Formal Descriptions of Programming Concepts, E.J. Neuhold (ed.)
North-Holland Publishing Company, (1978)

# EQUIVALENCE OF OPERATIONAL AND DENOTATIONAL SEMANTICS FOR A FRAGMENT OF PASCAL*

K.R. Apt
Mathematical Centre, Amsterdam

A fragment of PASCAL is considered in which nested systems of procedure declarations are allowed. Procedures can call parameters by value or by variable. Three semantics for the fragment are considered -two denotational ones and one operational and proved all three to be equivalent.

## INTRODUCTION

Much work has been done on comparisons between various methods of describing the semantics of programming languages. Without aiming at completeness we mention the papers of Lauer ('71), Hoare & Lauer ('74), Milne & Strachey ('76), Milner ('76) and Stoy ('76).

In this paper we focus on two methods of description - denotational and operational semantics. The aim is to show that they are equivalent for a fragment of PASCAL. The considered language (taken from Apt & De Bakker) contains simple and subscripted variables, some simple types of expressions, assignment, sequential composition, conditionals, declaration of simple and array variables, systems of (recursive) procedure declarations and procedure calls. Procedures can call parameters by value or variable.

The paper is organized as follows. In section 2, 3, 4 and 5 we define the language and give definitions of two (different) denotational semantics of it. This part of the paper is taken almost literally from Apt & De Bakker. The two proposed denotational semantics differ only in the treatment of procedure calls. In the first approach the meaning of a procedure call is determined at the moment the call is encountered in the program text. In the second approach the meaning of each procedure call is determined already at the moment the procedure is declared. In order to ensure that in both approaches scope problems are dealt with in a correct way we make extensive use of substitution. Parameter mechanisms are treated by means of the technique of "syntactic application" by which a procedure body together with the actuals of the call are mapped to a new piece of program text. In both approaches the meaning of a procedure call is determined by a suitable combination of the least fixed point technique and the technique of syntactic application.

* This publication is registered as Report IW 71/76 of the Mathematical Centre.

Having defined both semantics we show that they are equivalent. The proof of their equivalence is presented in section 6. It uses the eqivalence between simultaneous and iterated least fixed points. An important notion which turns out to be useful is that of the depth of a statement S which corresponds to the level of nesting of procedure declarations within S.

In section 7 we define an operational semantics for our language. It is defined in the style of Cook('75)(Cook while giving his definition credits it to Lauer ('71) and Hoare & Lauer('74)) although declarations and procedure calls are treated in a different way. A careful reader will observe that our operational semantics bears a strong resemblance to the first denotational semantics. Our intention was to define denotational and operational semantics in such a way that the proof of their equivalence could reduce to the essence of the problem while avoiding tedious, but straightforward, considerations.

The proof of the equivalence (in an appropriate sense) of both semantics is presented in section 8.

In order to show that operational semantics is included in the denotational one we use computational induction applying the results proved in section 6. To prove the converse inclusion we are forced to consider nested sequences of systems of procedure declarations. By looking at the definition of the meaning of procedure calls we see that the nesting is reflected in the use of iterated least fixed points. When considering the approximations of the appropriate least fixed points we come to somewhat complicated "nested" approximations. We prove the desired inclusion by induction on the so called information sequences which we associate with the nested approximation and a statement in question.

It is to be hoped that the proofs of this paper will shed some light on the difficulties arising when considering nested systems of mutually recursive procedures.

While writing this paper the work of Stoy('76) came to our attention. Stoy proves equivalence of denotational and interpretive semantics of a language incomparable with ours. Since in his language procedures are allowed as parameters he inevitably lands in the realm of reflexive domains. Stoy's paper provides an introduction to the techniques developed by Milne and used in Milne & Strachey('76).

ACKNOWLEDGEMENT.

I express my gratitude to J.W. de Bakker who introduced me to the field and whose willingness to help and patience enabled me to enter the subject. The work on the joint paper Apt & De Bakker and further discussions with him provided the basis to this work.

## 2. DEFINITION OF THE LANGUAGE

We start with the following classes of symbols:

$SV$ = {x,y,z,...} - simple variables
$AV$ = {a,b,...} - array variables
$PV$ = {P,Q,...} - procedure variables
$C$ = {m,n,...} - integer constants.

For later use we assume these sets to be well-ordered.

We now define the classes $IV$ (integer variables), $IE$ (integer expressions), and $BE$ (boolean expressions) as follows:

$IV$ (with el. v,w,...) v::= x|a[t]
$IE$ (with el. s,t,...) t::= v|n|$t_1 + t_2$|...
$BE$ (with el. p,q,...) p::= true|false|$t_1 = t_2$|¬p|...

Finally we introduce the class of statements $S$ using auxiliary classes $R^1$, $R^2$, $R^3$, $E$ and $PB$ defined as follows:

$S ::= R^1 \mid \underline{var}\ x;R^1$ ($S \in S$)
$R^1 ::= R^2 \mid \underline{array}\ a;R^2$ ($R^1 \in R^1$)
$R^2 ::= R^3 \mid E;R^3$ ($R^2 \in R^2$)
$R^3 ::= v:=t \mid R^3_2;R^3_2 \mid \underline{if}\ p\ \underline{then}\ R^3_1\ \underline{else}\ R^3_2\ \underline{fi} \mid P(t,v) \mid \underline{begin}\ S\ \underline{end}$ ($R^3 \in R^3$)
$E ::= P \Leftarrow B \mid E_1, E_2 \mid$ ($E \in E$)
(where it is required that in each declaration
$P_1 \Leftarrow B_1, \ldots, P_n \Leftarrow B_n$ $P_i \neq P_j$ for $1 \leq i, j \leq n$, $i \neq j$)
$B ::= \langle \underline{val}\ x;\underline{var}\ y \mid S \rangle$ (where $x \neq y$) ($B \in PB$)

REMARKS.

(i) The construct $P \Leftarrow \langle \underline{val}\ x;\underline{var}\ y \mid S \rangle$ corresponds to the PASCAL procedure declaration procedure P(x:integer;var y:integer); begin S end.

(ii) Separate treatment of the begin S end case, being trivial, is always omitted in the sequel.

(iii) The above defined language is essentially a subset of PASCAL (apart from the begin S end construct which ensures that the outcome of syntactic application (see section 4) is a correct statement).

(iv) For technical reasons we allow the empty system of procedure declarations.

(v) All consideration of this paper can be trivially extended to the case of, possibly empty, lists of variable declarations, array declarations, or formal parameters (this fact is implicitly assumed in the definition of syntactic application in section 4).

## 3. STATES AND ENVIRONMENTS

Let $I = \{\mu,\nu,\ldots\}$ be the set of integers and $A = \{\alpha,\beta,\ldots\}$ an infinite well-ordered set of addresses. Let

$$\Sigma = A \to I$$

$$Var = SV \cup (AV \times I)$$

and let $Env$ be the set of all $\varepsilon$: $Var \xrightarrow[part]{} A$ such that

(i) $\varepsilon$ is 1 - 1

(ii) $\{x \in SV: \varepsilon(x) \text{ is defined}\}$ is finite

(iii) $\{a \in AV:$ for some $\mu$ $\varepsilon((a,\mu))$ is defined$\}$ is finite

(iv) for all $\mu,\nu$ and $a$ $\varepsilon((a,\mu))$ is defined if $\varepsilon((a,\nu))$ is defined

(v) $A \setminus range(\varepsilon)$ is infinite.

The elements of $\Sigma(\sigma,\sigma',\ldots)$ are called *states* and the elements of $Env$ are called *environments*.

For any $\varepsilon \in Env$, $y \in SV$ such that $y \notin dom(\varepsilon)$ and $\alpha \in A$ such that $\alpha \notin range(\varepsilon)$, we write $\varepsilon \cup \langle y,\alpha\rangle$ for the extension of $\varepsilon$ yielding $\alpha$ when applied to $y$. Similarly we write $\varepsilon \cup \langle\langle a,\nu\rangle,\alpha_\nu\rangle_{\nu\in I}$ for the extension of $\varepsilon$ yielding $\alpha_\nu$ when applied to $\langle a,\nu\rangle$ $(\nu \in I)$.

For any $\sigma \in \Sigma$, $\mu \in I$ and $\alpha \in A$ $\sigma\{\mu/\alpha\}$ is the state such that $\sigma\{\mu/\alpha\}(\beta) = \mu$ if $\beta = \alpha$ and $\sigma\{\mu/\alpha\}(\beta) = \sigma(\beta)$ otherwise.

We introduce the mappings

$L$: $IV \xrightarrow[part]{} (Env \times \Sigma \to A)$ (*left-hand-value* of an integer variable)

$R$: $IE \xrightarrow[part]{} (Env \times \Sigma \to I)$ (*right-hand-value* of an integer expression)

$T$: $BE \xrightarrow[part]{} (Env \times \Sigma \to \{T,F\})$ (*value* of a boolean expression)

defined as follows:

$L(x)(\varepsilon,\sigma) = \varepsilon(x)$, $L(a[s])(\varepsilon,\sigma) = \varepsilon(a,R(s)(\varepsilon,\sigma))$

$R(v)(\varepsilon,\sigma) = \sigma(L(v)(\varepsilon,\sigma))$,

$R(n)(\varepsilon,\sigma) = \mu$ (where $\mu$ is the integer denoted by n)

$R(t_1+t_2)(\varepsilon,\sigma) = R(t_1)(\varepsilon,\sigma) + R(t_2)(\varepsilon,\sigma),\ldots,$

$T(\underline{true})(\varepsilon,\sigma) = T$, $T(\underline{false})(\varepsilon,\sigma) = F$,

$$T(t_1=t_2)(\varepsilon,\sigma) = \begin{cases} T & \text{if } R(t_1)(\varepsilon,\sigma) = R(t_2)(\varepsilon,\sigma) \\ F & \text{if } R(t_2)(\varepsilon,\sigma) \neq R(t_2)(\varepsilon,\sigma) \end{cases}$$

$T(\neg p)(\varepsilon,\sigma) = \neg T(p)(\varepsilon,\sigma),\ldots$ .

## 4. SYNTACTIC SUBSTITUTION AND SYNTACTIC APPLICATION

In order to insure that during the semantical considerations scope problems are treated in a correct way we make extensive use of substitution.

An occurrence of a simple variable x in a statement S is *bound* whenever it is within a substatement of S of the form $\underline{var}$ x;R[1] or <$\underline{val}$ x;var y|S> or <$\underline{val}$ z;$\underline{var}$ x|S> .

