stichting
mathematisch
centrum

MC

AFDELING INFORMATICA
(DEPARTMENT OF COMPUTER SCIENCE)

IW 97/78 JULI

K.R. APT

A SOUND AND COMPLETE HOARE-LIKE SYSTEM FOR A FRAGMENT OF PASCAL

Preprint

2e boerhaavestraat 49 amsterdam

Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O).

AMS(MOS) subject classification scheme (1970): 68A05

ACM - Computing Reviews - categories: 5.24

A sound and complete Hoare-like system for a fragment of PASCAL*)

by

K.R. Apt

ABSTRACT

A fragment of PASCAL is considered in which local declarations of simple variables, of array variables and of systems of mutually recursive parameterless procedures are allowed. A Hoare-like proof system for the fragment is presented which is proved to be both sound and complete in the sense of Cook.

KEY WORDS & PHRASES: *local declarations, substitution, denotational semantics, partial correctness, Hoare-like proof system, soundness, completeness in the sense of Cook.*

---

*) This report will be submitted for publication elsewhere.

SYSTEMS OF KNOWLEDGE

No system is any use if you merely possess it. Ownership requires operation.

No system is useful if one can only experiment with it. For a system to be useful, it must be correctly operated.

The means of operating a system must correspond with contemporary needs. It should not be imitatively traditionalistic.

Defectiveness of a system should not be confused with human shortcomings. People cannot attain certain things unless they have the means.

A system may be complete for one set of circumstances, defective for another.

Possession of a system, or any part of it, or an interest in it or in discovering one, should not be assumed to confer any licence or capacity to operate it.

Individual criticisms of a system, incapacity to operate it, or dissatisfaction with it should not be confused with any shortcoming of the system.

Consistency in a system, like inconsistency, is always more apparent than real: because what is coherent in one frame of reference may not be so in another.

These points are intended to emphasise that information and familiarisation with a system are much more important, vital and urgent than to apply existing imaginings about it to any attempt to understand or operate it.

Experience comes before understanding and before capacity to operate.

in: *Caravan of Dreams*, Idries Shah

CONTENTS

1. INTRODUCTION

We study in this paper a fragment of PASCAL containing simple and subscripted variables, assignment, sequential composition, conditionals, declarations of simple variables, of arrays and of parameterless procedures, and procedure calls. We present a Hoare-like proof system for the above language and prove its soundness and completeness in the sense of COOK[5]. Many ideas of this paper find their origin in APT & DE BAKKER[2] and APT[1]. Our results extend those of COOK[5] who proved soundness and completeness of a Hoare system for a fragment of Algol 60, in which, among others, the while statement and non-recursive procedures with parameters are allowed. We did not include the while statement into our language because it can be dealt with in the same way as in COOK[5]. On the other hand we have not allowed procedures to have parameters as this would greatly complicate the already complex arguments.

The results of this paper for a corresponding fragment of Algol 60 were claimed by GORELICK[7] who attempted to extend the results of COOK [5] to the case of a language allowing recursion. However, as we shall see in sections 7 and 12 he does not deal satisfactorily with some of the inevitably arising problems. Among others, his (and Cook's) treatment of scope problems is incorrect. CLARKE[4] proposed a modification of Gorelick's proof system which repairs the above deficiency. He claimed without proof that his system was sound and complete in the sense of Cook. The results of this paper to some extent support his claim. However, it must be noted that both GORELICK[7] and CLARKE[4] use a definition of soundness of a proof rule which is not sufficient to prove the soundness theorem. We discuss this issue in section 7.

The proper treatment of all the problems arising in proofs is much more difficult than might appear at first sight. The main difficulties stem from the inclusion of local procedure declarations in the language. To deal with the partial correctness of programs allowing recursive procedures HOARE[10] introduced a proof rule which as a premise uses the fact that a certain asserted statement is provable from another asserted statement, written as $A \vdash B$. If one admits local procedure declarations, in the proof required in the premise of the rule one has to allow the rule itself.

Because of it the notion of proof becomes extremely awkward. A rule which is defined in terms of itself is not an easy object to study and even to define its soundness becomes a highly non-trivial task. To resolve these difficulties we transformed the whole proof system into a system which uses the usual notion of proof.

Another problem arises by allowing local variable declarations. The rule of (recursive) procedure calls introduces the so-called dummy procedure variables which do not have any semantic meaning. It is by no means clear how to define the validity of asserted statements involving such variables. If one allows local variable declarations, the naive approach of quantifying over all possible meanings of the dummy procedure variables completely fails to work. This problem arises even if one disallows local procedure declarations and procedures with global variables in their bodies and in our opinion no satisfactory solution of it has been published as yet.

Allowing procedures with global variables in their bodies leads to scope problems. To resolve them we adopted a solution proposed by CLARKE[4] and independently by APT & DE BAKKER[2]: the use of substitution both in semantics and proof theory.

This is by no means a complete list of the difficulties encountered. Most of the problems are of a purely technical nature, yet we saw no possibility of reducing them.

The paper is organized as follows. In section 2 we define the subset of PASCAL we are concerned with. In sections 3 and 4 we give the definition of its semantics. These three sections are taken with minor changes from APT & DE BAKKER[2]. The proof system is defined in section 5. The system itself owes much to APT & DE BAKKER[2] and GORELICK[7]. The idea of separating a Hoare-like system from a deductive system concerning assertions comes from COOK[5].

Section 6 is devoted to a discussion concerning the notion of validity. In section 7 we indicate why various approaches to this notion, published unto this day, are incorrect. In section 8 we list some lemmata from APT[1] which are needed in the proofs of soundness and completeness theorems. The proof of the soundness theorem is presented in sections 9, 10 and 11. In section 12 we introduce and discuss Cook's notion of complete-

ness and in section 13 we prove the completeness theorem. Finally, in section 14 we discuss the problem of whether the use of an operational semantics might simplify the proofs in this paper.

2. PRELIMINARIES

To define the programming language we shall be concerned with we shall use the following classes of symbols:

SV - simple variables	with typical elements x,y,z,u,
AV - array variables	with typical elements a,b,c, .
PV - procedure variables	with typical elements P,Q.

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

Let $F = \{f_1,\ldots,f_{\ell_o}\}$ be a set of function symbols and let $RE = \{=,re_1,\ldots,re_{m_o}\}$ be a set of relation symbols. Let ℓ_i denote the arity of f_i and m_i the arity of re_i.

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

$$v ::= x \mid a[t] \qquad (v \in IV)$$

$$t ::= v \mid \ldots \mid f_i(t_1,\ldots,t_{\ell_i}) \mid \ldots \mid \underline{\text{if}}\ e\ \underline{\text{then}}\ t_1\ \underline{\text{else}}\ t_2\ \underline{\text{fi}} \qquad (t \in IE)$$

$$e ::= \underline{\text{true}} \mid \underline{\text{false}} \mid t_1 = t_2 \mid \ldots \mid re_i(t_1,\ldots,t_{m_i}) \mid \ldots \mid \neg e \mid e_1 \vee e_2 \mid \ldots \qquad (e \in BE)$$

Finally we introduce the class of statements S and the class of systems of procedure declarations E using auxiliary classes R^1, R^2 and R^3 as follows:

$$S ::= R^1 \mid \underline{\text{var}}\ x; R^1 \qquad (S \in S)$$

$$R^1 ::= R^2 \mid \underline{\text{array}}\ a; R^2 \qquad (R^1 \in R^1)$$

$$R^2 ::= R^3 \mid E; R^3 \qquad (R^2 \in R^2)$$

$$R^3 ::= v := t \mid R^3_1; R^3_2 \mid \underline{\text{if}}\ e\ \underline{\text{then}}\ R^3_1\ \underline{\text{else}}\ R^3_2\ \underline{\text{fi}} \mid P \qquad (R^3 \in R^3)$$

$$E ::= P \Leftarrow \langle S \rangle \mid E_1, E_2 \mid \qquad (E \in E)$$

(where it is required that in each declaration
$P_1 \Leftarrow \langle S_1 \rangle, \ldots, P_n \Leftarrow \langle S_n \rangle$, $P_i \not\equiv P_j$ for $1 \leq i < j \leq n$).

REMARKS.

(i) The above defined language is essentially a subset of PASCAL. The f_i are the unspecified primitive function symbols of the language and the r_i are the unspecified primitive relation symbols of the language.

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

(iii) $P \Leftarrow \langle S \rangle$ stands for the PASCAL procedure declaration procedure P; begin S end.

(iv) Arrays do not have bounds associated with them.

(v) All simple variables and all components of arrays are of the same (unspecified) type.

(vi) All considerations of this paper can be trivially extended to the case of lists of simple variable declarations or array variable declarations.

Finally we define the class of assertions AST as follows:

$$p ::= e \mid \neg p \mid p_1 \vee p_2 \mid \ldots \mid \forall x p \mid \exists x p \qquad (p \in AST)$$

Observe that only simple variables can be bound in assertions.

By var(E) where $E \in \mathbf{E}$ we denote the set of all simple variables which occur in E. It should be clear what we mean by var(S) or var(p,E,S). Similarly we define array(E).

We write $S_1 \equiv S_2$ to denote the fact that S_1 and S_2 are the same sequences of symbols.

An occurrence of a simple variable x in a statement S is *bound* whenever it is within a substatement of S of the form var x; R^1. An occurrence of x in S is *free* if it is not bound. By S[y/x] we mean a substitution of y for x in a statement S. The most important case in its definition is:

$$(\underline{\text{var}}\ y_0;R^1)[y/x] \equiv \begin{cases} \underline{\text{var}}\ y_0;R^1 & \text{, if } x \equiv y_0 \\ \underline{\text{var}}\ y_0;R^1[y/x] & \text{, if } x \not\equiv y_0 \text{ and } y \not\equiv y_0 \\ \underline{\text{var}}\ y';R^1[y'/y_0][y/x] & \text{, if } x \not\equiv y_0 \text{ and } y \equiv y_0, \end{cases}$$

where y' is the first simple variable not free in R^1, $y' \not\equiv x$ and $y' \not\equiv y_0$.

The other cases are left to the reader.

Similarly we define $S[b/a]$, $S[Q/P]$ and $S[\overline{Q}/\overline{P}]$ where $\overline{Q}$ and $\overline{P}$ are sequences of different procedure variables of the same length. By convention each occurrence of $P_i (1 \leq i \leq n)$ in E or $E;R^3$, where $E \equiv <P_i \Leftarrow <S_i>>_{i=1}^{n}$, is *bound*. In a similar way we define $p[y/x]$, $p[b/a]$, $p[\overline{y}/\overline{x}]$ and $p[\overline{b}/\overline{a}]$ where p is an assertion and $\overline{y}$ and $\overline{x}$ ($\overline{b}$ and $\overline{a}$) are sequences of different simple (array) variables of the same length. If $\overline{x} = (x_1,\ldots,x_k)$, $\overline{y} = (y_1,\ldots,y_m)$, $\overline{a} = (a_1,\ldots,a_k)$ and $\overline{b} = (b_1,\ldots,b_m)$, where $k \leq m$, then by definition $p[\overline{y}/\overline{x}] \equiv p[(y_1,\ldots,y_k)/(x_1,\ldots,x_k)]$ and $p[\overline{b}/\overline{a}] \equiv p[(b_1,\ldots,b_k)/(a_1,\ldots,a_k)]$.

By $p[t/v]$ where $t \in IE$ and $v \in IV$ we mean a substitution of t for v in an assertion p. It is defined precisely in DE BAKKER [3]. We present here only the central clause from its definition:

$$a[s][t_1/a[t_2]] \equiv \underline{\text{if}}\ s[t_1/a[t_2]] = t_2\ \underline{\text{then}}\ t_1\ \underline{\text{else}}\ a[s[t_1/a[t_2]]]\ \underline{\text{fi}}.$$

By d(S) we mean the *depth* of a statement S which we define 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) $d(\underline{\text{if}}\ e\ \underline{\text{then}}\ R_1^3\ \underline{\text{else}}\ R_2^3\ \underline{\text{fi}}) = \max\,(d(R_1^3), d(R_2^3))$

(vii) $d(P) = 0$

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

(ix) $d(P \Leftarrow <S>) = d(S) + 1$

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

d(S) corresponds to the level of nesting of procedure declarations within the statement S. By $\ell(S)$ we denote the length of the statement S.

If $A_1,\ldots,A_n$ are some well-ordered sets then $\prec_\ell$ denotes the lexicographical well-ordering on $A_1 \times\ldots\times A_n$, i.e.

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

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

For $E \in \mathcal{E}$ and $S \in \mathcal{S}$ let $c(E|S) = (d(E)+d(S),\ell(S))$. We call $c(E|S)$ the *complexity* of $E|S$. Many proofs in this paper proceed by $\prec_\ell$-induction with respect to the complexity of $E|S$.

3. STATES AND ENVIRONMENTS

An interpretation I for the primitive symbols of our language consists of a non-empty countable domain D and an assignment of functions $\underline{f}_i$ on D to function symbols f_i from $\mathcal{F}$ and relations $\underline{re}_i$ on D to relation symbols re_i from $\mathcal{RE}$. We assume I to be arbitrarily fixed throughout the paper.

Let $A = \{\delta_1,\delta_2,\ldots\}$ be an infinite well-ordered set of *addresses* whose elements are denoted by letters α,β with possible subscripts. Let

$$\Sigma = A \rightarrow D$$

$$\mathcal{Var} = \mathcal{SV} \cup (\mathcal{AV}\times D)$$

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

(i) ε is 1 - 1

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

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

(iv) for all $d_1,d_2 \in D$ and $a \in \mathcal{AV}$ $\varepsilon(a,d_1)$ is defined if $\varepsilon(a,d_2)$ is defined

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

Thus if for some $\varepsilon \in \mathcal{Env}$, $a \in \mathcal{AV}$ and $d \in D$ $\varepsilon(a,d)$ is defined then for all

$d \in D$ $\varepsilon(a,d)$ is defined and if $d_1 \neq d_2$ then $\varepsilon(a,d_1) \neq \varepsilon(a,d_2)$. This explains why we imposed on D the restriction to be countable: otherwise we would have $dom(\varepsilon) \subseteq SV$ for all $\varepsilon \in Env$, because A is assumed to be countable.

The elements of Σ are called *states* and denoted by letters $\sigma,\sigma',\ldots$ and the elements of Env are called *environments*. Roughly speaking environments map variables on addresses and states assign to adresses values from the domain D.

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 ε yielding α when applied to y. Similarly we define $\varepsilon \cup \langle \bar{y},\bar{\alpha}\rangle$ where $\bar{y}$ is a sequence of different simple variables not in $dom(\varepsilon)$ and $\bar{\alpha}$ is a sequence of different addresses not in $range(\varepsilon)$. Analogously we write $\varepsilon \cup \langle\langle a,d\rangle,\alpha_d\rangle_{d\in D}$ for the extension of ε yielding α_d when applied to (a,d).

