^n \underline{e}\ p)$$

$$S^* \underline{a}\ p = \overline{S^* \underline{e}\ \bar{p}}$$

$$= \overline{\forall n(S^n \underline{e}\ \bar{p})}$$

$$= \exists\ n(\overline{S^n \underline{e}\ \bar{p}})$$

$$= \exists\ n(S^n \underline{a}\ p)$$

$$\vdash (p_{n+1} \Rightarrow p_0 \vee S \underline{a} p_n) \Rightarrow \vdash (p_n \Rightarrow S^* \underline{a} p_0)$$

from antecedent $\quad \vdash (p_1 \Rightarrow p_0 \vee S \underline{a} p_0)$

$$\text{hence } \vdash (S \underline{a} p_1 \Rightarrow S \underline{a} p_0 \vee S^2 \underline{a} p_0)$$

$$\text{and } \vdash (p_2 \Rightarrow p_0 \vee S \underline{a} p_1)$$

$$\text{hence } \vdash (p_2 \Rightarrow (p_0 \vee S \underline{a} p_0 \vee S^2 \underline{a} p_0)$$

and by induction $\quad \vdash (p_n \Rightarrow p_0 \vee S \underline{a} p_0 \vee S^2 \underline{a} p_0 \vee \ldots\ldots S^n \underline{a} p_0)$

$$\therefore \vdash (p_n \Rightarrow \exists n (S^n \underline{a} p_0))$$