An occurrence of x in S is *free* if it is not bound.

We define a substitution of an integer variable v for a simple variable x in a statement S, written as S[v/x], as follows:

$(w:=t)[v/x] \equiv w[v/x]:=t[v/x]$

$(R_1^3;R_2^3)[v/x] \equiv R_1^3[v/x];R_2^3[v/x]$

$P(t,w)[v/x] \equiv P(t[v/x],w[v/x])$

$$(\underline{\text{var}}\ y;R^1)[v/x] \equiv \begin{cases} \underline{\text{var}}\ y;R^1 & \text{, if } x \equiv y \\ \underline{\text{var}}\ y;R^1[v/x] & \text{, if } x \not\equiv y \text{ and } y \text{ not free in } v, \\ \underline{\text{var}}\ y';R^1[y'/y][v/x] & \text{, if } x \not\equiv y \text{ and } y \text{ free in } v, \text{ where } y' \\ & \quad \text{is the first simple variable such that} \\ & \quad y' \not\equiv x \text{ and } y' \text{ not free in } R^1 \text{ or } v. \end{cases} \quad (*)$$

$$\langle \underline{\text{val}}\ z;\ \underline{\text{var}}\ y|S\rangle[v/x] \equiv \begin{cases} \langle \underline{\text{val}}\ z;\underline{\text{var}}\ y|S\rangle & \text{if } x \equiv z \text{ or } x \equiv y \\ \langle \underline{\text{val}}\ z;\underline{\text{var}}\ y|S[v/z]\rangle & \text{if } x \not\equiv z,\ x \not\equiv y \text{ and } z \text{ and } y \text{ not free in } v \\ \equiv \text{similar to (*) otherwise.} & \end{cases}$$

The other cases are left to the reader.

Mutatis mutandis we define $S[b/a]$, $S[Q/P]$ and $S[\bar{Q}/\bar{P}]$ (where $\bar{Q}$ and $\bar{P}$ are sequences of procedure variables). By convention each occurrence of $P_i$ $(1 \leq i \leq n)$ in E or $E;R^3$, where $E = \langle P_i \Leftarrow B_i \rangle_{i=1}^{n}$, is *bound*.

In order to insure that the parameter mechanisms are dealt with in a correct way while defining a meaning of procedure calls we make use of the technique of "syntactic application".

For each procedure body B we define its syntactic application $B[t,v]$ to the actuals t and v (corresponding appropriately to the formal value and formal variable paramater) as follows:

$(\langle \underline{\text{val}}\ x;\underline{\text{var}}\ y|S\rangle)[t,z] \equiv \underline{\text{var}}\ u;u:=t;\underline{\text{begin}}\ S[u/x][z/y]\underline{\text{end}},$

$(\langle \underline{\text{val}}\ x;\underline{\text{var}}\ y|S\rangle\ [t,a[s]] \equiv \underline{\text{var}}\ u_1,u_2;u_1:=t;u_2:=s;\underline{\text{begin}}\ S[u_1/x][a[u_2]/y]\underline{\text{end}},$

where it is required that u is the first variable $\not\equiv$ x,y and not free in S, t or z (analogously for $u_1,u_2$). Observe that

(i) this definition implies that the actual value parameter t is indeed evaluated before execution of S;

(ii) the precaution with the fresh u is necessary since a definition like $\underline{\text{var}}\ x;x:=t;\ldots$ might give a clash between the local x and possible occurrences of x in the actual t (cf. ALGOL 60 report, 4.7.3.2 or Jensen & Wirth ('74));

(iii) the two possibilities for the actual variable parameter v are
- $v \equiv z$, a simple variable. Call-by-variable then coincides with the ALGOL 60 call-by-name.
- $v \equiv a[s]$, a subscripted variable. Then s is evaluated (and stored in $u_2$) before execution of S.

## 5. DENOTATIONAL SEMANTICS

Let $H = IE \times IV \to (Env \times \Sigma \xrightarrow[\text{part}]{} \Sigma)$. For $\eta,\eta' \in H$ define

$$\eta \subseteq \eta' \quad \text{iff} \quad \forall t,v(\ \eta(t,v) \subseteq \eta'(t,v)).$$

$\subseteq$ naturally extends to a partial ordering on $H^n$ $(n \geq 0)$.

If $\Phi\colon H^n \to H^n$ then $\mu\Phi$ denotes the least element $\bar{\eta}$ of $H^n$ such that $\Phi(\bar{\eta}) = \bar{\eta}$. $\mu\Phi$ exists if $\Phi$ is monotone. Let $\Theta = PV \to H$. For each $\theta \in \Theta$, $\bar{\eta} = (\eta_1,\ldots,\eta_n) \in H^n$ and $\bar{P} = (P_1,\ldots,P_n)$ where $P_1,\ldots,P_n$ are some different procedure variables, let

$$\theta\{\bar{\eta}/\bar{P}\}(P) = \begin{cases} \eta_i & \text{if } P \equiv P_i \\ \theta(P) & \text{otherwise.} \end{cases}$$

We now define $M\colon E \times S \to (\Theta \to (EVV \times \Sigma \xrightarrow[\text{part}]{} \Sigma))$ as follows:

$$M(E|v:=t)(\theta)(\varepsilon,\sigma) = \sigma\{R(t)(\varepsilon,\sigma)/L(v)(\varepsilon,\sigma)\}$$

$$M(E|R_1^3;R_2^3)(\theta)\ (\varepsilon,\sigma) = M(E|R_2^3)(\theta)(\varepsilon,M(E|R_1^3)(\theta)(\varepsilon,\sigma))$$

$$M(E|\underline{\text{if}}\ p\ \underline{\text{then}}\ R_1^3\ \underline{\text{else}}\ R_2^3\ \underline{\text{fi}})(\theta)(\varepsilon,\sigma) = \begin{cases} M(E|R_1^3)(\theta)(\varepsilon,\sigma) & \text{if } T(p)(\varepsilon,\sigma) = T \\ M(E|R_2^3)(\theta)(\varepsilon,\sigma) & \text{if } T(p)(\varepsilon,\sigma) = F \end{cases}$$

$$M(E|\underline{\text{var}}\ x;R^1)(\theta)(\varepsilon,\sigma) = M(E|R^1[y/x])(\theta)(\varepsilon \cup \langle y,\alpha\rangle,\sigma),$$

(*) where y is the first variable $\in SV$ not in dom($\varepsilon$), and $\alpha$ the first address not in range($\varepsilon$)

$$M(E|\underline{\text{array}}\ a;R^2)(\theta)(\varepsilon,\sigma) = M(E|R^2[b/a])(\theta)(\varepsilon \cup \langle\langle b,\nu\rangle,\alpha_\nu\rangle_{\nu\in I},\sigma)$$

(**) where b is the first array variable such that no $\langle b,\nu\rangle$ is in dom($\varepsilon$), and where the $\alpha_\nu$ are chosen in some (unspecified but) unique way from $A\backslash$range($\varepsilon$)

$$M(E|P_1 \Leftarrow B_1,\ldots,P_n \Leftarrow B_n;R^3)(\theta)(\varepsilon,\sigma) =$$
$$M(E,Q_1 \Leftarrow B_1[\bar{Q}/\bar{P}],\ldots,Q_n \Leftarrow B_n[\bar{Q}/\bar{P}]|R^3[\bar{Q}/\bar{P}])(\theta)(\varepsilon,\sigma)$$

where $\bar{Q} = (Q_1,\ldots,Q_n)$, $\bar{P} = (P_1,\ldots P_n)$ and $Q_1,\ldots,Q_n$ are the first variables $\in PV$ such that for each $j = 1,\ldots,n$ $Q_j$ does not occur in $E,P_1 \Leftarrow B_1,\ldots,P_n \Leftarrow B_n$ or $R^3$

$$M(E|P(t,v))(\theta)(\varepsilon,\sigma) = \theta\{\mu\Phi^{E,\theta}/\bar{P}\}(P)(t,v)(\varepsilon,\sigma)$$

where $E = \langle P_i \Leftarrow B_i\rangle_{i=1}^n$, $\bar{P} = (P_1,\ldots P_n)$ and $\Phi^{E,\theta}\colon H^n \to H^n$ is defined as $\Phi^{E,\theta}(\bar{\eta}) = (\Phi_1^{E,\theta}(\bar{\eta}),\ldots,\Phi_n^{E,\theta}(\bar{\eta}))$ where for $i = 1,\ldots,n$

$$\Phi_i^{E,\theta}(\bar{\eta}) = \lambda t'\lambda v'M(\ |B_i[t',v'])(\theta\{\bar{\eta}/\bar{P}\})$$

$\Phi^{E,\theta}$ is clearly monotone, so $\mu\Phi^{E,\theta}$ exists.

Observe that if $P \not\equiv P_i$ for $i = 1,\ldots,n$ then simply

$$M(E|P(t,v))(\theta)(\varepsilon,\sigma) = \theta(P)(t,v)(\varepsilon,\sigma).$$

We now define a function $M_0$: $S \to (\Theta \to (Env \times \Sigma \xrightarrow[\text{part}]{} \Sigma))$ which describes a meaning of a statement S in a different way.

$$M_0(v:=t)(\theta)(\varepsilon,\sigma) = \sigma\{R(t)(\varepsilon,\sigma)/L(v)(\varepsilon,\sigma)\}$$

$$M_0(R_1^3;R_2^3)(\theta)(\varepsilon,\sigma) = M_0(R_2^3)(\theta)(\varepsilon,M_0(R_1^3)(\theta)(\varepsilon,\sigma))$$

$$M_0(\underline{\text{if}}\ p\ \underline{\text{then}}\ R_1^3\ \underline{\text{else}}\ R_2^3\ \underline{\text{fi}})(\theta)(\varepsilon,\sigma) = \begin{cases} M_0(R_1^3)(\theta)(\varepsilon,\sigma) & \text{if } T(p)(\varepsilon,\sigma) = T \\ M_0(R_2^3)(\theta)(\varepsilon,\sigma) & \text{if } T(p)(\varepsilon,\sigma) = F \end{cases}$$

$$M_0(\underline{\text{var}}\ x;R^1)(\theta)(\varepsilon,\sigma) = M_0(R^1[y/x])(\theta)(\varepsilon \cup \langle y,\alpha\rangle,\sigma)$$
where y and $\alpha$ are like in (*)

$$M_0(\underline{\text{array}}\ a;R^2)(\theta)(\varepsilon,\sigma) = M_0(R^2[b/a])(\theta)(\varepsilon \cup \langle\langle b,\nu\rangle,\alpha_\nu\rangle_{\nu\in I},\sigma)$$
where $\langle b,\nu\rangle$ and $\alpha_\nu$ are like in (**)

$$M_0(E;R^3)(\theta)(\varepsilon,\sigma) = M_0(R^3)(\theta\{\mu\Psi^{E,\theta}/\bar{P}\})(\varepsilon,\sigma)$$
where $E = \langle P_i \Leftarrow B_i\rangle_{i=1}^n$ and $\bar{P} = (P_1,\ldots,P_n)$ and $\Psi^{E,\theta}: H^n \to H^n$ is defined as $\Psi^{E,\theta}(\bar{\eta}) = (\Psi_1^{E,\theta}(\bar{\eta}),\ldots,\Psi_n^{E,\theta}(\bar{\eta}))$ where for $i = 1,\ldots,n$ $\Psi_i^{E,\theta}(\bar{\eta}) = \lambda t'\lambda v' M_0(B_i[t',v'])(\theta\{\bar{\eta}/\bar{P}\})$. $\Psi^{E,\theta}$ is clearly monotone.