For any $\sigma \in \Sigma$, $d \in D$ and $\alpha \in A$ $\sigma\{d/\alpha\}$ is the state such that $\sigma\{d/\alpha\}(\beta) = d$ if $\beta = \alpha$ and $\sigma\{d/\alpha\}(\beta) = \sigma(\beta)$ otherwise. We introduce the mappings

L: $IV \xrightarrow[part]{} (Env \times \Sigma \to A)$ (*left-hand-value* of an integer variable)
R: $IE \xrightarrow[part]{} (Env \times \Sigma \to D)$ (*right-hand-value* of an integer expression)
T: $AST \xrightarrow[part]{} (Env \times \Sigma \to \{\underline{T},\underline{F}\})$ (*truth value* of an assertion)

defined as follows:

$$L(x)(\varepsilon,\sigma) = \varepsilon(x), \quad L(a[s])(\varepsilon,\sigma) = \varepsilon(a,R(s)(\varepsilon,\sigma)),$$
$$R(v)(\varepsilon,\sigma) = \sigma(L(v)(\varepsilon,\sigma)),$$
$$R(f_i(t_1,\ldots,t_{\ell_i}))(\varepsilon,\sigma) = \underline{f}_i(R(t_1)(\varepsilon,\sigma),\ldots,R(t_{\ell_i})(\varepsilon,\sigma)),$$
$$T(\underline{\text{true}})(\varepsilon,\sigma) = \underline{T}, \quad T(\underline{\text{false}})(\varepsilon,\sigma) = \underline{F},$$

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

$$T(re_i(t_1,\ldots,t_{m_i}))(\varepsilon,\sigma) = \begin{cases} \underline{T} & \text{if } (R(t_1)(\varepsilon,\sigma),\ldots,R(t_{m_i})(\varepsilon,\sigma)) \in \underline{re}_i \\ \underline{F} & \text{is } (R(t_1)(\varepsilon,\sigma),\ldots,R(t_{m_i})(\varepsilon,\sigma)) \notin \underline{re}_i, \end{cases}$$

$$R(\underline{\text{if}}\ e\ \underline{\text{then}}\ t_1\ \underline{\text{else}}\ t_2\ \underline{\text{fi}})(\varepsilon,\sigma) = \begin{cases} R(t_1)(\varepsilon,\sigma) & \text{if } T(e)(\varepsilon,\sigma) = \underline{T} \\ R(t_2)(\varepsilon,\sigma) & \text{if } T(e)(\varepsilon,\sigma) = \underline{F}, \end{cases}$$

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

$$T(\exists x p)(\varepsilon,\sigma) = \begin{cases} \underline{T} & \text{if for some } d \in D\ T(p)(\varepsilon,\sigma\{d/\varepsilon(x)\}) = \underline{T} \\ \underline{F} & \text{if for all } d \in D\ T(p)(\varepsilon,\sigma\{d/\varepsilon(x)\}) = \underline{F}, \end{cases}$$

$$T(\forall x p)(\varepsilon,\sigma) = T(\neg\exists x \neg p)(\varepsilon,\sigma).$$

We say that p is *true* ($\models p$) if for all ε defined for all variables occurring in p and for all σ $T(p)(\varepsilon,\sigma) = \underline{T}$.

4. SEMANTICS

Let $H = Env \times \Sigma \xrightarrow[\text{part}]{} \Sigma$. H can be viewed as a set of all possible meanings of procedures. We denote the elements of H by η with possible subscripts. The set theoretical inclusion $\subseteq$ forms a natural partial ordering on H. $\subseteq$ naturally induces a partial ordering on H^n ($n \geq 0$). Let $\Theta = PV \to H$. Each $\theta \in \Theta$ can be viewed as a mapping assigning a meaning to each procedure variable.

For each $\overline{\eta} = (\eta_1,\ldots,\eta_n) \in H^n$ and $\overline{P} = (P_1,\ldots,P_n)$ where $P_1,\ldots,P_n$ are some different procedure variables, let

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

If $\Phi\colon H^n \to H^n$ then $\mu\Phi$ denotes the least element $\overline{\eta}$ of H^n such that $\Phi(\overline{\eta}) = \overline{\eta}$. $\mu\Phi$ exists if Φ is monotone, i.e. if Φ preserves the partial ordering on H^n.

We now define a function $M\colon E \times S \to (\Theta \to H)$ by $\prec_\ell$-induction with respect to $c(E|S)$ as follows:

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

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

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

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

where y is the first variable $\in SV$ not in dom(ε) and α the first address not in range(ε)

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

where b is the first variable $\in AV$ such that no $\langle b,d\rangle$ is in dom(ε) and where the α_d are chosen in some (unspecified but) unique way from A\range(ε)

$$M(E|\langle P_i \Leftarrow \langle S_i\rangle\rangle_{i=1}^n;R^3)(\theta)(\varepsilon,\sigma) =$$
$$M(E,\ \langle Q_i \Leftarrow \langle S_i[\overline{Q}/\overline{P}]\rangle\rangle_{i=1}^n|R^3[\overline{Q}/\overline{P}])(\theta)(\varepsilon,\sigma),$$

where $\overline{Q} = (Q_1,\ldots,Q_n)$, $\overline{P} = (P_1,\ldots,P_n)$ 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, $\langle P_i \Leftarrow \langle S_i\rangle\rangle_{i=1}^n$ or R^3

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

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

$$\Phi_i^{E,\theta}(\overline{\eta}) = M(|S_i)(\theta\{\overline{\eta}/\overline{P}\}).$$

$\Phi^{E,\theta}$ is clearly monotone, so $\mu\Phi^{E,\theta}$ exists. Observe that if $P \neq P_i$ for $i = 1,\ldots,n$ then simply $M(E|P)(\theta)(\varepsilon,\sigma) = \theta(P)(\varepsilon,\sigma)$.

In the further considerations we shall need the fact that $\Phi^{E,\theta}$ is *continuous*, i.e. that

$$\Phi^{E,\theta}(\bigcup_{k=0}^{\infty}\overline{\eta}_k) = \bigcup_{k=0}^{\infty}\Phi^{E,\theta}(\overline{\eta}_k) \quad \text{for all } \overline{\eta}_k (k=0,1,\ldots) \text{ such that } \overline{\eta}_0 \sqsubseteq \overline{\eta}_1 \sqsubseteq \ldots .$$

The proof of it is left to the reader.

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 of $\Phi^{E,\theta}$

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

The above defined semantics for our language is tuned to the proof system which will be presented in the next section. The intention is to simplify the proofs by having a semantics which is somewhat similar to the considered proof system. However, as we shall soon see, that even in spite of this choice we are faced with the necessity of extremely tedious and lengthy proofs. At the end of the paper we discuss the reasons why these proofs are so long and complicated.

We make extensive use of substitution, both to resolve scope problems and to make the semantics more similar to the proof system. The above semantics is taken from APT & DE BAKKER [2] and is further elaborated in APT [1] where an equivalence of this semantics with two other ones is proved. The only difference is that in both papers procedures can call parameters by value or by variable. Of course, the results of APT [1] hold also for the language here considered, in which only parameterless procedures are allowed.

5. PROOF SYSTEM

The proof system with which we are concerned is a modification of the usual Hoare system (see HOARE [9] and HOARE [10]) allowing to deal with nested systems of mutually recursive procedures. Before we define its axioms and proof rules we have to introduce some notation.

By an *atomic correctness formula* we mean a construct of the form $\{p\}S\{q\}$ or p, where p,q are assertions and S is a statement. By a *correctness formula* we mean a finite set of atomic correctness formulas. Letters γ,γ' denote atomic correctness formulas and letters Γ,Γ' denote

correctness formulas. By a *correctness phrase* we mean a construct of the form $\Gamma \to \langle E | \Gamma' \rangle$ where Γ and Γ' are correctness formulas and E is a system of procedure declarations ($E \in \mathcal{E}$). Γ can be viewed as a set of premises whereas Γ' can be viewed as a set of conclusions which can be deduced from Γ in the context of procedure declarations E. If Γ is empty we write $\langle E | \Gamma' \rangle$ instead of $\Gamma \to \langle E | \Gamma' \rangle$. *Axioms* in our system will be correctness phrases and *proof rules* will be constructs of the form $\frac{A_1,\ldots,A_n}{A_{n+1}}$ where $A_1,\ldots,A_{n+1}$ are correctness phrases.

The axioms and proof rules of our system H are as follows.

Axioms.

(A1) selection

$\Gamma \to \langle E | \gamma \rangle$ where $\gamma \in \Gamma$ and none of the procedure variables occurring in γ is declared in E.

(A2) assignment

$\Gamma \to \langle E | \{p[t/v]\} v:=t \{p\} \rangle$

(A3) invariance

$\Gamma \to \langle E | \{p\} P \{p\} \rangle$ where none of the variables occurring free in p occurs in E and P is not declared in E.

Proof rules.

(R1) composition

$$\frac{\Gamma \to \langle E | \{p\} R_1^3 \{q\}, \{q\} R_2^3 \{r\} \rangle}{\Gamma \to \langle E | \{p\} R_1^3 ; R_2^3 \{q\} \rangle}$$

(R2) conditional statements

$$\frac{\Gamma \to \langle E | \{p \wedge e\} R_1^3 \{q\}, \{p \wedge \neg e\} R_2^3 \{q\} \rangle}{\Gamma \to \langle E | \{p\} \underline{\text{if}}\ e\ \underline{\text{then}}\ R_1^3\ \underline{\text{else}}\ R_2^3\ \underline{\text{fi}}\ \{q\} \rangle}$$

(R3) variable declarations

$$\frac{\Gamma \to \langle E \mid \{p\} R^1[y/x] \{q\} \rangle}{\Gamma \to \langle E \mid \{p\} \underline{\text{var}}; R^1 \{q\} \rangle}$$

where y does not occur in E, p, R^1 or q.

(R4) array declarations

$$\frac{\Gamma \to \langle E \mid \{p\} R^2[b/a] \{q\} \rangle}{\Gamma \to \langle E \mid \{p\} \underline{\text{array}}\ a; R^2 \{q\} \rangle}$$

where b does not occur in E, p, R^2 or q.

(R5) procedure declarations

$$\frac{\Gamma \to \langle E, \langle P_i' \Leftarrow \langle S_i[\overline{P}'/\overline{P}] \rangle \rangle_{i=1}^n \mid \{p\} R^3[\overline{P}'/\overline{P}] \{q\} \rangle}{\Gamma \to \langle E \mid \{p\} \langle P_i \Leftarrow \langle S_i \rangle \rangle_{i=1}^n ; R^3 \{q\} \rangle}$$

where $\overline{P}' = (P_1', \ldots, P_n')$, $\overline{P} = (P_1, \ldots, P_n)$ and $P_1', \ldots, P_n'$ do not occur in $E, \langle P_i \Leftarrow \langle S_i \rangle \rangle_{i=1}^n$ or R^3.

(R6) consequence

$$\frac{\Gamma \to \langle E \mid p \to p_1,\ \{p_1\} S \{q_1\}, q_1 \to q \rangle}{\Gamma \to \langle E \mid \{p\} S \{q\} \rangle}$$

(R7) substitution

$$\frac{\Gamma \to \langle E \mid \{p\} P \{q\} \rangle}{\Gamma \to \langle E \mid \{p[\overline{y}/\overline{z}][\overline{b}/\overline{c}]\} P \{q[\overline{y}/\overline{z}][\overline{b}/\overline{c}]\} \rangle}$$

where 1°. $\overline{y}, \overline{z} \in SV$, $\overline{a}, \overline{b} \in AV$, all variables in $\overline{y}, \overline{z}, \overline{a}, \overline{b}$ are different and none of them occurs bound in p or q

2°. $\overline{z} \cap \text{var}(E) = \emptyset$, $\overline{c} \cap \text{array}(E) = \emptyset$

3°. if a variable from $\overline{y}$ occurs in E, p or q then the corresponding variable from $\overline{z}$ does not occur in q

4°. if a variable from $\overline{b}$ occurs in E, p or q then the corresponding variable from $\overline{c}$ does not occur in q.

(R8) procedure calls

$$\frac{\Gamma, \{p_1\} P_1' \{q_1\}, \ldots, \{p_n\} P_n' \{q_n\} \to \langle E, E' \mid \{p_i\} S_i' \{q_i\}_{i=0,\ldots,n} \rangle}{\Gamma \to \langle E, E' \mid \{p_0\} S_0 \{q_0\}, \{p_1\} P_1 \{q_1\}, \ldots, \{p_n\} P_n \{q_n\} \rangle}$$

where 1^{o}. $E' \equiv \langle P_i \Leftarrow \langle S_i \rangle \rangle_{i=1}^{n}$, and $P_1,\ldots,P_n$ do not occur in E
2^{o}. $P_1',\ldots,P_n'$ do not occur in Γ,E,E' or S_0
3^{o}. for i = 0,...,n $S_i' \equiv S_i[(P_1',\ldots,P_n')/(P_1,\ldots,P_n)]$.

(R9) conjunction

$$\frac{\Gamma \to \langle E \mid \{p\}P\{q\}, \{p\}P\{r\} \rangle}{\Gamma \to \langle E \mid \{p\}P\{q \wedge r\} \rangle}$$

(R10) collection

$$\frac{\Gamma \to \langle E \mid \gamma_1 \rangle, \ldots, \Gamma \to \langle E \mid \gamma_n \rangle}{\Gamma \to \langle E \mid \gamma_1, \ldots, \gamma_n \rangle}$$

(R11) selection

$$\frac{\Gamma \to \langle E \mid \Gamma \rangle}{\Gamma \to \langle E \mid \gamma \rangle} \qquad \text{where } \gamma \in \Gamma .$$

Our system is constructed in such a way that the so-called *proofs from assumptions* usually needed in the premise of a rule of (recursive) procedure calls are completely avoided. Consequently we get a proof system for which the notion of proof is simply the one used in formal logic. Two papers are known to us where a similar approach has been taken when dealing with recursion in a Hoare-like system. In SCHWARZ [12] so-called generic commands are introduced to avoid the use of proofs from assumptions in a premise of a rule. In HAREL, PNUELI & STAVI [8] *all* formulas of the system are deductions of the form $\Gamma \vdash \gamma$ for suitable Γ and γ and the notion of proof is the usual one.

Formulas of the form A $\vdash$ B appeared for the first time in SCOTT & DE BAKKER [13] where "$\vdash$" simply stands for the implication sign (e.g. on page 13: "Consider an implication $\Phi \vdash \psi$"). They appear in almost all papers dealing with proof systems concerning recursive procedures.

The use of the symbol "$\vdash$" to denote implication is in our opinion quite confusing. If we write $\frac{A \vdash B}{C \vdash D}$ to denote the fact that if B can be derived from A then D can be derived from C, we are, strictly speaking, dealing with a *metarule* of proof. On the other hand if we write $\frac{A \to B}{C \to D}$ we are dealing with a rule of proof, because the implication sign "$\to$" is a *logical* symbol whereas the

provability sign "$\vdash$" is a *metalogical* symbol. That both approaches are equivalent follows from the fact that for every reasonable logical system the deduction theorem holds: $A \vdash B$ iff $\vdash A \rightarrow B$. Once we use the implication sign "$\rightarrow$" we get a Gentzen-like proof system and we find ourselves on the familiar grounds of logic instead of wandering in an unexplored world of metarules and proofs from assumptions.

That the proofs from assumptions *can* lead to confusion we shall see later in section 7. However, to begin with, let us discuss the axioms and proof rules we have adopted in our system H.

The assignment axiom is taken from DE BAKKER [3]; it is an extension of the well-known Hoare's axiom, as it also covers the case of assignment to the subscripted variable. The invariance axiom is adopted from GORELICK [7].

The rule of variable declarations is taken from HOARE [10]. The corresponding rule of array declarations first appeared in APT & DE BAKKER [2]. Also the rule of procedure declarations comes from this paper. It appeared independently (somewhat earlier) in CLARKE [4]. The substitution rule is essentially the rule of variable substitution of GORELICK [7]. The rule of conjunction is taken from the same paper. The rule of procedure calls is actually the general case of Scott's induction (see SCOTT & DE BAKKER [13]) which for the first time was used in this framework in HOARE [10]. The collection rule is used to *collect* separately derived conclusions. This is, for example, needed in order to apply the composition rule. The selection rule is used to *select* an appropriate atomic correctness formula after the rule of procedure calls has been applied.

Observe that the only place where assertions appear as atomic correctness formulas is the rule of consequence. As long as we do not augment H with a formal system capable of proving some facts about assertions (like $p \rightarrow p_1$), we cannot meaningfully apply this rule. For example we cannot prove in H $< |\{p[t/v]\}v:=t\{\underline{true}\}>$ because we have no means to prove $< |p \rightarrow \underline{true}>$. Instead of supplementing H by a formal system concerning assertions we shall simply add to the axioms of H another axiom scheme or rather a set of axioms, namely

(A4) $\quad \Gamma \rightarrow <E|p> \quad$ for $p \in T$,

where T is a set of assertions. The resulting system will be called (H,T).

By a proof in (H,T) we mean a sequence of correctness phrases $A_1,\ldots,A_n$ such that each of the A_i $(1\leq i\leq n)$ is either an axiom (including (A4)) or follows from the preceding ones by an application of one of the proof rules. We say that a correctness phrase A is provable in (H,T), $(\vdash_{H,T}A)$, if there exists a proof in (H,T) $A_1,\ldots,A_n$ such that $A \equiv A_n$. In the subsequent sections of the paper the shall be concerned with particular properties of the system H *relative* to a set T and not in T itself. Therefore we leave further specifications of T open.