$$M_0(P(t,v))(\theta)(\varepsilon,\sigma) = \theta(P)(t,v)(\varepsilon,\sigma)\ .$$

Observe that the only difference between M and $M_0$ is in the treatment of procedure declarations and procedure calls. M determines the meaning of a call only at the moment it is encountered in the program text, whereas $M_0$ determines the meaning of each call already at the moment the procedure declaration is encountered.

In the definitions of M and $M_0$ it is always assumed that $\varepsilon$ is defined for all simple and array variables, which are free in E or S.

## 6. EQUIVALENCE OF M AND $M_0$

Our first task is to prove that M and $M_0$ are equivalent in the sense of the following theorem:

THEOREM 1. *For all* $E \in \mathcal{E}$, $S \in \mathcal{S}$ *and* $\theta \in \Theta$

(i) $M_0(S)(\theta) = M(\ |S)(\theta)$

(ii) $M(E|S)(\theta) = M_0(E;\underline{\text{begin}}\ S\ \underline{\text{end}})(\theta)$ .

Before proving the theorem we prove a few lemmata. We first introduce the following useful notion:

DEFINITION 1. We define d(S) (depth of a statement S) as follows:

(i) $d(\underline{\text{var}}\ x;R^1) = d(R^1)$

(ii) $d(\underline{\text{array}}\ a;R^2) = d(R^2)$

(iii) $d(E;R^3) = d(E)+d(R^3)+1$

(iv) $d(v:=t) = 0$

(v) $d(R_1^3;R_2^3) = \max(d(R_1^3),d(R_2^3))$

(vi) $d(\underline{\text{if}}\ p\ \underline{\text{then}}\ R_1^3\ \underline{\text{else}}\ R_2^3\ \underline{\text{fi}}) = \max(d(R_1^3),d(R_2^3))$

(vii) $d(\underline{\text{begin}}\ S\ \underline{\text{end}}) = d(S)$

(viii) $d(P(t,v)) = 0$

(ix) $d(E_1,E_2) = d(E_1)+d(E_2)$

(x) $d(P \Leftarrow B) = d(B)+1$

(xi) $d(\ ) = 0$ (depth of the empty system of procedure declarations is 0)

(xii) $d(<\underline{\text{val}}\ x;\underline{\text{var}}\ y|S>) = d(S)$

$d(S)$ corresponds to the level of nesting of procedure declarations within the statement S.

By $\ell(S)$ we denote the length of a statement S. Suppose that $A_1,\ldots,A_n,(n \geq 1)$ are some well-ordered sets. Then $\prec_\ell$ denotes the lexicographical well-ordering on $A_1 \times \ldots \times A_n$ i.e.

$$(a_1,\ldots,a_n) \prec_\ell (a_1',\ldots,a_n') \quad \text{iff}$$

$$\exists i(1 \leq i \leq n \wedge \forall j(1 \leq j < i \rightarrow a_j = a_j') \wedge (a_i < a_i')).$$

Observe that the definition of the meaning $M(E|S)(\theta)$ was given by the $\prec_\ell$-induction with respect to $(d(E)+d(S),\ell(S))$.

LEMMA 1. (Bekič ('69)) *Suppose that* $\Psi\colon H^{n+k} \to H^n$ *and* $\Omega\colon H^{n+k} \to H^k$ $(n,k \geq 0)$ *are monotone functions, Let for all* $\bar{\eta} \in H^n$

$$\Phi(\bar{\eta}) = \Psi(\bar{\eta},\mu[\lambda\eta_{n+1},\ldots,\eta_{n+k}\Omega(\bar{\eta},\eta_{n+1},\ldots,\eta_{n+k})]) .$$

*Then*

$$\mu(\Psi,\Omega) = (\mu\Phi,\mu(\lambda\eta_{n+1},\ldots,\eta_{n+k}\Omega(\mu\Phi,\eta_{n+1},\ldots,\eta_{n+k}))) .$$

LEMMA 2. *Let* E *and* $E' = <P_i \Leftarrow B_i>_{i=1}^n$ *be given systems of procedure declarations. Then*

$$M(E,Q_1 \Leftarrow B_1[\bar{Q}/\bar{P}],\ldots,Q_n \Leftarrow B_n[\bar{Q}/\bar{P}]|S[\bar{Q}/\bar{P}])(\theta)$$

$$= M(E,Q_1' \Leftarrow B_1[\bar{Q}'/\bar{P}],\ldots,Q_n' \Leftarrow B_n[\bar{Q}'/\bar{P}]|S[\bar{Q}'/\bar{P}])(\theta)$$

*for every* $\theta \in \Theta$, $S \in S$ *and sequences* $\bar{Q} = (Q_1,\ldots,Q_n)$ *and* $\bar{Q}' = (Q_1',\ldots,Q_n')$ *such that for* $j = 1,\ldots,n$ $Q_j$ *and* $Q_j'$ *do not occur free in* E, E' or S *and where* $\bar{P} = (P_1,\ldots,P_n)$.

PROOF. We leave it to the reader. The proof proceeds by $\prec_\ell$-induction w.r.t. $(d(E,E')+d(S),\ell(S))$ and is straightforward, though details are tedious.

LEMMA 3. *Let* $E = <P_i \Leftarrow B_i>_{i=1}^n$ *and* $E' = <P_j \Leftarrow B_j>_{j=n+1}^{n+k}$ $(n,k \geq 0)$ *be given systems of procedure declarations such that for* $j = n+1,\ldots,n+k$ $P_j$ *does not occur in* E.

*Then for all* $S \in \mathcal{S}$ *and* $\theta \in \Theta$

(i) $M(E,E'|S)(\theta) = M(E'|S)(\theta\{\mu\Phi^{E,\theta}/\bar{P}\})$ *where* $\bar{P} = (P_1,\ldots,P_n)$

(ii) $M(E|S)(\theta) = M(E,E'|S)(\theta)$ *under the assumption that for* $j = n+1,\ldots,n+k$ $P_j$ *does not occur in* S.

What we need for our considerations is the property i). However the proof of i) uses the property ii) and, what is worse, the proof of ii) uses i). We prove i) and ii) simultaneously by $\prec_\ell$-induction with respect to $(d(E,E')+d(S),\ell(S))$. The apparent circularity in the proof is avoided thanks to the observation that i) can be proved due to inductive assumption about i) and ii), whereas ii) follows from i), which at this moment is already proved, and the inductive assumption about ii).

PROOF. Let $\theta$ and S be arbitrarily fixed. Assume that i) and ii) are true for all $E_1,E_1',S_1$ and $\theta_1$ satisfying the assumptions and such that

$$(d(E_1,E_1')+d(S_1),\ell(S_1)) \prec_\ell (d(E,E')+d(S),\ell(S)).$$

We prove at first i).

We have to consider various cases depending on the form of S. All cases follow straightforwardly from the inductive assumption with the exception of two.

CASE I. S is $E";R^3$ where $E" = \langle Q_i \Leftarrow B_i' \rangle_{i=1}^m \in \mathcal{E}$ $(m \geq 0)$ and $R^3 \in \mathcal{R}^3$. Let $\bar{Q} = (Q_1,\ldots,Q_n)$. Then $M(E,E'|E";R^3)(\theta)$

= (by definition
$M(E,E',Q_1' \Leftarrow B_1'[\bar{Q}'/\bar{Q}],\ldots,Q_m' \Leftarrow B_m'[\bar{Q}'/\bar{Q}]|R^3[\bar{Q}'/\bar{Q}])(\theta)$
where $\bar{Q}' = (Q_1',\ldots,Q_m')$ and $Q_1',\ldots,Q_m'$ are first variables $\in PV$ such that for $i = 1,\ldots,m$ $Q_i'$ does not occur in E,E',E" or $R^3$

= (by inductive assumption)
$M(E',Q_1' \Leftarrow B_1'[\bar{Q}'/\bar{Q}],\ldots,Q_m' \Leftarrow B_m'[\bar{Q}'/\bar{Q}]|R^3[\bar{Q}'/\bar{Q}])(\theta\{\mu\Phi^{E,\theta}/\bar{P}\})$,
since $d(E,E',Q_1' \Leftarrow B_1'[\bar{Q}'/\bar{Q}],\ldots,Q_m' \Leftarrow B_m'[\bar{Q}'/\bar{Q}]) + d(R^3[\bar{Q}'/\bar{Q}])$
$= d(E,E',E") + d(R^3) = d(E,E') + d(E") + d(R^3) < d(E,E') + d(S)$,
so for $E_1 = E$, $E_1' = E',Q_1' \Leftarrow B_1'[\bar{Q}'/\bar{Q}],\ldots,Q_m' \Leftarrow B_m'[\bar{Q}'/\bar{Q}]$
and $S_1 = R^3[\bar{Q}'/\bar{Q}]$ i) holds

= (by lemma 2)
$M(E',Q_1' \Leftarrow B_1'[\bar{Q}"/\bar{Q}],\ldots,Q_m' \Leftarrow B_m'[\bar{Q}"/\bar{Q}]|R^3[\bar{Q}"/\bar{Q}])(\theta\{\mu\Phi^{E,\theta}/\bar{P}\})$
where $\bar{Q}" = (Q_1",\ldots,Q_m")$ and $Q_1",\ldots,Q_m"$ are the first variables $\in PV$ such that for $i = 1,\ldots,m$ $Q_i"$ does not occur in E', E" or $R^3$

= (by definition)
$M(E'|E";R^3)(\theta\{\mu\Phi^{E,\theta}/\bar{P}\})$.