6. VALIDITY

The usual (informal) definition of the meaning (or truth) of a formula {p}S{q} is as follows: whenever p is true *before* the execution of S and S terminates, then q is true *after* the execution of S. Since S can call some procedures, we shall rather consider the formulas of the form $\langle E|\{p\}S\{q\}\rangle$ where $E \in \mathcal{E}$ and all procedure calls within S are understood with respect to E. However procedures declared in E can in turn call some other procedures which are not declared in E. Since we wish to exclude this type of situations, we are led to the following definition.

DEFINITION 1. A correctness phrase $\langle E|\Gamma\rangle$ is called *normal* if all procedure variables occurring free in Γ are declared in E and there are no procedure variables occurring free in E.

Thus if $\langle E|\Gamma\rangle$ is normal then procedures declared in E can call only those procedures which are declared in E as well.

We wish to prove that the system H provides complete information about the true normal correctness phrases, relative to T. More precisely, we want to prove that for any normal correctness phrase A if T is sound and $\vdash_{H,T}A$, then A is true, and if T is complete and A is true, then $\vdash_{H,T}A$. We shall see in section 12 that the second part of this statement cannot be proved without additional assumptions (concerning the interpretation I).

To prove a normal correctness phrase in (H,T) unfortunately one has to deal with arbitrary correctness phrases which are needed in proofs concerning calls of recursive procedures. So in our considerations we cannot restrict

our attention to normal correctness phrases.

Suppose now that T is sound (i.e. all assertions from T are true). To prove the soundness of (H,T) we introduce the notion of *validity* of a correctness phrase such that

(i) all axioms of (H,T) are valid
(ii) the rules of H preserve the validity of correctness phrases
(iii) all valid normal correctness phrases are true.

At first glance one might think that it is enough to take for the notion of validity a simple extension of the truth notion of normal correctness phrases: a correctness phrase A is valid if for all possible meanings assigned to procedures occurring free in A, A is true (where $\rightarrow$ is interpreted as implication). Unfortunately this definition does not work here, and this for several reasons.

For the purpose of the discussion below it is sufficient to restrict our attention to correctness phrases of the form $\langle E|\{p\}S\{q\}\rangle$. The definition proposed above says that $\langle E|\{p\}S\{q\}\rangle$ is valid if for all $\theta \in \Theta$ $\langle E|\{p\}S\{q\}\rangle(\theta)$ holds, where $\langle E|\{p\}S\{q\}\rangle(\theta)$ holds if for all $\varepsilon \in Env$ defined for all variables occurring free in E,p,S,q and for all states σ and σ'

$$T(p)(\varepsilon,\sigma) = \underline{T} \text{ and } M(E|S)(\theta)(\varepsilon,\sigma) = \sigma' \text{ implies } T(q)(\varepsilon,\sigma') = \underline{T}.$$

With this definitions at hand we clearly violate the first of the above three requirements. Namely the invariance axiom becomes then invalid. Observe that due to the restrictions imposed in the axiom on P, for all $\theta \in \Theta$ $M(E|P)(\theta) = \theta(P)$. Now, let p be such that for some $\varepsilon \in Env$ defined for all relevant variables and states σ and σ' $T(p)(\varepsilon,\sigma) = \underline{T}$ and $T(p)(\varepsilon,\sigma') = \underline{F}$. Unless the domain D has exactly one element -an uninteresting case- we can always find such a p. If we now assume that $\theta(P)(\varepsilon,\sigma) = \sigma'$ then we have $M(E|P)(\theta)(\varepsilon,\sigma) = \sigma'$, which shows that indeed $\langle E|\{p\}P\{p\}\rangle$ is invalid. The invariance axiom usually will be used in the form $\Gamma \rightarrow \langle E\{p\}P\{q\}\rangle$ for some non-empty Γ which does not include $\{p\}P\{p\}$, but the above argument still retains its validity.

Since we cannot delete the invariance axiom, because it is needed in

the proof of the completeness theorem, we have to refine the above proposed definition of validity so that this axiom becomes valid. Before we do this it will be helpful to have a closer look at the role played by the undeclared procedure variables.

All free occurrences of procedure variables in $\langle E|\{p\}S\{q\}\rangle$ arise from the subproofs dealing with calls of recursive procedures. Our system is so constructed that the declarations of these procedures are always recorded in E. That is why in the rule of procedure calls we used the new procedure variables $P'_1,\ldots,P'_n$ and retained the system E' while not adopting the following seemingly equivalent rule

$$\frac{\Gamma,\{p_1\}P_1\{q_1\},\ldots,\{p_n\}P_n\{q_n\} \to \langle E|\{p_i\}S_i\{q_i\}_{i=0,\ldots,n}\rangle}{\Gamma \to \langle E,E'|\{p_0\}S_0\{q_0\},\{p_1\}P_1\{q_1\},\ldots,\{p_n\}P_n\{q_n\}\rangle}$$

where in the upper line the system E' is not recorded. Thus even though these new procedure variables do not have here any semantic meaning we know at least that they are related to procedures declared in E,E'. In the proof of soundness of the rule of procedure calls we shall need to assign to these new procedure variables as a meaning the approximations $\eta_k^{E,E',\theta}$ of the meaning of procedures declared in E,E'. And this is the only place where we shall "manipulate" with the meaning of the new procedure variables. So we shall always be dealing with meanings which are related to those of procedures declared in E,E'. This suggests that in the definition of validity we do not need to quantify over all $\theta \in \Theta$ but only over those θ-s which are in some way related to the meaning of the procedures declared in E. The whole problem is now to find the class $\Theta^E \subseteq \Theta$ over which we should quantify.

Call a variable *accessible* by a procedure P if it occurs as a global variable either in the body of P or in a body of a procedure which can be called by P. One of the essential properties of a parameterless procedure is that its call cannot change the values of the simple or array variables which are global to its body but not accessible by it. We want now to use this property in the definition of Θ^E. To be more precise we introduce the following definition.

DEFINITION 2. Let $F \subseteq SV$, $G \subseteq AV$ and $\theta \in \Theta$.

(i) We call θ F-*bound* if for all $P \in PV$, $\varepsilon \in Env$ such that $F \subseteq dom(\varepsilon)$ and $\sigma, \sigma' \in \Sigma$ $\theta(P)(\varepsilon,\sigma) = \sigma'$ implies that $\sigma(\varepsilon(y)) = \sigma'(\varepsilon(y))$ for all simple variables y such that $y \in dom(\varepsilon)\backslash F$.

(ii) We call θ G-*bound* if for all $P \in PV$, $\varepsilon \in Env$ such that $G \times D \subseteq dom(\varepsilon)$ and $\sigma, \sigma' \in \Sigma$ $\theta(P)(\varepsilon,\sigma) = \sigma'$ implies that $\sigma(\varepsilon(a,d)) = \sigma'(\varepsilon(a,d))$ for all $d \in D$ and array variables a such that $(a,d) \in dom(\varepsilon)\backslash(G\times D)$.

We shall now require that all $\theta \in \Theta^E$ should be var(E)-bound and array(E)-bound. This will insure that the invariance axiom will be valid because by the assumption none of the free variables of p occurs in var(E) or array(E). In this way we shall satisfy the first of our three requirements which we imposed on the notion of validity.

The above restriction however, does not yet imply that the second requirement is satisfied. Namely if we quantify in the definition of validity over all $\theta \in \Theta$ which are var(E)-bound and array(E)-bound we run into troubles with the variable declaration rule. To see this take for R^1 the statement P;x:=x where P is not declared in E and for p the formula $x_1 = x_2$ where x_1 and x_2 are some simple variables which occur in E. Let y be the second simple variable which does not occur in E or R^1 and let Γ be $\{p\}P\{p \wedge y=y\}$.

Clearly the correctness phrase

$$\Gamma \rightarrow \langle E | \{p\}P;y:=y\{p\}\rangle$$

is valid in the above proposed sense (where, by definition $\Gamma(\theta)$ holds if $\langle |\Gamma\rangle(\theta)$ holds). On the other hand the correctness phrase

$$\Gamma \rightarrow \langle E | \{p\}\text{var } x;P;x:=x\{p\}\rangle$$

is not valid.

Indeed, let $\varepsilon \in Env$ and $\sigma,\sigma' \in \Sigma$ be such that $T(p)(\varepsilon,\sigma) = T$, $T(p)(\varepsilon,\sigma') = F$ and $dom(\varepsilon) = var(E) \cup array(E)\times D$. Let u be the first simple variable not in dom(ε) (so $y \neq u$) and let α be the first address not in range(ε). Finally, let $\theta \in \Theta$ be such that $\theta(P)(\varepsilon \cup \langle u,\alpha\rangle,\sigma) = \sigma'$ and

$\theta(P)(\varepsilon_1,\sigma_1) = \sigma_1$ whenever $y \in dom(\varepsilon_1)$. Clearly θ can be chosen to be var(E)-bound since we can also assume that $\sigma(\alpha) = \sigma'(\alpha)$.

Now obviously $\Gamma(\theta)$ holds, but $\langle E|\{p\}\underline{var}\ x;P;x:=x\{p\}\rangle(\theta)$ doesn't because

$$M(E|\underline{var}\ x;P;x:=x)(\theta)(\varepsilon,\sigma) = M(E|P;u:=u)(\theta)(\varepsilon\cup\langle u,\alpha\rangle,\sigma) =$$

$$= \theta(P)(\varepsilon\cup\langle u,\alpha\rangle,\sigma) = \sigma'.$$

Observe that the essential property of θ needed for the above argument was that $\theta(P)(\varepsilon\cup\langle y,\alpha\rangle,\sigma) \neq \theta(P)(\varepsilon\cup\langle u,\alpha\rangle,\sigma)$. In our next refinement of the definition of validity we shall quantify over a subclass of θ to which the above θ does not belong. In an analogous way we shall take care of the similar problems resulting from the array declarations rule. Let us introduce the following definition.

DEFINITION 3. Let $F \subseteq SV$, $G \subseteq AV$ and $\theta \in \Theta$.

(i) We call θ F-*invariant* if for all $P \in PV$, $\varepsilon \in Env$ such that $F \subseteq dom(\varepsilon)$, $y,y' \in SV\backslash dom(\varepsilon)$, $\alpha \in A\backslash range(\varepsilon)$ and $\sigma \in \Sigma$

$$\theta(P)(\varepsilon\cup\langle y,\alpha\rangle,\sigma) = \theta(P)(\varepsilon\cup\langle y',\alpha\rangle,\sigma).$$

(ii) We call θ G-*invariant* if for all $P \in PV$, $\varepsilon \in Env$ such that $G \times D \subseteq dom(\varepsilon)$, $b,b' \in AV$ such that $\{b,b'\}\times D \cap dom(\varepsilon) = \emptyset$, $\alpha_d \in A\backslash range(\varepsilon)$ for $d \in D$ and $\sigma \in \Sigma$

$$\theta(P)(\varepsilon\cup\langle\langle b,d\rangle,\alpha_d\rangle_{d\in D},\sigma) = \theta(P)(\varepsilon\cup\langle\langle b',d\rangle,\alpha_d\rangle_{d\in D},\sigma).$$

It turns out that if we quantify in the definition of validity only over those $\theta \in \Theta$ which are var(E)-invariant and array(E)-invariant then we insure that the variable declaration rule and array declaration rule preserve the validity of correctness phrases.

However, there are some further problems to be dealt with. Consider the consequence rule. Even if we restrict our attention in the definition of validity to those $\theta \in \Theta$ which satisfy the four so far mentioned restrictions, the consequence rule does not preserve the validity of correctness

phrases. The following simple example gives evidence for it.

Let P be a procedure variable not declared in E and let y be a simple variable which does not occur in E. Let p be the formula $x_1 = x_2$, where x_1 and x_1 are some simple variables which occur in E. The correctness phrase

$$\{p \wedge y = y\}P\{p\} \rightarrow \langle E \mid \{p \wedge y = y\}P\{p\}\rangle$$

is obviously valid whereas

$$\{p \wedge y = y\}P\{p\} \rightarrow \langle E \mid \{p\}P\{p\}\rangle$$

is not. The proof is similar to the one which concerned the variable declarations rule and we leave the details to the reader.

The problem is that for $\varepsilon \in Env$ such that $dom(\varepsilon) = var(E) \cup array(E) \times D$ and $y \notin dom(\varepsilon)$, $\sigma \in \Sigma$ and $\alpha \in A \backslash range(\varepsilon)$ the states $\theta(P)(\varepsilon,\sigma)$ and $\theta(P)(\varepsilon \cup \langle y,\alpha\rangle,\sigma)$ are in general completely unrelated to each other even if we impose on θ the four above mentioned restrictions. What we need here is the existence of a $\sigma_1 \in \Sigma$ for which $T(p)(\varepsilon,\sigma) = T(p)(\varepsilon \cup \langle y,\alpha\rangle,\sigma_1)$, such that if $\theta(P)(\varepsilon,\sigma)$ is defined, then $\theta(P)(\varepsilon \cup \langle y,\alpha\rangle,\sigma_1)$ is defined and $T(p)(\varepsilon,\theta(P)(\varepsilon,\sigma)) = T(p)(\varepsilon \cup \langle y,\alpha\rangle,\theta(P)(\varepsilon \cup \langle y,\alpha\rangle,\sigma_1))$.

This leads us to the fifth and final restriction which we impose on θ-s. Because of our way of dealing with variable declarations (by extending the environment) we cannot in general require σ_1 to be simply σ.

DEFINITION 4. Let $F \subseteq SV$ and $G \subseteq AV$.

(i) Let $\varepsilon,\varepsilon' \in Env$ and $\sigma,\sigma' \in \Sigma$. We say that the pair (ε,σ) *fits* (ε',σ') *over* (F,G) if the following four conditions are fulfilled:

1°. $F \cup G \times D \subseteq dom(\varepsilon), dom(\varepsilon')$

2°. $\sigma(\varepsilon(z)) = \sigma'(\varepsilon'(z))$ for all $z \in SV$ such that $z \in dom(\varepsilon), dom(\varepsilon')$

3°. $\sigma(\varepsilon(a,d)) = \sigma'(\varepsilon'(a,d))$ for all $a \in AV$ and $d \in D$ such that $(a,d) \in dom(\varepsilon), dom(\varepsilon')$

4°. for all $k \geq 1$ if α is the k-th address in $A \backslash range(\varepsilon)$ and α' is the k-th address in $A \backslash range(\varepsilon')$ then $\sigma(\alpha) = \sigma'(\alpha')$.

(ii) Let $\theta \in \Theta$. We say that θ *fits* (F,G) if for all $P \in PV$ and all pairs (ε,σ) and (ε',σ') such that $\varepsilon,\varepsilon' \in Env$, $\sigma,\sigma' \in \Sigma$ and (ε,σ) fits

(ε',σ') over (F,G), $\theta(P)(\varepsilon,\sigma)$ is defined iff $\theta(P)(\varepsilon',\sigma')$ is defined and if they are both defined then $(\varepsilon,\theta(P)(\varepsilon,\sigma)$ fits $(\varepsilon',\theta(P)(\varepsilon',\sigma'))$ over (F,G).

The notion of fitting is closely related to the notion of *matching relative to the empty substitution* which has been introduced in COOK [5]. In turn the notion of F-boundness and F-invariance for $F \subseteq SV$ is a modification of the notion of E-invariance for $E \in \mathcal{E}$ introduced in APT & DE BAKKER [2].

Finally, we introduce the following definition.

DEFINITION 5. Let $F \subseteq SV$, $G \subseteq AV$ and $E \in \mathcal{E}$.

(i) By $\Theta^{F,G}$ we mean the set of all $\theta \in \Theta$ which are F-bound, G-bound, F-invariant, G-invariant and which fit (F,G).

(ii) By Θ^{E} we mean the set $\Theta^{var(E),array(E)}$.

Having defined the set Θ^{E} we can return to the definition of validity.

DEFINITION 6.

(i) A correctness phrase $\Gamma \rightarrow \langle E|\Gamma'\rangle$ is *valid* if for all $\theta \in \Theta^{E}$ whenever $\Gamma(\theta)$ holds then $\langle E|\Gamma'\rangle(\theta)$ holds.

(ii) $\Gamma(\theta)$ holds if $\langle|\Gamma\rangle(\theta)$ holds.

(iii) $\langle E|\Gamma\rangle(\theta)$ holds if for all $\gamma \in \Gamma$ $\langle E|\gamma\rangle(\theta)$ holds.

(iv) $\langle E|p\rangle(\theta)$ holds if p is true.

(v) $\langle E|\{p\}S\{q\}\rangle(\theta)$ holds if for all $\varepsilon \in Env$ defined for all variables occurring in E,p,S,q and for all states σ and σ' $T(p)(\varepsilon,\sigma) = \underline{T}$ and $M(E|S)(\theta)(\varepsilon,\sigma) = \sigma'$ implies that $T(q)(\varepsilon,\sigma') = \underline{T}$.

(vi) A proof rule $\dfrac{A_1,\ldots,A_n}{A_{n+1}}$ is *sound* if the validity $A_1,..,A_n$ implies the validity of A_{n+1}.

To avoid some uninteresting considerations concerning variables which occur bound in statements or assertions, we required in the clause (v) of the above definition that ε is defined for all variables which occur in E,p,S,q instead only for those which are free.

Observe that in contrast to the notion of proof the notions of

validity, soundness, truth of an assertion and meaning of a statement are all dependent on the interpretation I.

The system H has been so constructed that each of the proof rules is of the form

$$\frac{\Gamma, \Gamma_1 \rightarrow \langle E, E' | \Gamma_2 \rangle}{\Gamma \rightarrow \langle E | \Gamma_3 \rangle} .$$

The reason for it is the fact that for any $E,E' \in \mathcal{E}$ $\theta^E \subseteq \theta^{E,E'}$ which in many proofs of soundness of a rule provides the desired link between the validity of the lower and upper line.

To achieve this we did not incorporate into H the so-called extension rule (see CLARKE [4] and APT & DE BAKKER [2]) which is of the form

$$\frac{\Gamma \rightarrow \langle E | \Gamma_1 \rangle}{\Gamma \rightarrow \langle E, E' | \Gamma_1 \rangle} ,$$

and "shifted" all extensions to the axioms of H. Note that the extension rule is not sound in the sense defined above!

It should be clear by now that with such complex definitions at hand the proofs of soundness and completeness of our system H will be long and tedious. The most complicated proofs result from the restriction in the definitions of validity to θ-s which are var(E)-invariant and array(E)-invariant. We found that it was needed not only in the proofs of soundness of the variable array declarations rules but also in the proofs of soundness of the consequence rule and the substitution rule. However, we did not find any example showing that the restriction to these θ-s is really necessary. The example given in the text is not satisfactory and we included it merely to motivate the need for introducing the notions of var(E)-invariance and array(E)-invariance. Observe at first that the variable y occurs there in Γ, so to prevent the exemplified difficulties we might simply assume in the formulation of the variable declaration rule that y does not occur in Γ. Secondly, the restriction to θ-s which fit over (var(E),array(E)) needed anyway for the consequence rule (and the substitution rule) takes care of this example, as well. If it turned out

that the restriction to θ-s which are var(E)-invariant and array(E)-invariant is not necessary some of the proofs in this paper might be considerably simplified.

7. WHY NOT PROOFS FROM ASSUMPTIONS?

Starting from HOARE [10] the so-called proofs from assumptions have been used to deal with recursion in the axiomatic systems concerning partial correctness of programs. In order to be able to prove properties of a recursive procedure according to Hoare one should adjoin to the system taking care of other features of the language a special rule (or rules) dealing with recursion leaving the other rules and axioms intact. In the case of a declaration $P \Leftarrow \langle S \rangle$ of one parameterless recursive procedure one should adjoin the rule

$$\frac{\{p\}P'\{q\} \vdash \{p\}S[P'/P]\{q\}}{\{p\}P\{q\}}$$

where P' is a procedure variable not occurring in S (so-called dummy procedure variable). The reasoning presented by this rule is the following: infer $\{p\}P\{q\}$ from the fact that $\{p\}S[P'/P]\{q\}$ can be proved (using other rules) from the assumption $\{p\}P'\{q\}$. Actually, Hoare does not use dummy procedure variables but their introduction makes it easier to distinguish between two different roles played by the procedure variable P.

In GORELICK [7], CLARKE [4] and DONAHUE [6] a proof rule of this type is used for a programming language in which among others local variable declarations are allowed. There are two problems concerning this type of rules.

First, in the proof of $\{p\}S[P'/P]\{q\}$ from $\{p\}P'\{q\}$ one applies the other rules of proof to *different* constructs than before. In the case of non-recursive programs there were no dummy procedure variables which appear here. So even if the former rules of proof have been proved to be sound in the case of a language forbidding recursion we *cannot* conclude from it that they are sound in the case when dummy procedure variables appear as possible (substatements of) programs. Actually the whole notion of soundness changes then its meaning and, as we have already seen, to find a

proper definition of it in the case of a language allowing local variable declarations is not easy. To prove the soundness of a system obtained by incorporating the above rule one has not only to prove soundness (in an appropriate sense) of this rule but also to prove anew soundness (again in an appropriate sense) of the former proof rules and axioms now used for a bigger class of programs.

Neither GORELICK [7] nor CLARKE [4] do this. The notion of soundness of a proof rule which they use actually is not sufficient to provide these proofs. Also (as noted by R. Milne) the proofs of soundness in DONAHUE [6] are incorrect. DONAHUE [6] uses a definition of validity which, roughly speaking, corresponds to our first proposal, so as we have seen he cannot succeed.

The use of the rule of recursion in the above form makes it difficult to see what actually has to be proved because one simply thinks in terms of a new rule of proof and not in terms of a new notion of proof (namely a proof from assumptions).

It should be also noted here that with the definition of soundness given in APT & DE BAKKER [2] the presented there rule of extension is not sound.

The second problem is that if one admits local procedure declarations then in the proof of $\{p\}\, S\,[P'/P]\{q\}$ from $\{p\}P'\{q\}$ needed in the premise of the rule of recursion one is forced to use this rule for some other (local) procedures. The whole notion of proof becomes then extremely clumsy (try to define it!) and difficult if not impossible to study. How should one define (and prove) the soundness of a proof rule which is defined in terms of itself?

In all four above mentioned papers local procedure declarations are allowed but the notion of proof is either not defined or it is simply too restrictive.

8. AUXILIARY LEMMATA

In this section we list some lemmata proved or indicated in APT [1] which will turn out to be needed in the proofs of soundness and completeness theorems.

LEMMA 1. *Let* $E,E' \in \mathbf{E}$ *be given systems of procedure declarations such that no procedure variable declared in* E' *occurs in* E. *Then for all* $S \in \mathbf{S}$ *and* $\theta \in \Theta$

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

where $\overline{P}$ *is the sequence of the procedure variables declared in* E.

LEMMA 2. *Let* E *and* $E' \equiv \langle P_i \Leftarrow \langle S_i \rangle \rangle_{i=1}^{n}$ *be given systems of procedure declarations. Then for all* $S \in \mathbf{S}$, $\theta \in \Theta$ *and* $Q_1,\ldots,Q_n,Q_1',\ldots,Q_n' \in \mathbf{PV}$ *such that for* $j = 1,\ldots,n$ Q_j *and* Q_j' *do not occur in* E,E' *or* S

$$M(E,Q_1 \Leftarrow \langle S_1[\overline{Q}/\overline{P}]\rangle,\ldots,Q_n \Leftarrow \langle S_n[\overline{Q}/\overline{P}]\rangle \mid S[\overline{Q}/\overline{P}])(\theta)$$

$$= M(E,Q_1' \Leftarrow \langle S_1[\overline{Q}'/\overline{P}]\rangle,\ldots,Q_n' \Leftarrow \langle S_n[\overline{Q}'/\overline{P}]\rangle \mid S[\overline{Q}'/\overline{P}])(\theta)$$

where $\overline{P} = (P_1,\ldots,P_n)$, $\overline{Q} = (Q_1,\ldots,Q_n)$ *and* $\overline{Q}' = (Q_1',\ldots,Q_n')$.

LEMMA 3. *Let* $E,E' \in \mathbf{E}$ *be given systems of procedure declarations such that no procedure variable declared in* E' *occurs in* E. *Then for all* $S \in \mathbf{S}$ *such that no procedure variable declared in* E' *occurs in* S *and all* $\theta \in \Theta$

$$M(E,E'|S)(\theta) = M(E|S)(\theta).$$

We shall need a slightly stronger version of lemma 3, namely

LEMMA 4. *Let* E *and* E' *be as in lemma* 3. *Then for all* $S \in \mathbf{S}$ *such that no procedure variable declared in* E' *occurs freely in* S *and all* $\theta \in \Theta$

$$M(E,E'|S)(\theta) = M(E|S)(\theta).$$

To prove lemma 4 we introduce the following notion.

DEFINITION 7. We define a *variant* of a statement as follows.

(i) Each $S \in \mathbf{S}$ is a *variant* of S.

(ii) If $S_1,\ldots,S_n, R^3$ are respectively *variants* of $S'_1,\ldots,S'_n,R^3_1$ and for $i = 1,\ldots,n$ Q_i does not occur in $\langle P_i \Leftarrow \langle S_i \rangle\rangle_{i=1}^n ; R^3$ then $\langle Q_i \Leftarrow \langle S_i[\bar{Q}/\bar{P}]\rangle\rangle_{i=1}^n ; R^3[\bar{Q}/\bar{P}]$, where $\bar{Q} = (Q_1,\ldots,Q_n)$ and $\bar{P} = (P_1,\ldots,P_n)$, is a *variant* of $\langle P_i \Leftarrow \langle S'_i\rangle\rangle_{i=1}^n ; R^3_1$.

(iii) If R^3_1,R^3_2 are respectively *variants* of R^3_3,R^3_4 then $R^3_1;R^3_2$ is a *variant* of $R^3_3;R^3_4$ and if e then R^3_1 else R^3_2 fi is a *variant* of if e then R^3_3 else R^3_4 fi.

(iv) If R^1_1,R^2_1 are respectively *variants* of R^1_2,R^2_2 then array a;R^2_1 is a *variant* of array a;R^2_2 and var x;R^1_1 is a *variant* of var x;R^1_2.

The following lemma holds.

LEMMA 5. *Let* $E \in \mathcal{E}$, $\theta \in \Theta$ *and let* S *be a variant of* S_1. *Then*

$$M(E|S)(\theta) = M(E|S_1)(\theta).$$

PROOF. Straightforward by induction on the structure of S using the definition of substitution and lemma 2. □

Lemma 4 is now an immediate consequence of lemmata 3 and 5. Namely let E and E' be as in lemma 3, let $S \in \mathcal{S}$ be such that no procedure declared in E' occurs freely in S and let $\theta \in \Theta$. Let S_1 be a variant of S such that no procedure declared in E' occurs in S. Then by lemma 3 $M(E,E'|S_1)(\theta) = M(E|S_1)(\theta)$, so by lemma 5 $M(E,E'|S)(\theta) = M(E|S)(\theta)$.

Finally we shall need the following lemma.

LEMMA 6. *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* $\bar{\eta}_0 \sqsubseteq \bar{\eta}_1 \sqsubseteq \ldots$,

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

Observe that the continuity of $\Phi^{E,\theta}$ is a direct consequence of lemma 6.

9. SOUNDNESS THEOREM - THE CASE OF THE VARIABLE DECLARATION RULE

Our first task is to prove that the system H is sound in the sense of the following theorem.

SOUNDNESS THEOREM. *All axioms of* H *are valid and all proof rules are sound.*

COROLLARY. *Let* T *be a set of assertions. Assume that the interpretation* I *is such that each assertion from* T *is true. Then for every correctness phrase* A *if* $\vdash_{H,T}$ A *then* A *is valid.*

Observe that the soundness of H, in contrast to the soundness of (H,T), is *independent* of I. We now proceed with the proof of the soundness theorem.

To verify the validity of the selection axiom (A1) assume that $\gamma \in \Gamma$ and that none of the procedure variables occurring in γ is declared in E. We may assume that γ is of the form $\{p\}S\{q\}$ for some assertions p,q and a statement S. Suppose that $\Gamma(\theta)$ holds for some $\theta \in \Theta^E$. Then $<|\{p\}S\{q\}>(\theta)$ holds. We are to prove that $<E|\{p\}S\{q\}>(\theta)$ holds. To this end it is enough to show that $M(|S)(\theta) = M(E|S)(\theta)$. In view of our assumptions it is an immediate consequence of lemma 3.

The assignment axiom (A2) is proved to be valid in DE BAKKER [3]. The validity of the invariance axiom (A3) is an immediate consequence of the assumption that each $\theta \in \Theta^E$ is var(E)-bound and array(E)-bound and of the following obvious lemma.

LEMMA 7. *Suppose that* $p \in AST, \sigma, \sigma' \in \Sigma$ *and assume that* $\varepsilon \in Env$ *is defined for all variables occurring in* p. *If* $\sigma(\varepsilon(y)) = \sigma'(\varepsilon(y))$ *and* $\sigma(\varepsilon(a,d)) = \sigma(\varepsilon(a,d))$ *for all* $y \in SV$ *and* $a \in AV$ *which occur free in* p *and all* $d \in D$ *then*

$$T(p)(\varepsilon,\sigma) = T(p)(\varepsilon,\sigma').$$

PROOF. By induction on the structure of p. □

Five of the proof rules, namely (R1), (R2), (R9), (R10) and (R11) are obviously sound. The rest of this section is devoted to the detailed proof of soundness of the variable declarations rule (R3). In the proof we shall need the following theorem whose proof we postpone for a moment.

THEOREM 1. *Assume that* $E \in \mathcal{E}$, $S \in \mathcal{S}$, $var(E) \subseteq F \subseteq SV$ *and let* $\theta \in \Theta$ be F-*invariant. Then for all* $x,y,y' \in SV$, $\varepsilon \in Env$, $\alpha \in A$ *and* $\sigma \in \Sigma$ *such that* $var(S) \cup F \subseteq dom(\varepsilon)$, $y,y' \notin dom(\varepsilon)$ *and* $\alpha \in A \backslash range(\varepsilon)$

$$M(E|S[y/x])(\theta)(\varepsilon \cup \langle y,\alpha\rangle,\sigma) = M(E|S[y'/x])(\theta)(\varepsilon \cup \langle y',\alpha\rangle,\sigma).$$

Using this theorem we can easily prove the soundness of the variable declarations rule. Namely assume that $\Gamma \to \langle E|\{p\}R^1[y/x]\{q\}\rangle$ is valid where y does not occur in E,p,R^1 or q. Let $\theta \in \Theta^E$ be such that $\Gamma(\theta)$ holds. We are to prove that $\langle E|\{p\}\underline{var}\ x;R^1\{q\}\rangle(\theta)$ holds. Let $\varepsilon \in Env$ be defined for all variables occurring in $E,p,\underline{var}\ x;R^1,q$ and assume that for some $\sigma,\sigma' \in \Sigma$ $\mathcal{T}(p)(\varepsilon,\sigma) = \underline{T}$ and $M(E|\underline{var}\ x;R^1)(\theta)(\varepsilon,\sigma) = \sigma'$. We are to show that $\mathcal{T}(q)(\varepsilon,\sigma') = \underline{T}$. If $y \in dom(\varepsilon)$ then for some $\alpha \in A$ and $\varepsilon_o \in Env$ $\varepsilon = \varepsilon_o \cup \langle y,\alpha\rangle$. Since θ is var(E)-invariant, by theorem 1 (taking F = var(E)) we get

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

$$= M(E|(var\ x;R^1)[y'/x])(\theta)(\varepsilon_o \cup \langle y',\alpha\rangle,\sigma)$$

where $y' \notin dom(\varepsilon)$. This means that $M(E|\underline{var}\ x;R^1)(\theta)(\varepsilon_o \cup \langle y',\alpha\rangle,\sigma) = \sigma'$. y and y' do not occur in p or q, so $\mathcal{T}(p)(\varepsilon_o \cup \langle y',\alpha\rangle,\sigma) = \mathcal{T}(p)(\varepsilon_o,\sigma) = \mathcal{T}(p)(\varepsilon,\sigma) = \underline{T}$ and similarly $\mathcal{T}(q)(\varepsilon_o \cup \langle y',\alpha\rangle,\sigma') = \mathcal{T}(q)(\varepsilon,\sigma')$. Thus without loss of generality we can assume that $y \notin dom(\varepsilon)$. We have

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

where y' is the first variable $\in SV$ not in $dom(\varepsilon)$ and α is the first address not in $range(\varepsilon)$

$$= \text{(by theorem 1)}$$

$$M(E|R^1[y/x])(\theta)(\varepsilon \cup \langle y,\alpha\rangle,\sigma).$$

y does not occur in p, so $\mathcal{T}(p)(\varepsilon \cup \langle y,\alpha\rangle,\sigma) = \mathcal{T}(p)(\varepsilon,\sigma) = \underline{T}$. By the definition of validity $\langle E|\{p\}R^1[y/x]\{q\}\rangle(\theta)$ holds, so by the above $\mathcal{T}(q)(\varepsilon \cup \langle y,\alpha\rangle,\sigma') = \underline{T}$. y does not occur in q, so $\mathcal{T}(q)(\varepsilon,\sigma') = \mathcal{T}(q)(\varepsilon \cup \langle y,\alpha\rangle,\sigma') = \underline{T}$ what was to be proved.