CASE II. S is P(t,v) for some $P \in PV$, $t \in IE$ and $v \in IV$. Then by definition

$M(E,E'|P(t,v))(\theta) = \theta\{\mu\Phi^{E,E',\theta}/(P_1,\ldots,P_{n+k})\}(P)(t,v)$ and
$M(E'|P(t,v))(\theta\{\mu\Phi^{E,\theta}/\bar{P}\}) =$

$$= \theta\{\mu\Phi^{E,\theta}/(P_1,\ldots,P_n)\}\{\mu\Phi^{E',\theta\{\mu\Phi^{E,\theta}/\bar{P}\}}/(P_{n+1},\ldots,P_{n+k})\}(P)(t,v) .$$

Thus it is enough to prove that

(1) $$\mu\Phi^{E,E',\theta} = (\mu\Phi^{E,\theta}, \mu\Phi^{E',\theta\{\mu\Phi^{E,\theta}/\bar{P}\}}).$$

We have for all $\bar{\eta} \in H^n$ and $i = 1,\ldots,n$

$\Phi_i^{E,\theta}(\bar{\eta}) = \lambda t'\lambda v'M(\,|B_i[t',v'])(\theta\{\bar{\eta}/\bar{P}\})$

$= $ (by inductive assumption)
$\lambda t'\lambda v'M(E'|B_i[t',v'])(\theta\{\bar{\eta}/\bar{P}\})$,
since $d(E')+d(B_i[t',v']) = d(E')+d(B_i) < d(E,E')+d(S)$ ,
so for $E_1$ empty, $E_1' = E'$, $S_1 = B_i[t',v']$, $\theta_1 = \theta\{\bar{\eta}/\bar{P}\}$
ii) holds.

$= $ (by inductive assumption)
$\lambda t'\lambda v'M(\,|B_i[t',v'])(\theta\{\bar{\eta}/\bar{P}\}\{\mu\Phi^{E',\theta\{\bar{\eta}/\bar{P}\}}/(P_{n+1},\ldots,P_{n+k})\})$,
since $d(E')+d(B_i[t',v']) < d(E,E')+d(S)$, so for $E_1 = E'$,
$E_1'$ empty, $S_1 = B_i[t',v']$, $\theta_1 = \theta\{\bar{\eta}/\bar{P}\}$ i) holds.

$= $ (by definition of $\Phi^{E,E',\theta}$)
$\Phi_i^{E,E',\theta}(\bar{\eta},\mu\Phi^{E',\theta\{\bar{\eta}/\bar{P}\}}_i)$.

Define $\Psi\colon H^{n+k} \to H^n$ and $\Omega\colon H^{n+k} \to H^k$ as follows:

$$\Psi(\bar{\eta}) = (\Phi_1^{E,E',\theta}(\bar{\eta}),\ldots,\Phi_n^{E,E',\theta}(\bar{\eta}))$$

$$\Omega(\bar{\eta}) = (\Phi_{n+1}^{E,E',\theta}(\bar{\eta}),\ldots,\Phi_{n+k}^{E,E',\theta}(\bar{\eta})).$$

where $\bar{\eta} \in H^{n+k}$

So we have just proved that for all $\bar{\eta} \in H^n$

(2) $$\Phi^{E,\theta}(\bar{\eta}) = \Psi(\bar{\eta},\mu\Phi^{E',\theta\{\bar{\eta}/\bar{P}\}}).$$

Observe that for $j = 1,\ldots,k$ and for all $\eta_1,\ldots,\eta_{n+k} \in H$

$$\Phi_{n+j}^{E,E',\theta}(\eta_1,\ldots,\eta_{n+k})$$

$$= \lambda t'\lambda v'M(\,|B_{n+j}[t',v'])(\theta\{(\eta_1,\ldots,\eta_{n+k})/(P_1,\ldots,P_{n+k})\})$$

$$= \lambda t'\lambda v'M(\,|B_{n+j}[t',v'])(\theta\{(\eta_1,\ldots,\eta_n)/(P_1,\ldots,P_n)\}\{(\eta_{n+1},\ldots,\eta_{n+k})/(P_{n+1},\ldots,P_{n+k})\})$$

$$= \Phi_j^{E',\theta\{(\eta_1,\ldots,\eta_n)/(P_1,\ldots,P_n)\}}(\eta_{n+1},\ldots,\eta_{n+k}),$$

so for all $\eta_1,\ldots,\eta_{n+k} \in H$

$$\Omega(\eta_1,\ldots,\eta_{n+k}) = \Phi^{E',\theta\{(\eta_1,\ldots,\eta_n)/(P_1,\ldots,P_n)\}}(\eta_{n+1},\ldots,\eta_{n+k}) .$$

In particular for every $\bar{\eta} \in H^n$

$$\mu(\lambda\eta_{n+1},\ldots,\eta_{n+k}\Omega(\bar{\eta},\eta_{n+1},\ldots,\eta_{n+k})) = \mu\Phi^{E',\theta\{\bar{\eta}/\bar{P}\}}. \tag{3}$$

By (2) and (3) we get that for all $\bar{\eta} \in H^n$

$$\Phi^{E,\theta}(\bar{\eta}) = \Psi(\bar{\eta},\mu(\lambda\eta_{n+1},\ldots,\eta_{n+k}\Omega(\bar{\eta},\eta_{n+1},\ldots,\eta_{n+k}))).$$

Since $\Psi$ and $\Omega$ are clearly monotone, by lemma 2 we get

$$\mu(\Psi,\Omega) = (\mu\Phi^{E,\theta},\mu(\lambda\eta_{n+1},\ldots,\eta_{n+k}\Omega(\mu\Phi^{E,\theta},\eta_{n+1},\ldots,\eta_{n+k}))). \tag{4}$$

But by definition for all $\eta_1,\ldots,\eta_{n+k} \in H$

$$(\Psi(\eta_1,\ldots,\eta_{n+k}),\Omega(\eta_1,\ldots,\eta_{n+k})) = \Phi^{E,E',\theta}(\eta_1,\ldots,\eta_{n+k}),$$

so

$$\mu(\Psi,\Omega) = \mu\Phi^{E,E',\theta}. \tag{5}$$

Now by (4), (5) and (3) we get (1).

We now prove ii).
Again all cases are straightforward with the exception of the same two ones.

CASE I. S is $E";R^3$ where $E" = \langle Q_j \Leftarrow B'_j\rangle_{j=1}^m$ and $R^3 \in \mathcal{R}^3$. Let $\bar{Q} = (Q_1,\ldots,Q_m)$.
$M(E|E";R^3)(\theta)$

= (by definition)
$M(E,Q'_1\Leftarrow B'_1[\bar{Q}'/\bar{Q}],\ldots,Q'_m\Leftarrow B'_m[\bar{Q}'/\bar{Q}]|R^3[\bar{Q}'/\bar{Q}])(\theta)$
where $\bar{Q}' = (Q'_1,\ldots,Q'_m)$ and $Q'_1,\ldots,Q'_m$ are the first variables $\in PV$ such that for $i = 1,\ldots,m$ $Q'_i$ does not occur in E,E' or $R^3$.

= (by lemma 2)
$M(E,Q''_1\Leftarrow B'_1[\bar{Q}''/\bar{Q}],\ldots,Q''_m\Leftarrow B'_m[\bar{Q}''/\bar{Q}]|R^3[\bar{Q}''/\bar{Q}])(\theta)$
where $\bar{Q}'' = (Q''_1,\ldots,Q''_m)$ and $Q''_1,\ldots,Q''_m$ are the first variables $\in PV$ such that for $i = 1,\ldots,m$ $Q''_i$ does not occur in E,E',E" or $R^3$.

= (by inductive assumption)
$M(E,E',Q''_1\Leftarrow B'_1[\bar{Q}''/\bar{Q}],\ldots,Q''_m\Leftarrow B'_m[\bar{Q}''/\bar{Q}]|R^3[\bar{Q}''/\bar{Q}])(\theta)$,
since by the choice of $Q''_1,\ldots,Q''_m$ $P_j$ $(j = n+1,\ldots,n+k)$ does not occur in

$E,Q_1''\Leftarrow B_1'[\bar{Q}''/\bar{Q}],\ldots,Q_m''\Leftarrow B_m'[\bar{Q}''/\bar{Q}]$ or $R^3[\bar{Q}''/\bar{Q}]$. Also
$d(E,E',Q_1''\Leftarrow B_1'[\bar{Q}''/\bar{Q}],\ldots,Q_m''\Leftarrow B_m'[\bar{Q}''/\bar{Q}]) + d(R^3[\bar{Q}''/\bar{Q}])$
$= d(E,E',E'') + d(R^3) < d(E,E') + d(S)$, so for
$E_1 = E, Q_1''\Leftarrow B_1'[\bar{Q}''/\bar{Q}],\ldots,Q_m''\Leftarrow B_m'[\bar{Q}''/\bar{Q}]$, $E_1' = E'$ and $S_1 = R^3[\bar{Q}''/\bar{Q}]$
ii) holds.

$=$ (by definition)
$M(E,E'|E'';R^3)(\theta)$.

CASE II. S is $P(t,v)$ for some $P \in PV$, $t \in IE$ and $v \in IV$. Then by definition

$$M(E|P(t,v))(\theta) = \theta\{\mu\Phi^{E,\theta}/(P_1,\ldots,P_n)\}(P)(t,v).$$

(i) *is already proved* for E,E' and P(t,v), so

$$\begin{aligned} &M(E,E'|P(t,v))(\theta)\\ &= M(E'|P(t,v))(\theta\{\mu\Phi^{E,\theta}/(P_1,\ldots,P_n)\})\\ &= \theta\{\mu\Phi^{E,\theta}/(P_1,\ldots,P_n)\}\\ &\quad \{\mu\Phi^{E',\theta\{\mu\Phi^{E,\theta}/(P_1,\ldots,P_n)\}}/(P_{n+1},\ldots,P_{n+k})\}(P)(t,v). \end{aligned}$$

By assumption $P \neq P_{n+j}$ for $j = 1,\ldots,k$, so clearly

$$\begin{aligned} &\theta\{\mu\Phi^{E,\theta}/(P_1,\ldots,P_n)\}(P)(t,v)\\ &= \theta\{\mu\Phi^{E,\theta}/(P_1,\ldots,P_n)\}\\ &\quad \{\mu\Phi^{E',\theta\{\mu\Phi^{E,\theta}/(P_1,\ldots,P_n)\}}/(P_{n+1},\ldots,P_{n+k})\}(P)(t,v), \end{aligned}$$

which concludes the proof.