The proof of theorem 1 is so complicated that it took us more time than all of the rest of the paper. The appreciate the difficulties encountered let us have a look why some of the simpler approaches fail.

First observe that the proof by $\prec_\ell$-induction with respect to $c(E|S)$ does not work. Indeed, for $S \equiv E';R^3$ we cannot apply the induction hypothesis. One could remedy this trying to prove a stronger result namely that for all $E,E' \in \mathcal{E}$ and $S \in \mathcal{S}$ such that no procedure variable declared in E' occurs in E

$$M(E,E'[y/x]|S[y/x])(\theta)(\varepsilon\cup\langle y,\alpha\rangle,\sigma)$$
$$= M(E,E'[y'/x]|S[y'/x])(\theta)(\varepsilon\cup\langle y',\alpha\rangle,\sigma),$$

where $var(E',S) \cup F \subseteq dom(\varepsilon)$ and θ,F,x,y,y',α and σ are as in theorem 1.

One might be tempted to prove the above by $\prec_\ell$-induction with respect to $c(E,E'|S)$. However, for $S \equiv P$ where P is declared in E' one starts to consider on the left and right hand side different procedure calls, since in general $E'[y/x] \not\equiv E'[y'/x]$. This leads to constructs with different θ-s because $\Phi^{E'[y/x],\theta} \neq \Phi^{E'[y'/x],\theta}$.

To remedy this one is forced once again to strengthen the claim and try to prove that if θ and θ' are in some sense congruent then

$$M(E,E'[y/x]|S[y/x])(\theta)(\varepsilon\cup\langle y,\alpha\rangle,\sigma)$$
$$= M(E,E'[y'/x]|S[y'/x])(\theta')(\varepsilon\cup\langle y',\alpha\rangle,\sigma)$$

for all $E,E',S,y,y',x,\varepsilon,\alpha$ and σ as above.

The last refinement is still not strong enough in order to prove it directly by $\prec_\ell$-induction. To tackle the case $S \equiv \underline{var}\ x;R^3$ one is forced to resort to iterated substitutions instead of a single one. Now in turn the problem arises how to define the congruency of θ and θ'. The actual difficulties start here. The above tactics of gradual refinements finally leads to lemma 9 which can be directly proved by $\prec_\ell$-induction.

If there is any place in this paper where the use of substitution took vengeance on us it is there.

To formulate the above lemma we have to introduce a couple of notions

and some notation.

By $^\frown$ we denote a concatenation of two sequences. Instead of saying that $\bar{k}$ is a sequence of elements from a set A we simply write $\bar{k} \in A$. Similarly we write $\bar{k} \notin A$. $|\bar{k}|$ denotes the length of $\bar{k}$.

DEFINITION 8. Let $\bar{y},\bar{y}' \in SV$ be such that $|\bar{y}| = |\bar{y}'|$ and all variables in $\bar{y}$, as well as in $\bar{y}'$, are different.

(i) Let $\varepsilon,\varepsilon' \in Env$. ε and ε' are called *congruent over* $\langle\bar{y},\bar{y}'\rangle$ if for some $\varepsilon_0 \in Env$ and $\bar{\alpha} \in A$

$$\varepsilon = \varepsilon_0 \cup \langle\bar{y},\bar{\alpha}\rangle \text{ and } \varepsilon' = \varepsilon_0 \cup \langle\bar{y}',\bar{\alpha}\rangle.$$

(ii) Let $\theta,\theta' \in \Theta$ and let $F \subseteq SV$ be such that $\bar{y},\bar{y}' \notin F$. θ and θ' are called F-*congruent over* $\langle\bar{y},\bar{y}'\rangle$ if for all $P \in PV$, $\bar{y}_1,\bar{y}_1' \in SV\backslash F$, $\varepsilon,\varepsilon' \in Env$ and $\sigma \in \Sigma$ such that $F \subseteq dom(\varepsilon)$, $dom(\varepsilon')$ and ε and ε' are congruent over $\langle\bar{y}^\frown\bar{y}_1,\bar{y}'^\frown\bar{y}_1'\rangle$

$$\theta(P)(\varepsilon,\sigma) = \theta'(P)(\varepsilon',\sigma).$$

Observe that if θ and θ' are F-congruent over $\langle\bar{y},\bar{y}'\rangle$ then for all $\bar{y}_1,\bar{y}_1' \in SV$ such that $|\bar{y}_1| = |\bar{y}_1'|$ and all variables in $\bar{y}^\frown\bar{y}_1$, as well as in $\bar{y}'^\frown\bar{y}_1'$, are different, θ and θ' are G-congruent over $\langle\bar{y}^\frown\bar{y}_1,\bar{y}'^\frown\bar{y}_1'\rangle$ for any G such that $F \subseteq G \subseteq SV\backslash\bar{y},\bar{y}_1,\bar{y}',\bar{y}_1'$.

Let $\bar{y} = (y_1,\ldots,y_\ell)$ and $\bar{x} = (x_1,\ldots,x_\ell)$ where $\ell \geq 0$. We define $S[\bar{y}\leftarrow\bar{x}]$ as $S[y_1/x_1]\ldots[y_\ell/x_\ell]$. If $\ell = 0$ then $S[\bar{y}\leftarrow\bar{x}]$ is simply S. In a similar way we define $E[\bar{y}\leftarrow\bar{x}]$, $p[\bar{y}\leftarrow\bar{x}]$, $t[\bar{y}\leftarrow\bar{x}]$ and $v[\bar{y}\leftarrow\bar{x}]$.

Observe that in general $S[\bar{y}\leftarrow\bar{x}]$ is not simply $S[\bar{y}/\bar{x}]$, the result of simultaneous substitution of $\bar{y}$ for $\bar{x}$. The latter is for example not defined when some of the x_i-s are the same variables.

The following lemma connects the introduced notions.

LEMMA 8. *Let* $\bar{x},\bar{y},\bar{y}' \in SV$, $\varepsilon,\varepsilon' \in Env$ *and* $\sigma \in \Sigma$ *be such that* ε *and* ε' *are congruent over* $\langle\bar{y},\bar{y}'\rangle$ *and* $|\bar{x}| = |\bar{y}| = |\bar{y}'|$. *Then*

(i) $L(v[\bar{y}\leftarrow\bar{x}])(\varepsilon,\sigma) = L(v[\bar{y}'\leftarrow\bar{x}])(\varepsilon',\sigma)$

(ii) $R(t[\bar{y}\leftarrow\bar{x}])(\varepsilon,\sigma) = R(t[\bar{y}'\leftarrow\bar{x}])(\varepsilon',\sigma)$

(iii) $T(e[\bar{y}\leftarrow\bar{x}])(\varepsilon,\sigma) = T(e[\bar{y}'\leftarrow\bar{x}])(\varepsilon',\sigma)$

for all $v \in IV$, $t \in IE$ *and* $e \in BE$.

PROOF. By simultaneous induction. □

Now we prove a lemma from which theorem 1 trivially follows.

LEMMA 9. *Suppose that* $E,E' \in \mathcal{E}$ *are such that no procedure variable declared in* E' *occurs in* E. *Let* $S \in \mathcal{S}$, $\theta,\theta' \in \Theta$, $\bar{y},\bar{y}' \in \mathcal{SV}$ *and* $F \subseteq \mathcal{SV}$. *Assume that*

(1)
$$\theta \text{ and } \theta' \text{ are } F\text{-congruent over } \langle \bar{y},\bar{y}' \rangle,$$
$$\bar{y},\bar{y}' \notin \mathrm{var}(E,E',S) \cup F.$$

Then for all $\bar{x},\bar{y}_1,\bar{y}_1' \in \mathcal{SV}$ *such that* $|\bar{x}| = |\bar{y}^{\frown}\bar{y}_1| = |\bar{y}'^{\frown}\bar{y}_1'|$ *and* $\bar{y}_1,\bar{y}_1' \notin \mathrm{var}(E) \cup F \cup (\mathrm{var}(E',S)\setminus\{x \in \bar{x}: x \text{ does not occur bound in } E' \text{ or } S\})$, $\varepsilon,\varepsilon' \in \mathcal{Env}$ *such that* $\mathrm{var}(E) \cup F \cup (\mathrm{var}(E',S)\setminus\{x \in \bar{x}: x \text{ does not occur bound in } E' \text{ or } S\}) \subseteq \mathrm{dom}(\varepsilon)$, $\mathrm{dom}(\varepsilon')$ *and* ε *and* ε' *are congruent over* $\langle \bar{y}^{\frown}\bar{y}_1, \bar{y}^{\frown}\bar{y}_1' \rangle$ *and* $\sigma \in \Sigma$

(2)
$$M(E,E'[\bar{y}^{\frown}\bar{y}_1 \leftarrow \bar{x}]\,|\,S[\bar{y}^{\frown}\bar{y}_1 \leftarrow \bar{x}])(\theta)(\varepsilon,\sigma)$$
$$= M(E,E'[\bar{y}'^{\frown}\bar{y}_1' \leftarrow \bar{x}]\,|\,S[\bar{y}'^{\frown}\bar{y}_1' \leftarrow \bar{x}])(\theta')(\varepsilon',\sigma).$$

PROOF. We proceed by $\prec_\ell$-induction with respect to $c(E,E'\,|\,S)$. Let $\bar{x},\bar{y}_1,\bar{y}_1' \in \mathcal{SV}$, $\varepsilon,\varepsilon' \in \mathcal{Env}$ and σ be as defined above. We are to prove that (2) holds.

Our induction hypothesis is that the lemma holds for all $E_1,E_1' \in \mathcal{E}$ and $S_1 \in \mathcal{S}$ such that $c(E_1,E_1'\,|\,S_1) \prec_\ell c(E,E'\,|\,S)$. We have to consider various cases depending on the form of S.

Case I. S is $v:=t$.

Straightforward by lemma 8.

Case II. S is P_0.

Let $\bar{P} = (P_1,\ldots,P_n)$ and $\bar{P}' = (P_1',\ldots,P_n')$, where $E \equiv \langle P_i \Leftarrow \langle S_i \rangle\rangle_{i=1}^{n}$ and $E' \equiv \langle P_i' \Leftarrow \langle S_i' \rangle\rangle_{i=1}^{m}$. By lemma 1 we have

(3)
$$M(E,E'[\bar{y}^{\frown}\bar{y}_1 \leftarrow \bar{x}]\,|\,P_0)(\theta)(\varepsilon,\sigma) = \theta_1\{\mu\Phi^{E'[\bar{y}^{\frown}\bar{y}_1 \leftarrow \bar{x}],\theta_1/\bar{P}'}\}(P_0)(\varepsilon,\sigma),$$

where $\theta_1 = \theta\{\mu\Phi^{E,\theta}/\bar{P}\}$.

Similarly

(4) $$M(E,E'[\overline{y}'^{\frown}\overline{y}'_1\leftarrow\overline{x}]|P_0)(\theta')(\varepsilon',\sigma) = \theta'_1\{\mu\Phi^{E'[\overline{y}'^{\frown}\overline{y}'_1\leftarrow\overline{x}],\theta'_1}/\overline{P}'\}(P_0)(\varepsilon',\sigma),$$

where $\theta'_1 = \theta'\{\mu\Phi^{E,\theta'}/\overline{P}\}$.

We prove first that

(5) θ_1 and θ'_1 are var(E) $\cup$ F-congruent over $\langle\overline{y},\overline{y}'\rangle$.

Clearly it is sufficient to prove that for each $k \geq 0$

(6) $\theta\{\eta_k^{E,\theta}/\overline{P}\}$ and $\theta'\{\eta_k^{E,\theta'}/\overline{P}\}$ are var(E) $\cup$ F-congruent over $\langle\overline{y},\overline{y}'\rangle$.

We prove it by induction with respect to k. It is clearly true for $k = 0$ since we assumed (1). Assume now that (6) holds for some $k \geq 0$. Let $\overline{u},\overline{u}' \in SV$, $\varepsilon_1,\varepsilon'_1 \in Env$ be such that $\overline{u},\overline{u}' \notin$ var(E) $\cup$ F, var(E) $\cup$ F $\subseteq$ dom(ε_1), dom(ε'_1) and ε_1 and ε'_1 are congruent over $\langle\overline{y}^{\frown}\overline{u},\overline{y}'^{\frown}\overline{u}'\rangle$ and let $\sigma_1 \in \Sigma$ be arbitrarily fixed. We are to prove that for all $P \in PV$

(7) $$\theta\{\eta_{k+1}^{E,\theta}/\overline{P}\}(P)(\varepsilon_1,\sigma_1) = \theta'\{\eta_{k+1}^{E,\theta'}/\overline{P}\}(P)(\varepsilon'_1,\sigma_1).$$

In view of (1) we can assume that $n \geq 1$ and $P \equiv P_i$ for some i such that $1 \leq i \leq n$. We have

$$\theta\{\eta_{k+1}^{E,\theta}/\overline{P}\}(P_i)(\varepsilon_1,\sigma_1) = (\eta_{k+1}^{E,\theta})_i(\varepsilon_1,\sigma_1)$$

$$= M(|S_i)(\theta\{\eta_k^{E,\theta}/\overline{P}\})(\varepsilon_1,\sigma_1) = M(|S_i[\overline{y}^{\frown}\overline{u}\leftarrow\overline{x}_0])(\theta\{\eta_k^{E,\theta}/\overline{P}\})(\varepsilon_1,\sigma_1),$$

where $\overline{x}_0 \notin$ var(S_i) and $|\overline{x}_0| = |\overline{y}^{\frown}\overline{u}|$

= (by the induction hypothesis of the lemma)

$$M(|S_i[\overline{y}'^{\frown}\overline{u}'\leftarrow\overline{x}_0])(\theta'\{\eta_k^{E,\theta'}/\overline{P}\})(\varepsilon'_1,\sigma_1),$$

since $c(\ |S_i) \prec_\ell c(E,E'|S)$ and we assumed that (6) holds

$$= \theta'\{\eta_{k+1}^{E,\theta'}/\overline{P}\}(P_i)(\varepsilon_1',\sigma_1)$$

by a similar string of equalities.

In a similar way using (5) instead of (1) one can prove that

(8) $$\theta_1\{\mu\Phi^{E'[\overline{y}^\frown\overline{y}_1\leftarrow\overline{x}]},\theta_1/\overline{P}'\} \text{ and } \theta_1'\{\mu\Phi^{E'[\overline{y}'^\frown\overline{y}_1'\leftarrow\overline{x}]},\theta_1'/\overline{P}'\}$$

are $\mathrm{var}(E) \cup F \cup (\mathrm{var}(E',S)\setminus\{x \in \overline{x}: x \text{ is not bound in } E' \text{ or } S\})$-congruent over $\langle\overline{y}^\frown\overline{y}_1,\overline{y}'^\frown\overline{y}_1'\rangle$.

Now (8) in conjunction with (3) and (4) settles the proof for this case.

Case III. S is $\underline{\text{var}}\ u;R^1$.

Subcase 1^o. u does not occur in $\overline{x}$.