This finishes the proof of lemma 3.

COROLLARY. *Suppose that* $E = \langle P_i\Leftarrow B_i\rangle_{i=1}^n \in E$. *Then for all* $\theta \in \Theta$, $t \in IE$, $v \in IV$ *and* $i = 1,\ldots,n$

$$M(E|P_i(t,v))(\theta) = M(E|B_i[t,v])(\theta).$$

PROOF. By definition

$$\begin{aligned} M(E|P_i(t,v))(\theta) &= (\mu\Phi^{E,\theta})_i(t,v)\\ &= (\Phi^{E,\theta}(\mu\Phi^{E,\theta}))_i(t,v)\\ &= \Phi_i^{E,\theta}(\mu\Phi^{E,\theta})(t,v)\\ &= M(\ |B_i[t,v])(\theta\{\mu\Phi^{E,\theta}/(P_1,\ldots,P_n)\}) \end{aligned}$$

$$= \text{(by lemma 3(i))}$$

$$M(E|B_i[t,v])(\theta).$$

Now we are in a position to prove theorem 1.

PROOF OF THEOREM 1.

(i) We prove it by $\prec_\ell$-induction w.r.t. $(d(E)+d(S),\ell(S))$. All cases are straightforward with the exception of the case when S is $E;R^3$ for some $E = \langle P_i \Leftarrow B_i \rangle_{i=1}^n \in \mathcal{E}$ and $R^3 \in \mathcal{R}^3$. Then

$$M_0(E;R^3)(\theta) = M_0(R^3)(\theta\{\mu\Psi^{E,\theta}/\bar{P}\}) \quad \text{where } \bar{P} = (P_1,\ldots,P_n)$$

by ind. assumption $= M_0(R^3)(\theta\{\mu\Phi^{E,\theta}/\bar{P}\})$

—— " —— $= M(\ |R^3)(\theta\{\mu\Phi^{E,\theta}/\bar{P}\})$

by lemma 3(i) $= M(E|R^3)(\theta)$

by lemma 2 $= M(Q_1 \Leftarrow B_1[\bar{Q}/\bar{P}],\ldots,Q_n \Leftarrow B_n[\bar{Q}/\bar{P}]|R^3[\bar{Q}/\bar{P}])(\theta)$

where $Q = (Q_1,\ldots,Q_n)$ and $Q_1,\ldots,Q_n$ are the first variables $\in PV$ which do not occur in E or $R^3$

by definition $= M(\ |E;R^3)(\theta)$.

(ii) Assume that $E = \langle P_i \Leftarrow B_i \rangle_{i=1}^n \in \mathcal{E}$ and let $\bar{P} = (P_1,\ldots,P_n)$. We have

$$M(E|S)(\theta) = \text{(by lemma 3(i))}$$

$$M(\ |S)(\theta\{\mu\Phi^{E,\theta}/\bar{P}\})$$

by(i) $= M(\ |S)(\theta\{\mu\Psi^{E,\theta}/\bar{P}\})$

— " — $= M_0(S)(\theta\{\mu\Psi^{E,\theta}/\bar{P}\})$

by definition $= M_0(E;\underline{\text{begin}}\ S\ \underline{\text{end}})(\theta)$.

## 7. OPERATIONAL SEMANTICS

Now we introduce an operational semantics of our language. Let $\Sigma^\omega$ denote the set of all finite or infinite sequences of states and let $^\frown$ denote concatenation of two sequences. We define a function

Comp: $S \times \Sigma \times Env \times \mathcal{E} \xrightarrow[\text{part}]{} \Sigma^\omega$ as follows (Out(S,σ,ε,E) denotes the last element of Comp(S,σ,ε,E) if that sequence is finite, and is undefined otherwise):

$$\text{Comp}(v:=t,\sigma,\varepsilon,E) = \langle\sigma\{\mathcal{R}(t)(\varepsilon,\sigma)/\mathcal{L}(v)(\varepsilon,\sigma)\}\rangle$$

$$\text{Comp}(R_1^3;R_2^3,\sigma,\varepsilon,E) = \text{Comp}(R_1^3,\sigma,\varepsilon,E)^\frown\text{Comp}(R_2^3,\text{Out}(R_1^3,\sigma,\varepsilon,E),\varepsilon,E)$$

$$\text{Comp}(\underline{\text{if}}\ p\ \underline{\text{then}}\ R_1^3\ \underline{\text{else}}\ R_2^3\ \underline{\text{fi}},\sigma,\varepsilon,E) = \begin{cases} \langle\sigma\rangle^\frown\text{Comp}(R_1^3,\sigma,\varepsilon,E) & \text{if } \mathcal{T}(p)(\varepsilon,\sigma) = T \\ \langle\sigma\rangle^\frown\text{Comp}(R_2^3,\sigma,\varepsilon,E) & \text{if } \mathcal{T}(p)(\varepsilon,\sigma) = F \end{cases}$$

$\mathrm{Comp}(P(t,v),\sigma,\varepsilon,E) = \langle\sigma\rangle^{\frown}\mathrm{Comp}(B[t,v],\sigma,\varepsilon,E)$
where $P \Leftarrow B$ is an element in the sequence E

$\mathrm{Comp}(\underline{\text{var}}\ x;R^1,\sigma,\varepsilon,E) = \langle\sigma\rangle^{\frown}\mathrm{Comp}(R^1[y/x],\sigma,\varepsilon\cup\langle y,\alpha\rangle,E)$
where y and $\alpha$ are like in (*) from section 5

$\mathrm{Comp}(\underline{\text{array}}\ a;R^2,\sigma,\varepsilon,E) = \langle\sigma\rangle^{\frown}\mathrm{Comp}(R^2[b/a],\sigma,\varepsilon\cup\langle\langle b,v\rangle,\alpha_v\rangle_{v\in I},E)$
where $\langle b,v\rangle$ and $\alpha_v$ are like in (**) from section 5

$\mathrm{Comp}(P_1\Leftarrow B_1,\ldots,P_n\Leftarrow B_n;R^3,\sigma,\varepsilon,E)$
$= \langle\sigma\rangle^{\frown}\mathrm{Comp}(R^3[\bar{Q}/\bar{P}],\sigma,\varepsilon,E,Q_1\Leftarrow B_1[\bar{Q}/\bar{P}],\ldots,Q_n\Leftarrow B_n[\bar{Q}/\bar{P}])$
where $\bar{Q} = (Q_1,\ldots,Q_n)$, $\bar{P} = (P_1,\ldots,P_n)$ and
$Q_1,\ldots,Q_n$ are the first variables $\in PV$ such that for each $j = 1,\ldots,n$ $Q_j$ do not occur in $E,P_1\Leftarrow B_1,\ldots,P_n\Leftarrow B_n$ or $R^3$.

Intuitively $\mathrm{Comp}(S,\sigma,\varepsilon,E)$ represents the sequence of successive states of the computation determined by S from the initial state $\sigma$ in the environment $\varepsilon$ and with procedure declarations E.

It is always assumed that $\varepsilon$ is defined for all simple and array variables which occur freely in E or S and that there are no procedure variables which occur freely in E;begin S end. The last assumption is clearly necessary for procedure calls.

## 8. EQUIVALENCE OF OPERATIONAL AND DENOTATIONAL SEMANTICS

Now we prove that operational and denotational semantics which we defined are equivalent. Observe that in the definition of $M(E|S)(\theta)(\varepsilon,\sigma)$ it is not required that there are no procedure variables occurring freely in E;begin S end. Thus $M(E|S)(\theta)(\varepsilon,\sigma)$ can be defined whereas $\mathrm{Comp}(S,\sigma,\varepsilon,E)$ not, so an additional assumption is necessary. What we prove is the following theorem:

THEOREM 2. *Suppose that* $E \in \mathcal{E}$, $S \in \mathcal{S}$ *and that there are no procedure variables which occur freely in* E;begin S end. *Then for all* $\varepsilon \in Env$, $\sigma \in \Sigma$ *and* $\theta \in \Theta$

$$M(E|S)(\theta)(\varepsilon,\sigma) = \mathrm{Out}(S,\sigma,\varepsilon,E).$$

*More precisely: either* $M(E|S)(\theta)(\varepsilon,\sigma)$ *and* $\mathrm{Out}(S,\sigma,\varepsilon,E)$ *are both defined and are equal or are both undefined.*

PROOF. Suppose that for some $\varepsilon \in Env$ and $\sigma \in \Sigma$ $\mathrm{Out}(S,\sigma,\varepsilon,E)$ is defined. We prove that then $\mathrm{Out}(S,\sigma,\varepsilon,E) = M(E|S)(\theta)(\varepsilon,\sigma)$.

Suppose by induction that it is true for all $\theta',S',\sigma',\varepsilon'$ and E' satisfying the assumptions and such that the length of $\mathrm{Comp}(S',\sigma',\varepsilon',E')$ is shorter than the length of $\mathrm{Comp}(S,\sigma,\varepsilon,E)$. We have to consider various cases depending on the form of S. In all of them the claim follows straightforwardly from the inductive assumption. Only the case of procedure calls is not obvious.

Suppose that S is P(t,v). Then

$$Out(P(t,v),\sigma,\varepsilon,E) = Out(B[t,v],\sigma,\varepsilon,E)$$

where P⇐B is taken from E. By inductive assumption

$$M(E|B[t,v])(\theta)(\varepsilon,\sigma)$$

is defined and equal to

$$Out(B[t,v],\sigma,\varepsilon,E).$$

By corrollary 1

$$M(E|B[t,v])(\theta)(\varepsilon,\sigma) = M(E|P(t,v))(\theta)(\varepsilon,\sigma),$$

so the claim follows.

The proof that the converse implication holds, to which the rest of the paper is devoted, is much more difficult. We shall need the following two lemmata.

LEMMA 4. *Suppose that for some* $E \in \mathcal{E}$, $\varepsilon \in Env$, $\sigma \in \Sigma$ *and* $S \in \mathcal{S}$ $Comp(S,\sigma,\varepsilon,E)$ *is a finite sequence. Then for every* E' *such that* $E,E' \in \mathcal{E}$ *and all procedure variables occurring freely in* E' *are declared in* E *the sequences* $Comp(S,\sigma,\varepsilon,E)$ *and* $Comp(S,\sigma,\varepsilon,E,E')$ *are identical.*

PROOF. We leave it to the reader. The proof proceeds by induction with respect to the length of $Comp(S,\sigma,\varepsilon,E)$.

LEMMA 5. *For every* $E \in \mathcal{E}$, $S \in \mathcal{S}$, $\theta \in \Theta$, $\bar{P} = (P_1,\ldots,P_n)$ *and* $\bar{\eta}_k \in H^n (k = 0,1,\ldots)$ *where* $n \geq 0$ *and* $\bar{\eta}_0 \subseteq \bar{\eta}_1 \subseteq \bar{\eta}_2 \ldots$

$$M(E|S)(\theta\{\bigcup_{k=0}^{\infty} \bar{\eta}_k/\bar{P}\}) = \bigcup_{k=0}^{\infty} M(E|S)(\theta\{\bar{\eta}_k/\bar{P}\})$$

PROOF. We leave it to the reader. The proof proceeds by $\prec_\ell$-induction w.r.t $(d(E) + d(S), \ell(S))$ and is straightforward. Observe that by lemma 2 we can assume that all procedure variables declared in E are different from $P_i - s (i = 1,\ldots,n)$.

Suppose now that $\theta \in \Theta$ and $E = \langle P_i \Leftarrow B_i \rangle_{i=1}^{n} \in \mathcal{E}$ $(n \geq 0)$. Up till now we have only used the fact that the function $\Phi^{E,\theta}$ is monotone. What we need now is that $\Phi^{E,\theta}$ is continuous, i.e., that

$$\Phi^{E,\theta}\left(\bigcup_{k=0}^{\infty} \bar{\eta}_k\right) = \bigcup_{k=0}^{\infty} \Phi^{E,\theta}(\bar{\eta}_k) \quad \text{for all} \quad \bar{\eta}_k \in H^n (k = 0,1,\ldots) \text{ such that } \bar{\eta}_0 \subseteq \bar{\eta}_1 \subseteq \bar{\eta}_2 \ldots$$

and this is an immediate consequence of lemma 5.

Now define $\eta_k^{E,\theta} \in H^n$ $(k \geq 0)$ as follows:

$$\eta_k^{E,\theta} = \begin{cases} \underbrace{(\emptyset,\ldots,\emptyset)}_{n\text{-times}} & \text{if } k = 0 \\ \phi^{E,\theta}(\eta_{k-1}^{E,\theta}) & \text{if } k > 0, \end{cases}$$

where $\emptyset$ is the empty function. Then, by continuity,

$$\mu\Phi^{E,\theta} = \bigcup_{k=0}^{\infty} \eta_k^{E,\theta}$$

Now assume that $\ell \geq 0$. Let for $i = 1,\ldots,\ell+1$ $E_i = \langle P_j^i \Leftarrow B_j^i \rangle_{j=1}^{n_i}$ be a system of procedure declarations and let $S \in S$.

DEFINITION 2. The sequence $E_1 \cdot \ldots \cdot E_{\ell+1} \cdot S$ (where dots signify separators and are used instead of commas in order to avoid ambiguities) is called *nested* if

(i) whenever a procedure variable P occurs freely in $E_j$ then $j > 1$ and P is declared in $E_1,\ldots,E_{j-1}$

(ii) all procedure variables which occur freely in S are declared in $E_1,\ldots,E_{\ell+1}$

(iii) for $\ell > 0$ $d(E_{\ell+1}) + d(S) < d(E_\ell) < \ldots < d(E_1)$.

Intuitively a sequence $E_1 \cdot \ldots \cdot E_\ell \cdot E_{\ell+1} \cdot S$ is nested for $j = 1,\ldots,\ell$ $E_{j+1}$ occurs in a procedure body of a procedure declared in $E_j$ and S is a statement in the scope of $E_{\ell+1}$ (in a procedure body $B_{j_0}^\ell$ $(1 \leq j_0 \leq n_\ell)$).

If we represent the nesting of procedure declarations within $E_1$ in the form of a tree (see fig. 1) a nested sequence $E_1 \cdot \ldots \cdot E_\ell \cdot E_{\ell+1} \cdot S$ (forgetting for a moment about S) will correspond to the marked subtree.

DEFINITION 3. Let $E_1 \cdot \ldots \cdot E_\ell \cdot E_{\ell+1} \cdot S$ be a nested sequence, $\theta \in \Theta$ and let $k_1,\ldots,k_\ell$ be some non-negative integers. For $i = 1,\ldots,\ell-1$ let

$$\theta_{k_1} = \theta\left\{\eta_{k_1}^{E_1,\theta} \Big/ \bar{P}_1\right\}$$

$$\theta_{\vec{k}_i,k_{i+1}} = \theta_{\vec{k}_i}\left\{\eta_{k_{i+1}}^{E_{i+1},\theta_{\vec{k}_i}} \Big/ \bar{P}_{i+1}\right\}$$

where $\bar{P}_i = (P_1^i,\ldots,P_{n_i}^i)$ and $\vec{k}_i$ stands for $k_1,\ldots,k_i$. (Strictly speaking $\theta_{\vec{k}_i}$ depends on $E_1,\ldots,E_i$. We drop indices indicating this dependence since no confusion should arise.) If $\ell = 0$ then simply $\theta_{\vec{k}_\ell}$ is $\theta$.

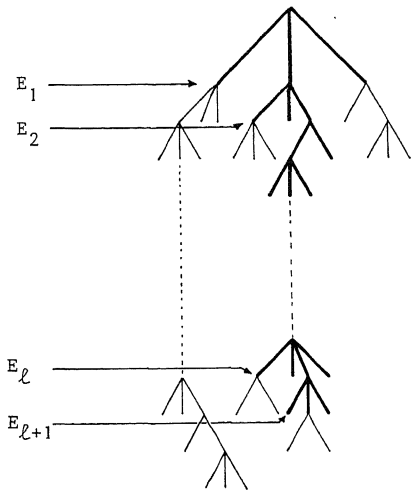


Figure 1.

A word of explanation should help. $\theta_{k_1}$, i.e.,

$$\theta\left\{{}^{E_1,\theta}_{\eta_{k_1}} \Big/ \bar{P}_1\right\}$$

assigns to procedure variables $P^1_1,\ldots,P^1_{n_1}$ the $k_1$-th approximation of their meaning computed with respect to $\theta$. $\theta^{\rightarrow}_{k_2}$, i.e.,

$$\theta\left\{{}^{E_1,\theta}_{\eta_{k_1}} \Big/ \bar{P}_1\right\}\left\{{}^{E_2,\theta_{k_1}}_{\eta_{k_2}} \Big/ \bar{P}_2\right\}$$

agrees with $\theta_{k_1}$ on all procedure variables different from $P^2_1,\ldots,P^2_{n_2}$ and assigns to $P^2_1,\ldots,P^2_{n_2}$ the $k_2$-th approximation of their meaning but now computed with respect to $\theta_{k_1}$. And so on. Summarizing, values assigned to $P^i_j - s$ where $1 \le i < i_0 \le \ell$, $1 \le j \le n_i$ enter the definition of the values assigned to $P^{i_0}_j - s$ (where $1 \le j \le n_{i_0}$).

LEMMA 6. *Assume that for some nested sequence* $E_1 \cdot \ldots \cdot E_\ell \cdot E_{\ell+1} \cdot S$, *non-negative integers* $k_1,\ldots,k_\ell$, $\theta \in \Theta$, $\varepsilon \in Env$ *and* $\sigma \in \Sigma$ $M(E_{\ell+1}|S)(\theta^{\rightarrow}_{k_\ell})(\varepsilon,\sigma)$ *is defined. Then*

$$M(E_{\ell+1}|S)(\theta^{\rightarrow}_{k_\ell})(\varepsilon,\sigma) = Out(S,\sigma,\varepsilon,E_1,\ldots,E_{\ell+1}).$$

PROOF. With every nested sequence $E_1 \cdot \ldots \cdot E_\ell \cdot E_{\ell+1} \cdot S$ and non-negative integers $k_1,\ldots,k_\ell$ we associate an *information sequence*

$$(*) \qquad (k_1,\ldots,k_\ell,\underbrace{\infty,\ldots,\infty}_{c\text{-times}},d(E_{\ell+1}) + d(S),\ \ell(S))$$

where if $\ell = 0$ then $c = d(E_{\ell+1}) + d(S)$, otherwise $c = d(E_1) - \ell$.

The following explanation should clarify the above notion. Suppose that $M(E_{\ell+1}|S)(\theta^{\rightarrow}_{k_\ell})(\varepsilon,\sigma)$ is defined. Then for any $i = 1,\ldots,\ell$ and $j = 1,\ldots,n_i$ during the execution of S starting from the state $\sigma$ in the environment $\varepsilon$ and with procedure declarations $E_1,\ldots,E_{\ell+1}$ the stack of currently active procedures will never contain more than $k_i$ copies of calls of $P^i_j$. We can view $k_i$ - s as bounds associated with appropriate levels of the marked tree corresponding to the nested sequence $E_1 \cdot \ldots \cdot E_\ell \cdot E_{\ell+1} \cdot S$ (see fig. 2). We cannot say anything about the execution of other procedures which could be called during the execution of S. $\infty$ - s stand for the bounds associated with each other level of the nesting tree of $E_1$. On the whole there can be at most $d(E_1)$ (or $d(E_{\ell+1}) + d(S)$ if $\ell = 0$) levels of nesting, which explains the choice of c.

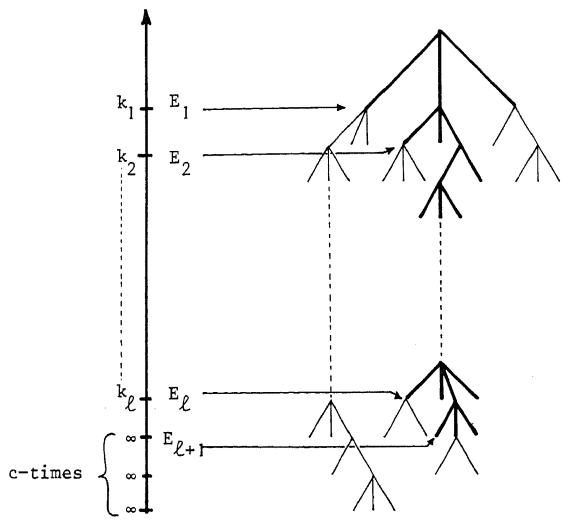


Figure 2.