By assumption u does not occur in $\overline{y},\overline{y}_1,\overline{y}',\overline{y}_1'$, so by the definition of substitution $S[\overline{y}^\frown\overline{y}_1\leftarrow\overline{x}]$ is $\underline{\text{var}}\ u;R^1[\overline{y}^\frown\overline{y}_1\leftarrow\overline{x}]$ and $S[\overline{y}'^\frown\overline{y}_1'\leftarrow\overline{x}]$ is $\underline{\text{var}}\ u;R^1[\overline{y}'^\frown\overline{y}_1'\leftarrow\overline{x}]$. We have

$$M(E,E'[\overline{y}^\frown\overline{y}_1\leftarrow\overline{x}]|\underline{\text{var}}\ u;R^1[\overline{y}^\frown\overline{y}_1\leftarrow\overline{x}])(\theta)(\varepsilon,\sigma)$$

$$= M(E,E'[\overline{y}^\frown\overline{y}_1\leftarrow\overline{x}]|R^1[\overline{y}^\frown\overline{y}_1\leftarrow\overline{x}][y/u])(\theta)(\varepsilon\cup\langle y,\alpha\rangle,\sigma),$$

where y is the first variable $\in SV$ not in $\mathrm{dom}(\varepsilon)$ and α is the first address not in $\mathrm{range}(\varepsilon)$

$$= M(E,E'[\overline{y}^\frown\overline{y}_1\leftarrow\overline{x}]|R^1[u_1/u][\overline{y}^\frown\overline{y}_1\leftarrow\overline{x}][y/u_1])(\theta)(\varepsilon\cup\langle y,\alpha\rangle,\sigma),$$

where $u_1 \in SV$ and u_1 is completely fresh,
since $R^1[\overline{y}^\frown\overline{y}_1\leftarrow\overline{x}][y/u] \equiv R^1[u_1/u][\overline{y}^\frown\overline{y}_1\leftarrow\overline{x}][y/u_1]$ due to assumptions concerning $\overline{y},\overline{y}_1$, u,u_1 and ε

$$= M(E,E'[\overline{y}^\frown\overline{y}_1{}^\frown(y)\leftarrow\overline{x}^\frown(u_1)]|R^1[u_1/u][\overline{y}^\frown\overline{y}_1{}^\frown(y)\leftarrow\overline{x}^\frown(u_1)])(\theta)(\varepsilon\cup\langle y,\alpha\rangle,\sigma),$$

since u_1 does not occur in $E'[\overline{y}^{\frown}\overline{y}_1 \leftarrow \overline{x}]$

= (by the induction hypothesis)

$$M(E,E'[\overline{y}'^{\frown}\overline{y}_1'^{\frown}(y') \leftarrow \overline{x}^{\frown}(u_1)] \mid R^1[u_1/u][\overline{y}'^{\frown}\overline{y}_1'^{\frown}(y') \leftarrow \overline{x}^{\frown}(u_1)])(\theta')(\varepsilon' \cup \langle y',\alpha\rangle,\sigma),$$

where y' is the first variable $\in SV$ not in $dom(\varepsilon')$,
since $c(E,E' \mid R^1[u_1/u]) \prec_{\ell} c(E,E' \mid S)$, (1) holds and $\varepsilon \cup \langle y,\alpha\rangle$ and $\varepsilon' \cup \langle y',\alpha\rangle$ are congruent over $\langle \overline{y}^{\frown}\overline{y}_1^{\frown}(y), \overline{y}'^{\frown}\overline{y}_1'^{\frown}(y')\rangle$

$$= M(E,E' \mid [\overline{y}'^{\frown}\overline{y}_1' \leftarrow \overline{x}] \mid \underline{\text{var}}\ u; R^1[\overline{y}'^{\frown}\overline{y}_1' \leftarrow \overline{x}])(\theta')(\varepsilon',\sigma)$$

by a similar string of equalities.

Subcase 2°. u occurs in $\overline{x}$.

Let $\tilde{x}$ be the substring of $\overline{x}$ resulting from deleting from $\overline{x}$ all the occurrences of u and let $\tilde{y},\tilde{y}'$ be respectively the corresponding substrings of $\overline{y}^{\frown}\overline{y}_1$ and $\overline{y}'^{\frown}\overline{y}_1'$. By the definition of substitution $S[\overline{y}^{\frown}\overline{y}_1 \leftarrow \overline{x}]$ is $\underline{\text{var}}\ u; R^1[\tilde{y} \leftarrow \tilde{x}]$ and $S[\overline{y}'^{\frown}\overline{y}_1' \leftarrow \overline{x}]$ is $\underline{\text{var}}\ u; R^1[\tilde{y}' \leftarrow \tilde{x}]$. Observe that for any $y \in SV$ not in $dom(\varepsilon)$ $R^1[\tilde{y} \leftarrow \tilde{x}][y/u] \equiv R^1[u_1/u][\overline{y}^{\frown}\overline{y}_1 \leftarrow \overline{x}][y/u_1]$, where $u_1 \in SV$ and u_1 is completely fresh and similarly for $\tilde{y}'$ and $\overline{y}'^{\frown}\overline{y}_1'$. Thus the argument used in subcase 1° works in this subcase, as well.

Observe that if we used in the claim of the lemma the set $var(E,E',S) \cup F$ instead of $var(E) \cup F \cup (var(E',S) \setminus \{x \in \overline{x}: x \text{ is not bound in } E' \text{ or } S\})$ then the above argument would not work. Indeed, we could not use then the induction hypothesis as we would have no guarantee that $var(R^1[u_1/u]) \subseteq dom(\varepsilon \cup \langle y,\alpha\rangle)$, $dom(\varepsilon \cup \langle y',\alpha\rangle)$. The additional proviso saves the situation: $u_1 \in \{x \in \overline{x}^{\frown}(u_1): x \text{ is not bound in } E' \text{ or } R^1[u_1/u]\}$ so we do not need here to have $u_1 \in dom(\varepsilon \cup \langle y,\alpha\rangle)$, $dom(\varepsilon \cup \langle y',\alpha\rangle)$ to be able to use the induction hypothesis.

The proof of other cases uses lemma 8 and is straightforward. □

The proof of theorem 1 is now immediate. It is easy to see that if θ is F-invariant then θ and θ are F-congruent over $\langle, \rangle$. Choose now E' to be the empty system of procedure declarations, θ' to be θ, and $\overline{y}$ and $\overline{y}'$ to be the empty sequences. Then (1) holds and clearly lemma 9 implies theorem 1.

This concludes the proof of soundness of the variable declarations rule (R3).

In the sequel we shall need the following corollary to theorem 1.

COROLLARY 1. *Assume that* $E \in \mathcal{E}$, $\mathrm{var}(E) \subseteq F \subseteq SV$ *and* $\theta \in \Theta$. *Let* $\overline{P}$ *be the sequence of the procedure variables declared in* E. *If* θ *is* F*-invariant then for all* $k \geq 0$ $\theta\{\eta_k^{E,\theta}/\overline{P}\}$ *is* F*-invariant.*

PROOF. Straightforward by induction using theorem 1. □

10. PROOF OF THE SOUNDNESS THEOREM (CONTINUED)

The proof of soundness of the array declarations rule (R4) is analogous to that of the variable declarations rule reason why we omit it. The proof uses the following theorem analogous to the theorem 1.

THEOREM 2. *Assume that* $E \in \mathcal{E}$, $S \in \mathcal{S}$, $\mathrm{array}(E) \subseteq G \subseteq AV$ *and let* $\theta \in \Theta$ *be* G*-invariant. Then for all* $a,b,b' \in AV$, $\varepsilon \in Env$, $\alpha_d \in A(d \in D)$ *and* $\sigma \in \Sigma$ *such that* $(\mathrm{array}(S) \cup G) \times D \subseteq \mathrm{dom}(\varepsilon)$, $\{b,b'\} \times D \cap \mathrm{dom}(\varepsilon) = \emptyset$ *and* $\alpha_d \in A \backslash \mathrm{range}(\varepsilon)$

$$M(E|S[b/a])(\theta)(\varepsilon \cup \langle\langle b,d\rangle,\alpha_d\rangle_{d\in D},\sigma)$$

$$= M(E|S[b'/a])(\theta)(\varepsilon \cup \langle\langle b',d\rangle,\alpha_d\rangle_{d\in D},\sigma).$$

In the sequel we shall need the following corollary to theorem 2.

COROLLARY 2. *Assume that* $E \in \mathcal{E}$, $\mathrm{array}(E) \subseteq G \subseteq AV$ *and* $\theta \in \Theta$. *Let* $\overline{P}$ *be the sequence of the procedure variables declared in* E. *If* θ *is* G*-invariant then for all* $k \geq 0$ $\theta\{\eta_k^{E,\theta}/\overline{P}\}$ *is* G*-invariant.*

PROOF. Straightforward by induction using theorem 2. □

An observation that for any $E,E' \subset \mathcal{E}$ $\Theta^E \subseteq \Theta^{E,E'}$ together with lemma 2 clearly implies the soundness of the procedure declarations rule (R5).

In the proof of soundness of the consequence rule and the substitution

rule we shall need the following lemma and theorem.

LEMMA 10. *Suppose that* $t \in IE$, $p \in AST$, $var(p,t) \subseteq F \subseteq SV$ *and* $array(p,t) \subseteq G \subseteq AV$. *If* (ε,σ) *fits* (ε',σ') *over* (F,G) *then*

(a) $R(t)(\varepsilon,\sigma) = R(t)(\varepsilon',\sigma')$

(b) $T(p)(\varepsilon,\sigma) = T(p)(\varepsilon',\sigma')$.

PROOF. Straightforward by induction on the structure of t and p. □

THEOREM 3. *Let* $E \in E$, $S \in S$, $F \subseteq SV$ and $G \subseteq AV$ *be such that* $var(E,S) \subseteq F$ *and* $array(E,S) \subseteq G$. *Assume that* θ *is* F*-invariant,* G*-invariant, fits* (F,G) *and that* (ε,σ) *fits* (ε',σ') *over* (F,G). *Then* $(\varepsilon,M(E|S)(\theta)(\varepsilon,\sigma))$ *fits* $(\varepsilon',M(E|S)(\theta)(\varepsilon',\sigma'))$ *over* (F,G).

More precisely: $M(E|S)(\theta)(\varepsilon,\sigma)$ *is defined iff* $M(E|S)(\theta)(\varepsilon',\sigma')$ *is defined and if both sides are defined then* $(\varepsilon,M(E|S)(\theta)(\varepsilon,\sigma))$ *fits* $(\varepsilon',M(E|S)(\theta)(\varepsilon',\sigma'))$ *over* (F,G).

PROOF. We proceed by $\prec_\ell$-induction with respect to $c(E|S)$.

Case I. S is v:=t.

Let $\sigma_1 = \sigma\{R(t)(\varepsilon,\sigma)/L(v)(\varepsilon,\sigma)\}$ and $\sigma_1' = \sigma'\{R(t)(\varepsilon',\sigma')/L(v)(\varepsilon',\sigma')\}$. By lemma 10 $R(t)(\varepsilon,\sigma) = R(t)(\varepsilon',\sigma')$. Clearly $L(v)(\varepsilon,\sigma) \in range(\varepsilon)$ and $L(v)(\varepsilon',\sigma') \in range(\varepsilon')$.

If v is a simple variable, say z, then $L(z)(\varepsilon,\sigma) = \varepsilon(z)$ and $L(z)(\varepsilon',\sigma') = \varepsilon'(z)$. Thus in this case $\sigma_1(\varepsilon(z)) = \sigma_1'(\varepsilon'(z))$.

If v is a subscripted variable, say a[s], then $L(a[s])(\varepsilon,\sigma) = \varepsilon(a,d)$ and $L(a[s])(\varepsilon',\sigma') = \varepsilon'(a,d)$, where $d = R(s)(\varepsilon,\sigma) = R(s)(\varepsilon',\sigma')$. Thus in this case $\sigma_1(\varepsilon(a,d)) = \sigma_1'(\varepsilon'(a,d))$.

This together with the definition of σ_1 and σ_1' implies that (ε,σ_1) fits (ε',σ_1') over (F,G).

Case II. S is $\underline{var}\ x;R^1$.

We have

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

where y is the first variable $\in SV$ not in $dom(\varepsilon)$ and α is the first address in $A \setminus range(\varepsilon)$,

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

where y' is the first variable $\in SV$ not in dom(ε') and α' is the first address in A\range(ε').
Let z be a simple variable which does not occur in dom(ε) or dom(ε'). By theorem 1 and the above equalities

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

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

Clearly $(\varepsilon \cup \langle z,\alpha\rangle,\sigma)$ fits $(\varepsilon' \cup \langle z,\alpha'\rangle,\sigma')$ over $(F \cup \{z\},G)$, so by the induction hypothesis $(\varepsilon \cup \langle z,\alpha\rangle, M(E|R^1[z/x])(\theta)(\varepsilon \cup \langle z,\alpha\rangle,\sigma))$ fits $(\varepsilon' \cup \langle z,\alpha'\rangle, M(E|R^1[z/x])(\theta)(\varepsilon' \cup \langle z,\alpha'\rangle,\sigma'))$ over $(F \cup \{z\},G)$. This implies that $(\varepsilon, M(E|R^1[z/x])(\theta)(\varepsilon \cup \langle z,\alpha\rangle,\sigma))$ fits $(\varepsilon', M(E|R^1[z/x])(\theta)(\varepsilon' \cup \langle z,\alpha'\rangle,\sigma'))$ over (F,G) which settles the proof in this case.

Case III. S is $\underline{\text{array}}\ a;R^2$.

The proof uses theorem 2 and is analogous to the proof in the previous case so we omit it. The only problem arising here is that we have not specified how the addresses $\alpha_d (d \in D)$ needed for the definition of $M(E|\underline{\text{array}}\ a;R^2)(\theta)(\varepsilon,\sigma)$ are actually chosen from A\range(ε). We can assume that D is well-ordered. For $d \in D$ being the k-th element of D define α_d to be the 2k+1-address in A\range(ε). This choice is clearly satisfactory both for the proof in this case and for the requirements entering the definition of environments.

Case IV. S is P.

E is of the form $\langle P_i \Leftarrow \langle S_i \rangle \rangle_{i=1}^n$. Let $\overline{P} = (P_1,\ldots,P_n)$. The straightforward proof by induction on k shows that for all $k \geq 0$ $\theta\{\eta_k^{E,\theta}/\overline{P}\}$ fits (F,G). The proof uses the induction hypothesis of the theorem and corollaries 1 and 2. Hence by continuity of $\Phi^{E,\theta}$ $\theta\{\mu\Phi^{E,\theta}/\overline{P}\}$ fits (F,G), which clearly settles the proof in this case.

The proof of other cases is straighforward. □

In the sequel we shall need the following corollary to theorem 3.

COROLLARY 3. *Assume that* $E \in \mathcal{E}$, $var(E) \subseteq F \subseteq SV$, $array(E) \subseteq G \subseteq AV$ and $\theta \in \Theta$. *Let* $\overline{P}$ *be the sequence of the procedure variables declared in* E. *If* θ *is* F-*invariant*, G-*invariant and* θ *fits* (F,G) *then for all* $k \geq 0$ $\theta\{\eta_k^{E,\theta}/\overline{P}\}$ *fits* (F,G).

PROOF. Straightforward by induction using theorem 3 and corollaries 1 and 2. □

To prove the soundness of the consequence rule (R6) assume that

(1) $\Gamma \to \langle E|p \to p_1, \{p_1\}S\{q_1\}, q_1 \to q\rangle$ is valid.

Let $\theta \in \Theta^E$ be such that $\Gamma(\theta)$ holds. We are to prove that $\langle E|\{p\}S\{q\}\rangle(\theta)$ holds. Let $\varepsilon \in \mathcal{Env}$ be defined for all variables occurring in E,p,S,q and assume that for some $\sigma,\sigma_1 \in \Sigma$ $\mathcal{T}(p)(\varepsilon,\sigma) = \underline{T}$ and $\mathcal{M}(E|S)(\theta)(\varepsilon,\sigma) = \sigma_1$. We are to show that $\mathcal{T}(q)(\varepsilon,\sigma_1) = \underline{T}$.

Let $\varepsilon' \in \mathcal{Env}$ be an extension of ε defined for all variables occurring in p_1 and q_1. Let $\sigma' \in \Sigma$ be such that $\sigma'(\alpha) = \sigma(\alpha)$ if $\alpha \in range(\varepsilon)$ and $\sigma'(\alpha) = \sigma(\alpha')$ if α is the k-th address in $\mathcal{A}\backslash range(\varepsilon')$ and α' is the k-th address in $\mathcal{A}\backslash range(\varepsilon)$. Let $F = var(E,p,S,q)$ and $G = array(E,p,S,q)$. Clearly

(2) (ε,σ) fits (ε',σ') over (F,G).

Since $\mathcal{T}(p)(\varepsilon,\sigma) = \underline{T}$, by lemma 10 and (2) $\mathcal{T}(p)(\varepsilon',\sigma') = \underline{T}$. Hence by (1) $\mathcal{T}(p_1)(\varepsilon',\sigma') = \underline{T}$.

$\theta \in \Theta^E$ so θ is F-invariant, G-invariant and θ fits (F,G). Thus by (2) and theorem 3

(3) (ε,σ_1) fits (ε',σ_1') over (F,G),

where $\sigma_1' = \mathcal{M}(E|S)(\theta)(\varepsilon',\sigma')$. $\mathcal{T}(p_1)(\varepsilon',\sigma') = \underline{T}$ so (1) implies $\mathcal{T}(q)(\varepsilon',\sigma_1') = \underline{T}$. By lemma 10 and (3) $\mathcal{T}(q)(\varepsilon,\sigma_1) = \underline{T}$ what was to be proved.

We now prove the soundness of the substitution rule (R7). Assume that $\Gamma \to \langle E|\{p\}P\{q\}\rangle$ is valid. Suppose that $\overline{y},\overline{z},\overline{b}$ and $\overline{c}$ satisfy the conditions

1^o - 4^o of the substitution rule. Clearly by 1^o $p[\overline{y}/\overline{z}][\overline{b}/\overline{c}] \equiv p[\overline{y} \leftarrow \overline{z}][\overline{b} \leftarrow \overline{c}]$ and $q[\overline{y}/\overline{z}][\overline{b}/\overline{c}] \equiv q[\overline{y} \leftarrow \overline{z}][\overline{b} \leftarrow \overline{c}]$. Let now $z \in SV$ and $c \in AV$ be some variables which do not occur in E. To prove the soundness of the substitution rule it is sufficient to prove validity of the following four correctness phrases

(a) $\Gamma \rightarrow \langle E|\{p[y/z]\}P\{q[y/z]\}\rangle$,

where $y \in SV$ and y does not occur in E,p or q.

(b) $\Gamma \rightarrow \langle E|\{p[y/z]\}P\{q\}\rangle$,

where $y \in SV$ and z does not occur in q.

(c) $\Gamma \rightarrow \langle E|\{p[b/c]\}P\{q[b/c]\}\rangle$,

where $b \in AV$ and b does not occur in E,p or q.

(d) $\Gamma \rightarrow \langle E|\{p[b/c]\}P\{q\}\rangle$,

where $b \in AV$ and c does not occur in q.

Indeed, if we prove that the validity of $\Gamma \rightarrow \langle E|\{p\{P\{q\}\rangle$ implies the validity of (a) - (d) then by the repeated use of this implication we finally get that

$$\Gamma \rightarrow \langle E|\{p[\overline{y} \leftarrow \overline{z}][\overline{b} \leftarrow \overline{c}]\}P\{q[\overline{y} \leftarrow \overline{z}][\overline{b} \leftarrow \overline{c}]\}\rangle$$

is valid. In view of the above syntactic equalities this will show that the substitution rule is sound. So let us prove the validity of the above four correctness phrases. Let $\theta \in \Theta^E$ be such that $\Gamma(\theta)$ holds.

ad (a).

Suppose that $y \in SV$ and y does not occur in E,p or q. We prove that

$$\langle E|\{p[y/z] \wedge y = y \wedge z = z\}P\{q[y/z]\}\rangle(\theta)$$

holds which together with the soundness of the consequence rule will imply that (a) is valid. So assume that for some $\varsigma \in Env$ defined for all variables occurring in $E,p[y/z] \wedge y = y \wedge z = z$ and $q[y/z]$ and some $\sigma,\sigma' \in \Sigma$ we have $\mathcal{T}(p[y/z] \wedge y = y \wedge z = z)(\varepsilon,\sigma) = \underline{T}$ and $\mathcal{M}(E|P)(\theta)(\varepsilon,\sigma) = \sigma'$. We are to prove that $\mathcal{T}(q[y/z])(\varepsilon,\sigma') = \underline{T}$.

There is $\varepsilon_1 \in Env$ and $\alpha,\beta \in A$ such that $\varepsilon = \varepsilon_1 \cup \langle y,\alpha\rangle \cup \langle z,\beta\rangle$. Let x and u be

completely new simple variables. Let $p' \equiv p[u/z]$. We have

$$
\begin{aligned}
& T(p)(\varepsilon_1 \cup \langle z,\alpha\rangle \cup \langle x,\beta\rangle,\sigma) \\
&= T(p'[z/u])(\varepsilon_1 \cup \langle z,\alpha\rangle \cup \langle x,\beta\rangle,\sigma), \text{ since } p'[z/u] \equiv p \\
&= T(p'[y/u])(\varepsilon_1 \cup \langle y,\alpha\rangle \cup \langle x,\beta\rangle,\sigma), \text{ since z and y do not occur in } p' \\
&= T(p'[y/u])(\varepsilon_1 \cup \langle y,\alpha\rangle \cup \langle z,\beta\rangle,\sigma), \text{ since x and z do not occur in } p'[y/u] \\
&= T(p[y/z])(\varepsilon,\sigma) = \underline{T}, \text{ since } p[y/z] \equiv p'[y/u].
\end{aligned}
$$

Also, due to theorem 1

$$
\begin{aligned}
& M(E|P)(\theta)(\varepsilon_1 \cup \langle z,\alpha\rangle \cup \langle x,\beta\rangle,\sigma) \\
&= M(E|P)(\theta)(\varepsilon_1 \cup \langle y,\alpha\rangle \cup \langle x,\beta\rangle,\sigma) \\
&= M(E|P)(\theta)(\varepsilon_1 \cup \langle y,\alpha\rangle \cup \langle z,\beta\rangle,\sigma) \\
&= M(E|P)(\theta)(\varepsilon,\sigma) = \sigma', \\
& \quad \text{since } P \equiv P[z/u] \equiv P[y/u] \equiv P[x/u] \equiv P[z/u].
\end{aligned}
$$

$\Gamma \rightarrow \langle E|\{p\}P\{q\}\rangle$ is valid so $\langle E|\{p\}P\{q\}\rangle(\theta)$ holds. By this and the above we get that $T(q)(\varepsilon_1 \cup \langle z,\alpha\rangle \cup \langle x,\beta\rangle,\sigma') = \underline{T}$. By the identical string of equalities as that concerning p we get that $T(q[y/z])(\varepsilon,\sigma') = \underline{T}$ what was to be proved.

ad (b).

Suppose that z does not occur in q. We prove that

$$\langle E|\{p[y/z] \wedge y = y \wedge z = z\}P\{q\}\rangle(\theta)$$

holds which together with the soundness of the consequence rule will imply that (b) is valid. So assume that for some $\varepsilon \in Env$ defined for all variables occurring in $E,p[y/z] \wedge y = y \wedge z = z$ and q and some $\sigma,\sigma' \in \Sigma$ we have $T(p[y/z] \wedge y = y \wedge z = z)(\varepsilon,\sigma) = \underline{T}$ and $M(E|P)(\theta)(\varepsilon,\sigma) = \sigma'$. We are to prove that $T(q)(\varepsilon,\sigma') = \underline{T}$. Let $\sigma_1 = \sigma\{\sigma(\varepsilon(y))/\varepsilon(z))\}$.

Clearly $T(p[y/z])(\varepsilon,\sigma) = T(p)(\varepsilon,\sigma_1)$, so $T(p)(\varepsilon,\sigma_1) = \underline{T}$. Let $F = var(E,q)$

and G = array(E,q). Since $z \notin F$, (ε,σ) fits (ε,σ_1) over (F,G). By theorem 3 (ε,σ') fits (ε,σ_1') over (F,G), where $\sigma_1' = M(E|P)(\theta)(\varepsilon,\sigma_1)$. $<E|\{p\}P\{q\}>(\theta)$ holds, so $T(q)(\varepsilon,\sigma_1') = \underline{T}$. Now by lemma 10 and the above $T(q)(\varepsilon,\sigma') = \underline{T}$ what was to be proved.

ad (c), (d).

The proofs use theorems 2 and 3 and are analogous to the above ones so we omit them.

This concludes the proof of soundness of the substitution rule (R7).

11. PROOF OF THE SOUNDNESS THEOREM - THE CASE OF THE PROCEDURE CALLS RULE

We now turn to the proof of soundness of the procedure calls rule (R8). In the proof we shall use the following theorem which we shall prove later.

THEOREM 4. *If* $E \in \bar{E}$, $var(E) \subseteq F \subseteq SV$, $array(E) \subseteq G \subseteq AV$ and $\theta \in \Theta^{F,G}$ *then for all* $k \geq 0$ $\theta\{\eta_k^{E,\theta}/\bar{P}\} \in \Theta^{F,G}$, *where* $\bar{P}$ *is the sequence of the procedure variables declared in* E.

Let now E and $E' \equiv <P_i \Leftarrow <S_i>>_{i=1}^n$ be given systems of procedure declarations such that for $i = 1,\ldots,n$ P_i does not occur in E. Let Γ be a correctness formula and let S_0 be a statement. Assume that $P_1',\ldots,P_n'$ are some procedure variables which do not occur in Γ,E,E' or S_0. Let $\bar{P} = (P_1,\ldots,P_n)$ and $\bar{P}' = (P_1',\ldots,P_n')$. Assume that

$$(1)\quad \Gamma,\{p_1\}P_1'\{q_1\},\ldots,\{p_n\}P_n'\{q_n\} \rightarrow <E,E'|\{p_0\}S_0'\{q_0\},\{p_1\}S_1'\{q_1\},\ldots,\{p_n\}S_n'\{q_n\}>$$

is valid, where for $i = 0,\ldots,n$ $S_i' \equiv S_i[\bar{P}'/\bar{P}]$. We are to show that $\Gamma \rightarrow <E,E'|\Gamma_0>$, where $\Gamma_0 = \{p_0\}S_0\{q_0\},\{p_1\}P_1\{q_1\},\ldots,\{p_n\}P_n\{q_n\}$, is valid. So assume that $\theta \in \Theta^{E,E'}$ and suppose that $\Gamma(\theta)$ holds. We are to prove that $<E,E'|\Gamma_0>(\theta)$ holds. Due to assumptions about E and E' and lemma 1 it amounts to showing that $<E'|\Gamma_0>(\theta')$ holds, where $\theta' = \theta\{\mu\Phi^{E,\theta}/\bar{Q}\}$ and $\bar{Q}$ is the sequence of procedure variables declared in E. By lemma 2 it is equivalent to show that $<E_1|\Gamma_1>(\theta')$ holds, where $E_1 \equiv <P_i' \Leftarrow <S_i'>>_{i=1}^n$ and $\Gamma_1 = \{p_0\}S_0'\{q_0\},\{p_1\}P_1'\{q_1\},\ldots,\{p_n\}P_n'\{q_n\}$. By lemma 1 it is enough to prove that $<|\Gamma_1>(\theta'\{\mu\Phi^{E_1,\theta'}/\bar{P}'\}$ holds or, due to lemma 6, that for all $k \geq 0$

(2) $\qquad \Gamma_1(\theta'\{\eta_k^{E_1,\theta'}/\overline{P}'\})$

holds. To prove (2) we shall need the following lemma.

LEMMA 11. *For every* $E \in \mathcal{E}$, $S \in \mathcal{S}$, $\theta \in \Theta$

$$M(E|S)(\theta) = M(E|S)(\theta\{\overline{\eta}/\overline{P}\})$$

for all $\overline{P} = (P_1,\ldots,P_n)$ *such that* $P_1,\ldots,P_n$ *do not occur in* E *or* S *and all* $\overline{\eta} \in \mathcal{H}^n$.

PROOF. Straightforward by $\prec_\ell$-induction with respect to $c(E|S)$. In the case when S is of the form $E';R^3$ one uses lemma 2 to be able to apply the induction hypothesis. □

COROLLARY 4. *For every* $E \in \mathcal{E}$, $\theta,\theta_1 \in \Theta$ *if* $\theta_1 = \theta\{\overline{\eta}/\overline{P}\}$, *where* $\overline{P} = (P_1,\ldots,P_n)$, $\overline{\eta} \in \mathcal{H}^n$ *and for* $i = 1,\ldots,n$ P_i *does not occur in* E, *then*

$$\mu\Phi^{E,\theta} = \mu\Phi^{E,\theta_1}.$$

PROOF. By lemma 11. □

Let for $k \geq 0$ $\theta_k = \theta\{\eta_k^{E_1,\theta'}/\overline{P}'\}$. Observe now that for all $k \geq 0$

$$\theta'\{\eta_k^{E_1,\theta'}/\overline{P}'\} = \theta\{\mu\Phi^{E,\theta}/\overline{Q}\}\{\eta_k^{E_1,\theta'}/\overline{P}'\} = \text{(by corollary 4)}$$

$$\theta\{\mu\Phi^{E,\theta_k}/\overline{Q}\}\{\eta_k^{E_1,\theta'}/\overline{P}'\} = \theta_k\{\mu\Phi^{E,\theta_k}/\overline{Q}\}.$$

$\theta \in \Theta^{E,E'}$, so by continuity of $\mu\Phi^{E,\theta}$ and theorem 4 we have $\theta' \in \Theta^{E,E'} = \Theta^{E,E_1}$. Once again by theorem 4 for any $k \geq 0$ $\theta'\{\eta_k^{E_1,\theta'}/\overline{P}'\} \in \Theta^{E,E_1}$, i.e. by the above $\theta_k\{\mu\Phi^{E,\theta_k}/\overline{Q}\} \in \Theta^{E,E_1}$. Since $\theta \in \Theta^{E,E_1}$, we get that $\theta_k \in \Theta^{E,E_1}$, i.e. for every $k \geq 0$

(3) $\qquad \theta_k \in \Theta^{E,E'}$.

Let now S be a statement and let S' denote $S[\overline{P}'/\overline{P}]$. We have

$$M(\,|S')(\theta'\{\eta_k^{E_1,\theta'}/\overline{P}'\})$$

$$= \text{(by the above) } M(\,|S')(\theta_k\{\mu\Phi^{E,\theta_k}/\overline{Q}\}$$

$$= \text{(by lemma 1) } M(E|S')(\theta_k)$$

$$= \text{(by lemma 4) } M(E,E'|S')(\theta_k).$$

So we have proved that for any statement S and $k \geq 0$

$$(4) \qquad M(\,|S[\overline{P}'/\overline{P}])(\theta'\{\eta_k^{E_1,\theta'}/\overline{P}'\}) = M(E,E'|S[\overline{P}'/\overline{P}])(\theta_k).$$

We prove now that for every $k \geq 0$

$$(5_k) \qquad < |\{p_1\}P_1'\{q_1\},\ldots,\{p_n\}P_n'\{q_n\}>(\theta_k)$$

implies

$$(6_k) \qquad < |\{p_0\}S_0'\{q_0\},\{p_1\}S_1'\{q_1\},\ldots,\{p_n\}S_n'\{q_n\}>(\theta'\{\eta_k^{E_1,\theta'}/\overline{P}'\}).$$

Suppose that for some $k \geq 0$ (5_k) holds. We assumed that $\Gamma(\theta)$ holds. Since $P_1',\ldots,P_n'$ do not occur in Γ, by lemma 11 $\Gamma(\theta_k)$ holds, so $(\Gamma,\{p_1\}P_1'\{q_1\},\ldots,\{p_n\}P_n'\{q_n\})(\theta_k)$ holds. We assumed that (1) is valid, so by (3) we get that for $i = 0,\ldots,n$ $<E,E'|\{p_i\}S_i'\{q_i\}>(\theta_k)$ holds. This in view of (4) implies that (6_k) holds.