If $(a_1,\ldots,a_n,a_{n+1},a_{n+2})$ and $(b_1,\ldots,b_k,b_{k+1},b_{k+2})$ are two information sequences, then by definition

$$(a_1,\ldots,a_n,a_{n+1},a_{n+2}) \prec (b_1,\ldots,b_k,b_{k+1},b_{k+2}) \quad \text{iff } n \leq k$$

and

$$(a_1,\ldots,a_n,a_{n+1},a_{n+2}) \prec_\ell (b_1,\ldots,b_n,b_{k+1},b_{k+2}).$$

So, for example $(3,3,4,\infty,8,5) \prec (3,3,\infty,\infty,1,0)$. $\prec$ is clearly a well-ordering on the information sequences.

We prove the lemma by $\prec$-induction with respect to associated information sequences. Let $a$ be the information sequence associated with $E_1\cdot\ldots\cdot E_\ell\cdot E_{\ell+1}\cdot S$ and $k_1,\ldots,k_\ell$ and assume that the claim is true for all nested sequences $E_1'\cdot\ldots\cdot E_{\ell'}'\cdot E_{\ell'+1}'\cdot S'$, non-negative integers $k_1',\ldots,k_{\ell'}'$, $\theta' \in \Theta$, $\varepsilon' \in Env$ and $\sigma' \in \Sigma$ such that $b \prec a$ where $b$ is the associated information sequence.

We have to consider various cases depending on the form of S.

CASE I. S is v:=t. Obvious.

CASE II. S is $R_1^3;R_2^3$. By definition

$$M(E_{\ell+1}|R_1^3;R_2^3)(\theta\vec{}_{k_\ell})(\varepsilon,\sigma) = M(E_{\ell+1}|R_2^3)(\theta\vec{}_{k_\ell})(\varepsilon,M(E_{\ell+1}|R_1^3)(\theta\vec{}_{k_\ell})(\varepsilon,\sigma)).$$

Observe that

$$d(E_{\ell+1}) + d(R_1^3) \leq d(E_{\ell+1}) + d(S)$$

$$d(E_{\ell+1}) + d(R_2^3) \leq d(E_{\ell+1}) + d(S),$$

so $E_1\cdot\ldots\cdot E_\ell\cdot E_{\ell+1}\cdot R_1^3$ and $E_1\cdot\ldots\cdot E_\ell\cdot E_{\ell+1}\cdot R_2^3$ are nested sequences and clearly for $i = 1,2$

$$(k_1,\ldots,k_\ell,\underbrace{\infty,\ldots,\infty}_{c_i\text{-times}},d(E_{\ell+1}) + d(R_i^3),\ \ell(R_i^3)) \prec a$$

where if $\ell = 0$ then $c_i = d(E_{\ell+1}) + d(R_i^3)$, otherwise $c_i = d(E_1) - \ell$, because

$$\ell(R_i^3) < \ell(S) \quad \text{for } i = 1,2.$$

Thus by inductive assumption

$$M(E_{\ell+1}|R_2^3)(\vec{\theta}_{k_\ell})(\varepsilon, M(E_{\ell+1}|R_1^3)(\vec{\theta}_{k_\ell})(\varepsilon,\sigma))$$

$$= M(E_{\ell+1}|R_2^3)(\vec{\theta}_{k_\ell})(\varepsilon, Out(R_1^3,\sigma,\varepsilon,E_1,\ldots,E_{\ell+1}))$$

$$= Out(R_2^3, Out(R_1^3,\sigma,\varepsilon,E_1,\ldots,E_{\ell+1}),\varepsilon,E_1,\ldots,E_{\ell+1})$$

$$= Out(R_1^3;R_2^3,\sigma,\varepsilon,E_1,\ldots,E_{\ell+1}).$$

CASE III. S is if p then $R_1^3$ else $R_2^3$ fi. Similar to case II.

CASE IV. S is var x;$R^1$. Then

$$M(E_{\ell+1}|\underline{\text{var}}\ x;R^1)(\vec{\theta}_{k_\ell})(\varepsilon,\sigma) = M(E_{\ell+1}|R^1[y/x])(\vec{\theta}_{k_\ell})(\varepsilon\cup\langle y,\alpha\rangle,\sigma)$$

where y and $\alpha$ are as in (*) from section 5. Observe that $d(R^1[y/x]) = d(S)$, so $E_1\cdot\ldots\cdot E_\ell\cdot E_{\ell+1}\cdot R^1[y/x]$ is a nested sequence and since $\ell(R^1[y/x]) < \ell(S)$

$$(k_1,\ldots,k_\ell,\underbrace{\infty,\ldots,\infty}_{c_1\text{-times}},d(E_{\ell+1}) + d(R^1[y/x]),\ \ell(R^1[y/x])) \prec \alpha$$

where if $\ell = 0$ then $c_1 = d(E_{\ell+1}) + d(R^1[y/x])$, otherwise $c_1 = d(E_1) - \ell$. So by inductive assumption

$$M(E_{\ell+1}|R^1[y/x])(\vec{\theta}_{k_\ell})(\varepsilon\cup\langle y,\alpha\rangle,\sigma)$$

$$= Out(R^1[y/x],\sigma,\varepsilon\cup\langle y,\alpha\rangle,E_1,\ldots,E_{\ell+1})$$

$$= Out(\underline{\text{var}}\ x;R^1,\sigma,\varepsilon,E_1,\ldots,E_{\ell+1}).$$

CASE V. S is array a;$R^2$. Similar to case IV.

CASE VI. S is E;$R^3$ where $E = \langle Q_j \Leftarrow B_j\rangle_{j=1}^m$. Then

$$M(E_{\ell+1}|E;R^3)(\vec{\theta}_{k_\ell})(\varepsilon,\sigma)$$

$$= M(E_{\ell+1}, Q_1'\Leftarrow B_1[\bar{Q}'/\bar{Q}],\ldots,Q_m'\Leftarrow B_m[\bar{Q}'/\bar{Q}]|R^3[\bar{Q}'/\bar{Q}])(\vec{\theta}_{k_\ell})(\varepsilon,\sigma)$$

where $\bar{Q} = (Q_1,\ldots,Q_m)$, $\bar{Q}' = (Q_1',\ldots,Q_m')$ and $Q_1',\ldots,Q_m'$ are the first variables in $PV$ which do not occur in $E_{\ell+1}$, E or $R^3$

= (by lemma 2)

$$M(E_{\ell+1}, Q_1'' \Leftarrow B_1[\bar{Q}''/\bar{Q}], \ldots, Q_m'' \Leftarrow B_m[\bar{Q}''/\bar{Q}] \mid R^3[\bar{Q}''/\bar{Q}])(\vec{\theta}_{k_\ell})(\varepsilon,\sigma)$$

where $\bar{Q}'' = (Q_1'', \ldots, \bar{Q}_m)$ and $Q_1'', \ldots, Q_m''$ are the first variables in $PV$ which do not occur in $E_1, \ldots, E_{\ell+1}$, E or $R^3$.

Observe that

(i) all procedure variables which occur freely in $Q_1'' \Leftarrow B_1[\bar{Q}''/\bar{Q}], \ldots, Q_m'' \Leftarrow B_m[\bar{Q}''/\bar{Q}]$ occur freely in E;

(ii) if a procedure variable occurs freely in $R^3[\bar{Q}''/\bar{Q}]$ then it is either $Q_i''$ for some $i \leq m$ or it occurs freely in $E;R^3$;

(iii) if $d = d(E_{\ell+1}, Q_1'' \Leftarrow B_1[\bar{Q}''/\bar{Q}]; \ldots; Q_m'' \Leftarrow B_m[\bar{Q}''/\bar{Q}]) + d(R^3[\bar{Q}''/\bar{Q}])$ then $d \leq d(E_{\ell+1}) + d(E) + d(R^3) < d(E_{\ell+1}) + d(E;R^3)$.

Thus, since $E_1 \cdot \ldots \cdot E_\ell \cdot E_{\ell+1} \cdot E;R^3$ is a nested sequence,

$E_1 \cdot \ldots \cdot E_\ell \cdot E_{\ell+1}, Q_1'' \Leftarrow B_1[\bar{Q}''/\bar{Q}], \ldots, Q_m'' \Leftarrow B_m[\bar{Q}''/\bar{Q}] \cdot R^3[\bar{Q}''/\bar{Q}]$ is a nested sequence, as well. Clearly, by (iii)

$$(k_1, \ldots, k_\ell, \underbrace{\infty, \ldots, \infty}_{c_1\text{-times}}, d, \ell(R^3[\bar{Q}''/\bar{Q}])) \prec a$$

where if $\ell = 0$ then $c_1 = d$, otherwise $c_1 = d(E_1) - \ell$, so by inductive assumption

$$M(E_{\ell+1}, Q_1'' \Leftarrow B_1[\bar{Q}''/\bar{Q}], \ldots, Q_m'' \Leftarrow B_m[\bar{Q}''/\bar{Q}] \mid R^3[\bar{Q}''/\bar{Q}])(\vec{\theta}_{k_\ell})(\varepsilon,\sigma)$$

$$Out(R^3[\bar{Q}''/\bar{Q}], \sigma, \varepsilon, E_1, \ldots, E_{\ell+1}, Q_1'' \Leftarrow B_1[\bar{Q}''/\bar{Q}], \ldots, \bar{Q}_m'' \Leftarrow B_m[\bar{Q}''/\bar{Q}]) =$$

= (by definition)

$$Out(E;R^3, \sigma, \varepsilon, E_1, \ldots, E_{\ell+1}).$$

CASE VII. S is P(t,v) for some $P \in PV$, $t \in IE$ and $v \in IV$. Since $E_1 \cdot \ldots \cdot E_\ell \cdot E_{\ell+1} \cdot P(t,v)$ is a nested sequence P is declared in $E_1, \ldots, E_{\ell+1}$.

Subcase 1. P is declared in $E_1, \ldots, E_\ell$. Thus $\ell > 0$ and for some i and j such that $0 \leq i \leq \ell-1$, $1 \leq j \leq n_i$ $P \equiv P_j^{i+1}$. Then by definition

$$M(E_{\ell+1} \mid P_j^{i+1}(t,v))(\vec{\theta}_{k_\ell})(\varepsilon,\sigma) = \vec{\theta}_{k_\ell}(P_j^{i+1}(t,v)(\varepsilon,\sigma) =$$

$$= \vec{\theta}_{k_i, k_{i+1}}(P_j^{i+1})(t,v)(\varepsilon,\sigma) =$$

$$= \left(\eta_{k_{i+1}}^{E_{i+1},\vec{\theta}_{k_i}}\right)_j (t,v)(\varepsilon,\sigma)$$

$$= \left(\phi^{E_{i+1},\vec{\theta}_{k_i}}\left(\eta_{k_{i+1}^{-1}}^{E_{i+1},\vec{\theta}_{k_i}}\right)\right)_j (t,v)(\varepsilon,\sigma)$$

because $k_{i+1} > 0$, since $\left(\eta_{k_{i+1}}^{E_{i+1},\vec{\theta}_{k_i}}\right)_j (t,v)(\varepsilon,\sigma)$ is defined

$$= \phi_j^{E_{i+1},\vec{\theta}_{k_i}}\left(\eta_{k_{i+1}^{-1}}^{E_{i+1},\vec{\theta}_{k_i}}\right)(t,v)(\varepsilon,\sigma)$$

$$= M(\,|B_j^{i+1}[t,v])\left(\vec{\theta}_{k_i}\left\{\eta_{k_{i+1}}^{E_{i+1},\vec{\theta}_{k_i}}\Big/\bar{P}_{i+1}\right\}\right)(\varepsilon,\sigma)$$

$$= M(\,|B_j^{i+1}[t,v])(\vec{\theta}_{k_i},k_{i+1}^{-1})(\varepsilon,\sigma).$$

Observe that $d(B_j^{i+1}[t,v]) < d(E_{i+1})$, so clearly $E_1\cdot\ldots\cdot E_{i+1}\cdot B_j^{i+1}[t,v]$ is a nested sequence. By definition

$$(k_1,\ldots,k_i,k_{i+1}^{-1},\underbrace{\infty,\ldots,\infty}_{c_1\text{-times}},d(B_j^{i+1}[t,v]),\ell(B_j^{i+1}[t,v])) \prec \alpha,$$

where (since $\ell > 0$) $c_1 = d(E_1) - (i+1)$. Thus by inductive assumption

$$M(\,|B_j^{i+1}[t,v])(\vec{\theta}_{k_i},k_{i+1}^{-1})(\varepsilon,\sigma)$$

$$= \mathrm{Out}(B_j^{i+1}[t,v],\sigma,\varepsilon,E_1,\ldots,E_{i+1}) =$$

$$= \mathrm{Out}(P_j^{i+1}(t,v),\sigma,\varepsilon,E_1,\ldots,E_{i+1})$$

= (by lemma 4 (which assumptions are satisfied since $E_1\cdot\ldots\cdot E_{\ell+1}\cdot S$ is a nested sequence))
$\mathrm{Out}(P_j^{i+1}(t,v),\sigma,\varepsilon,E_1,\ldots,E_{\ell+1})$.

Subcase 2. P is declared in $E_{\ell+1}$. Thus for some i such that $1 \le i \le n_{\ell+1}$ $P \equiv P_i^{\ell+1}$. This means that $d(E_{\ell+1}) > 0$, so if $\ell > 0$ then, since $E_1\cdot\ldots\cdot E_\ell\cdot E_{\ell+1}\cdot S$ is a nested sequence, there are $\ell$ natural numbers smaller than $d(E_1)$, which implies that $d(E_1) \ge \ell+1$. This means that in (*) $c > 0$. By definition

$$M(E_{\ell+1}|P_i^{\ell+1}(t,v))(\vec{\theta}_{k_\ell})(\varepsilon,\sigma) = \left(\mu\Phi^{E_{\ell+1},\vec{\theta}_{k_\ell}}\right)_i(t,v)(\varepsilon,\sigma)$$

$$= \left(\bigcup_{k=0}^{\infty} \eta_k^{E_{\ell+1},\vec{\theta}_{k_\ell}}\right)_i(t,v)(\varepsilon,\sigma)$$

$$= \left(\eta_{k+1}^{E_{\ell+1},\vec{\theta}_{k_\ell}}\right)_i(t,v)(\varepsilon,\sigma) \qquad \text{for some } k \geq 0$$

(because $\left(\bigcup_{k=0}^{\infty} \eta_k^{E_{\ell+1},\vec{\theta}_{k_\ell}}\right)_i(t,v)(\varepsilon,\sigma)$ is defined)

$$= \left(\Phi^{E_{\ell+1},\vec{\theta}_{k_\ell}}\left(\eta_k^{E_{\ell+1},\vec{\theta}_{k_\ell}}\right)\right)_i(t,v)(\varepsilon,\sigma)$$

$$= \Phi_i^{E_{\ell+1},\vec{\theta}_{k_\ell}}\left(\eta_k^{E_{\ell+1},\vec{\theta}_{k_\ell}}\right)(t,v)(\varepsilon,\sigma)$$

$$= M(|B_i^{\ell+1}[t,v])\left(\vec{\theta}_{k_\ell}\left\{\eta_k^{E_{\ell+1},\vec{\theta}_{k_\ell}}\Big/\bar{P}_{\ell+1}\right\}\right)(\varepsilon,\sigma)$$

$$= M(|B_i^{\ell+1}[t,v])(\vec{\theta}_{k_\ell,k})(\varepsilon,\sigma).$$

Observe that $d(B_i^{\ell+1}[t,v]) < d(E_{\ell+1})$, so clearly $E_1 \cdot \ldots \cdot E_{\ell+1} \cdot B_i^{\ell+1}[t,v]$ is a nested sequence. Set $c_1 = d(B_i^{\ell+1}[t,v])$ if $\ell = 0$, otherwise set $c_1 = d(E_1) - (\ell+1)$. By definition

$$(k_1,\ldots,k_\ell,k,\underbrace{\infty,\ldots,\infty}_{c_1\text{-times}},d(B_i^{\ell+1}[t,v]),\ell(B_i^{\ell+1}[t,v])) \prec \alpha$$

because $k < \infty$.

By inductive assumption

$$M(|B_i^{\ell+1}[t,v])(\vec{\theta}_{k_\ell,k})(\varepsilon,\sigma)$$

$$= \text{Out}(B_i^{\ell+1}[t,v],\sigma,\varepsilon,E_1,\ldots,E_{\ell+1})$$

$$= \text{Out}(P_i^{\ell+1}(t,v),\sigma,\varepsilon,E_1,\ldots,E_{\ell+1}).$$

This finishes the proof of lemma 6.

Now the proof of theorem 2 is immediate. Namely, suppose that for some $\varepsilon \in Env$ and $\sigma \in \Sigma$ $M(E|S)(\theta)(\varepsilon,\sigma)$ is defined. Then by assumptions of theorem 2 $E \cdot S$ is a nested sequence. Taking $\ell = 0$ and applying lemma 6 we get that

$$M(E|S)(\theta)(\varepsilon,\sigma) = Out(S,\sigma,\varepsilon,E)$$

what was to be proved.

Observe the results of this paper hold also for the appropriate fragment of ALGOL 60. If we require that the variables have a dynamic scope instead of a static one then after the appropriate changes in all three semantics (for example putting

$$M(\underline{var}\ x;R^1)(\theta)(\varepsilon,\sigma) = M(R^1)(\theta)(\varepsilon',\sigma),$$

where

$$\varepsilon'(y) = \begin{cases} \alpha & \text{if } y \equiv x \\ \varepsilon(y) & \text{if } y \not\equiv x \end{cases}$$

and $\alpha$ is the first address not in range($\varepsilon$)) the same results hold.

Infact, in both cases the same proofs work.

REFERENCES

Apt, K.R. & J.W. de Bakker. Semantics and proof theory of PASCAL procedures. Proc. 4th Coll. Automata, Languages and Programming. to appear in Lecture Notes in Comp. Science.

Bekič, H. (1969). Definable operators in general algebra and the theory of automata and flowcharts. Report IBM Laboratory, Vienna.

Cook, S.A. (1975). Axiomatic and interpretive semantics for an ALGOL fragment. Technical Report no. 79, University of Toronto.

Hoare, C.A.R. & P.E. Lauer (1974). Consistent and complementary formal theories of the semantics of programming languages. Acta Informatica 3, pp. 135-153.

Jensen, K. & N. Wirth (1974). PASCAL: user manual and report. Lecture Notes in Computer Science 18, Springer.

Lauer, P.E. (1971). Consistent formal theories of the semantics of programming languages. Report TR 25121, IBM Laboratory, Vienna.

Milne, R. & C. Strachey (1976). A theory of programming language semantics. Chapman and Hall, London and Wiley, New York.

Milner, R. (1976). Program semantics and mechanized proof. in: Foundations of Computer Science II, Part 2 (K.R. Apt, J.W. de Bakker, eds.), pp. 3-44, Mathematical Centre Tracts 82.

Stoy, J.E. (1976). The congruence of two programming language definitions. Oxford University Computing Laboratory.

DISCUSSION

Robert Tennent: Your first two semantics are not "denotational" in the sense of Scott/Strachey/Milne because the meaning of the procedure call construct is not defined only in terms of the meanings of its components; they are thus partly operational in nature.

Apt: I must admit that in the papers of Scott and Strachey I don't remember having encountered a precise formulation of what exactly the word "denotational" means. I agree with you that both "denotational" semantics I presented use syntactic substitution to obtain the meaning of procedure calls, so in your sense these are not purely denotational.

Tennent: Would you explain how your syntactic application models static scope; there seems to be no allowance for properly binding non-local identifiers of the procedure body.

Apt: Static scope is achieved by renaming local identifiers in order to avoid clashes with non-locals.

Tennent: It seems to me that it would be much simpler and clearer to use only environments, rather than environments and substitution.

Apt: Let me remark that the concept of syntactic application is a formalization of the clauses from the Algol 60 Report concerning the treatment of procedure calls and parameters. For me, syntactic substitution seems to be a natural technique to deal with static scope.

Hans Langmaack: Which of the three definitions would you recommend for the pratical programmer?

Apt: The definition which seems to be more natural is probably the operational semantics which talks about computation sequences.

Langmaack: The question now is whether the theory can be extended to allow procedures as parameters.

Apt: Within a case of operational semantics, there will be no problems, but in the denotational case the problem of self-application arises. In that case I would not make any claim about the equivalence between the two semantics.