Observe now that for $i = 1,\ldots,n$

$$M(\,|S_i')(\theta'\{\eta_k^{E_1,\theta'}/\overline{P}'\}) = \Phi_i^{E_1}(\eta_k^{E_1,\theta'}) = (\eta_{k+1}^{E_1,\theta'})_i =$$

$$= M(\,|P_i')(\theta'\{\eta_{k+1}^{E_1,\theta'}/\overline{P}'\})$$

$$= \text{(by (4)) } M(E,E'|P_i')(\theta_{k+1})$$

$$= \text{(by lemma 3) } M(\,|P_i')(\theta_{k+1}).$$

This shows that for every $k \geq 0$ (6_k) implies (5_{k+1}). Since (5_k) implies (6_k) we see that for every $k \geq 0$ (5_k) implies (5_{k+1}). Obviously (5_0) holds, so by induction for all $k \geq 0$ (5_k) holds. Thus for all $k \geq 0$ (6_k) holds.
In particular for all $k \geq 0$

(7) $\quad < |\{p_0\}S_0\{q_0\}>(\theta'\{\eta_k^{E_1,\theta'}/\overline{P}'\})$

holds.

Observe that for i = 1,...,n and k ≥ 0

$$M(|P_i')(\theta'\{\eta_k^{E_1,\theta'}/\overline{P}'\}$$
$$= \text{(by (4))}\ M(E,E'|P_i')(\theta_k)$$
$$= \text{(by lemma 3)}\ M(|P_i')(\theta_k).$$

Since for all k ≥ 0 (5_k) holds, this shows that for all k ≥ 0

(8) $\quad < |\{p_1\}P_1'\{q_1\},\ldots,\{p_n\}P_n'\{q_n\}>(\theta'\{\eta_k^{E_1,\theta'}/\overline{P}'\})$

holds. (7) and (8) imply (2) so we have proved that the procedure calls rule (R8) is sound.

To prove theorem 4 we have to check all 5 properties entering the definition of $\Theta^{F,G}$. Assume that $E \in \mathcal{E}$, $var(E) \subseteq F \subseteq SV$, $array(E) \subseteq G \subseteq AV$ and $\theta \in \Theta^{F,G}$. Let $\overline{P}$ be the sequence of the procedure variables declared in E.

LEMMA 12. *If* θ *is* F*-bound then for all* $k \geq 0$ $\theta\{\eta_k^{E,\theta}/\overline{P}\}$ *is* F*-bound.*

PROOF. Straightforward by induction using the following lemma.

LEMMA 13. *For all* $F' \subseteq SV$, $E' \in \mathcal{E}$, $S \in \mathcal{S}$ *such that* $var(E',S) \subseteq F'$, $\theta \in \Theta$ *which is* F*-bound,* $\varepsilon \in Env$ *such that* $F' \subseteq dom(\varepsilon)$ *and* $\sigma,\sigma' \in \Sigma$ *if*

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

then for all $y \in dom(\varepsilon)\backslash F'$

$$\sigma(\varepsilon(y)) = \sigma'(\varepsilon(y)).$$

PROOF. The proof proceeds by $\prec_\ell$-induction with respect to $c(E'|S)$. All cases are straightforward with the exception of these of variable and array declarations. These two cases are easily handled thanks to the observation that if

$F' \subseteq F'' \subseteq SV$ *and* θ *is* F'*-bound then* θ *is* F''*-bound.* □□

LEMMA 14. *If* θ *is* G*-bound then for all* $k \geq 0$ $\theta\{\eta_k^{E,\theta}/\overline{P}\}$ *is* G*-bound.*

PROOF. Straightforward by induction using the following lemma.

LEMMA 15. *For all* $G' \subseteq AV$, $E' \in E$, $S \in S$ *such that* $\mathrm{array}(E',S) \subseteq G'$, $\theta \in \Theta$ *which is* G'*-bound,* $\varepsilon \in Env$ *such that* $G' \times D \subseteq \mathrm{dom}(\varepsilon)$ *and* $\sigma,\sigma' \in \Sigma$ *if*

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

then for all $a \in AV$ *and* $d \in D$ *such that* $(a,d) \in \mathrm{dom}(\varepsilon) \setminus (G' \times D)$

$$\sigma(\varepsilon(a,d)) = \sigma'(\varepsilon(a,d)).$$

PROOF. Similar to the proof of lemma 13. □□

Theorem 4 is now a direct consequence of corrollaries 1,2,3 and lemmata 12 and 14.

12. THE PROBLEM OF COMPLETENESS

Having proved the soundness of H we now turn to the question of completeness of H. COOK [5] was the first to define a completeness of a Hoare-like system in a way which, among others, avoids the well-known consequences of Gödel's Incompleteness Theorem. Before presenting his definition let us introduce some notions.

DEFINITION 9. A pair $E|S$ where $E \in E$ and $S \in S$ is called *normal* if the correctness phrase $\langle E|\{\underline{\text{true}}\}S\{\underline{\text{true}}\}\rangle$ is normal (see def. 1).

Observe that if a pair $E|S$ is normal then for any $\theta \in \Theta$, $\varepsilon \in Env$ and $\sigma \in \Sigma$ the state $M(E|S)(\theta)(\varepsilon,\sigma)$ does not depend on θ, so we shall simply write $M(E|S)(\varepsilon,\sigma)$ to denote it.

DEFINITION 10. Let p be an assertion and let $E|S$ be a normal pair. We say that an assertion q *expresses the strongest post condition corresponding to* p *and* $E|S$ if $\mathrm{var}(p,E,S) \subseteq \mathrm{var}(q)$ $\mathrm{array}(p,E,S) \subseteq \mathrm{array}(q)$ and for all

$\varepsilon \in Env$ defined for all variables occurring in q and all $\sigma \in \Sigma$

$$T(q)(\varepsilon,\sigma) = \underline{T} \text{ iff for some } \sigma' \in \Sigma\ M(E|S)(\varepsilon,\sigma') = \sigma \text{ and } T(p)(\varepsilon,\sigma') = \underline{T}.$$

We now introduce Cook's notion of expressibility slightly adopted to our particular situation.

DEFINITION 11. (COOK [5]). The assertion language is *expressive* relative to the interpretation I if for every assertion p and a normal pair E|S there exists an assertion q which expresses the strongest post condition corresponding to p and E|S.

Let now Tr denote the set of all true assertions.

DEFINITION 12. The proof system H is *complete in the sense of Cook* if, under the assumption that the assertion language is expressive relative to I, for every normal valid correctness phrase A, $\vdash_{H,Tr} A$.

In the next section we prove that the system H is complete in the sense of Cook. Before presenting the proof let us discuss the introduced notions.

We observed already before that the system H is too weak to prove all valid normal correctness phrases because it lacks any means to prove necessary facts about assertions. Supplementing H by an axiomatic system concerning assertions is of no help because Gödel's Incompleteness Theorem trivially implies that no axiomatic extension of H can be complete. The best one might hope for would be to prove that H is complete *provided* one can use an oracle that can supply answers to all questions of the form "is an assertion p true?". However even this cannot be proved. WAND [14] exhibited a particular interpretation for which the usual Hoare system for while programs is not in this sense complete. The incompleteness comes from the fact that the first order language with this interpretation is not powerful enough to express the necessary intermediate assertions. COOK [5] took care of this problem by restricting the question of completeness to such interpretations I that the assertion language is expressive relative to I.

A natural question now arises how restrictive is the assumption of expressiveness. In the case of our programming language if we take for the primitive symbols $\{f_1,\ldots,f_{\ell_o},re_1,\ldots,re_{m_o}\}$ of the language the non-logical

symbols of Peano arithmetic and take their standard interpretation I_N in the domain of the natural numbers then the assertion language is expressive relative to I_N. This can be proved in the same way that COOK [5] proved the expressiveness of the corresponding assertion language relative to I_N in the case of a corresponding fragment of Algol 60. The fact that Cook used an operational semantics is not a real issue since we proved in APT [1] the equivalence of both semantics for all normal pairs E|S.

Also, as observed in CLARKE [4], if the domain of I is finite then the expressiveness is guaranteed.

This might suggest that expressiveness is not a serious restriction. However, as a theorem of DeMillo, Lipton and Snyder shows, actually the converse is true. An interested reader can find more informations in LIPTON [11].

COOK [5] proved that a Hoare system for a subset of Algol 60 including while statement and non-recursive procedures is in the above sense complete. The paper by GORELICK [7] attempts to extend Cook's result to a class of programs allowing recursive procedures.

Neither COOK [5] nor GORELICK [7] keep track in their proof systems of the procedure declarations accessible to programs. In the case of Cook's system, as Cook himself observes, the problem is not created by the assumption that a given procedure name cannot refer to more than one declaration in a given program. However in the case of Gorelick's system, despite of Gorelick's claim, the omission of the context of procedure declarations does result in a confusion. The proof he gives is actually the proof of completeness for a language which does not allow local procedure declarations. As already mentioned before both papers incorrectly deal with scope problems. CLARKE [4] claims without proof that Gorelick's result can be modified to a language with a proper scope treatment by simple changes in semantics and the rule of variable declarations. The system Clarke proposes also differs from the Gorelick's one in that it does keep track of the procedure declarations accessible to programs.

It should be noted that in the above three papers procedures with parameters are allowed (subject to some restrictions) and much work is devoted there to the study of the parameter mechanisms. The same concerns APT & DE BAKKER [2] where procedures can call their parameters by value or

by variable. In the last paper no claims about completeness are made.

13. COMPLETENESS THEOREM

In this section we prove the following theorem.

COMPLETENESS THEOREM. *The proof system* H *is complete in the sense of Cook.*

To prove it we generalize Gorelick's completeness proof to the case of a language admitting local procedure declarations. To achieve this we have to consider nested systems of procedure declarations. The key notion relevant to the proof is that of a stable sequence (see definition 13). The main step in the proof is lemma 19 which is a generalization of lemma 6 from GORELICK [7].

Throughout this section we assume that the assertion language is expressive relative to I. If $E|S$ is a normal pair and p is an assertion then by $sp(p,E|S)$ we denote a particular assertion which expresses the strongest post condition corresponding to p and $E|S$ (say, the one with the smallest Gödel number). Instead of "$\vdash_{H,Tr} A$" we simply write "$\vdash A$". In the proof of lemma 19 we do not mention the use of the collection rule and the selection rule. We shall need below the following two lemmata.

LEMMA 16. *Suppose that* $\langle E|\{p\}\underline{\text{var}}\ x;R^1\{q\}\rangle$ *is a valid correctness phrase and* y *is a simple variable which does not occur in* $E,p,\underline{\text{var}}\ x;R^1$ *or* q. *Then* $\langle E|\{p\}R^1[y/x]\{q\}\rangle$ *is valid.*

PROOF. We prove at first that

(1) $\quad \langle E|\{p \wedge x = x \wedge y = y\}R^1[y/x]\{q\}\rangle$ is valid.

Suppose that $\theta \in \Theta^E$ and assume that for some $\varepsilon \in Env$ defined for all variables occurring in $E,p \wedge x = x \wedge y = y$, $R^1[y/x]$ and q and $\sigma,\sigma' \in \Sigma$ we have $T(p \wedge x = x \wedge y = y)(\varepsilon,\sigma) = \underline{T}$ and $M(E|R^1[y/x])(\theta)(\varepsilon,\sigma) = \sigma_1$. We are to show that $T(q)(\varepsilon,\sigma_1) = \underline{T}$.

There is $\varepsilon_1 \in Env$ and $\alpha_1 \in A$ such that $\varepsilon = \varepsilon_1 \cup \langle y,\alpha_1\rangle$. By theorem 1 $\sigma_1 = M(E|R^1[y'/x])(\theta)(\varepsilon_1 \cup \langle y',\alpha_1\rangle,\sigma)$, where y' is the first variable $\in SV$ not in $dom(\varepsilon_1)$. Let β be the first address not in $range(\varepsilon_1)$. Define $\sigma' \in \Sigma$

as follows:

$$\sigma'(\alpha) = \begin{cases} \sigma(\alpha) & \text{if } \alpha \in \text{range}(\varepsilon_1) \\ \sigma(\alpha_1) & \text{if } \alpha = \beta \\ \sigma(\alpha_0) & \text{if } \alpha \text{ is the k-th address in } A\backslash\text{range}(\varepsilon_1 \cup \langle y',\beta\rangle) \\ & \text{and } \alpha_0 \text{ is the k-th address in } A\backslash\text{range}(\varepsilon_1 \cup \langle y',\alpha_1\rangle) \end{cases}$$

Clearly

(2) $(\varepsilon_1 \cup \langle y',\alpha_1\rangle,\sigma)$ fits $(\varepsilon_1 \cup \langle y',\beta\rangle,\sigma')$ over (F,G),
where $F = \text{var}(E,R^1[y'/x],p,q)$ and $G = \text{array}(E,R^1[y'/x],p,q)$.

By theorem 3 and (2)

(3) $(\varepsilon_1 \cup \langle y',\alpha_1\rangle,\sigma_1)$ fits $(\varepsilon_1 \cup \langle y',\beta\rangle,\sigma_1')$ over (F,G),
where $\sigma_1' = M(E|R^1[y'/x])(\theta)(\varepsilon_1 \cup \langle y',\beta\rangle,\sigma')$.

By the definition of M we have $M(E|\underline{\text{var}}\ x;R^1)(\theta)(\varepsilon_1,\sigma') = \sigma_1'$. Also $T(p)(\varepsilon_1,\sigma') = T(p)(\varepsilon_1 \cup \langle y',\beta\rangle,\sigma') = T(p)(\varepsilon_1 \cup \langle y',\alpha_1\rangle,\sigma) = T(p)(\varepsilon,\sigma) = \underline{T}$, where the second equality is implied by lemma 10 and (2).

Since $\langle E|\{p\}\underline{\text{var}}\ x;R^1\{q\}\rangle$ is valid, $T(q)(\varepsilon_1,\sigma_1') = \underline{T}$ holds. We have now $T(q)(\varepsilon,\sigma_1) = T(q)(\varepsilon_1 \cup \langle y',\alpha_1\rangle,\sigma_1) = T(q)(\varepsilon_1 \cup \langle y',\beta\rangle,\sigma_1') = T(q)(\varepsilon_1,\sigma_1') = \underline{T}$, where the second equality is implied by lemma 10 and (3).

This shows that (1) holds. Now the lemma follows from (1) due to the soundness of the consequence rule. □

In a similar way using theorem 2 one can prove the following lemma.

LEMMA 17. *Suppose that* $\langle E|\{p\}\underline{\text{array}}\ a;R^2\{q\}\rangle$ *is a valid correctness phrase and* b *is an array variable which does not occur in* $E,p,\underline{\text{array}}\ a;R^2$ *or* q. *Then* $\langle E|\{p\}R^2[b/a]\{q\}\rangle$ *is valid.* □

We shall also need the following lemma proved in APT [1].

LEMMA 18. *Suppose that* $E \equiv \langle P_i \Leftarrow \langle S_i \rangle\rangle_{i=1}^n \in E$. *Then for all* $\theta \in \Theta$ *and* $i \in \{1,\ldots,n\}$

$$M(E|P_i)(\theta) = M(E|S_i)(\theta). \quad \square$$

Assume now that $\ell \geq 0$. Let for $i = 1,\ldots,\ell+1$ $E_i \equiv \langle P^i_j \Leftarrow \langle S^i_j \rangle\rangle_{j=1}^{n_i}$ be a system of procedure declarations and let $S \in \mathcal{S}$. Denote $(P^i_1,\ldots,P^i_{n_i})$ for $i = 1,\ldots,\ell+1$ by $\overline{P}_i$.

DEFINITION 13. The sequence $E_1 \cdot \ldots \cdot E_\ell \cdot E_{\ell+1} \cdot S$ (where dots signify separators and are used instead of commas in order to avoid ambiquities) is called *stable* if

(i) for all $i = 2,\ldots,\ell+1$ and $j = 1,\ldots,n_i$ P^i_j does not occur in $E_1,\ldots,E_{i-1}$.

(ii) the pair $E_1,\ldots,E_\ell,E_{\ell+1}|S$ is normal.

The above notion is closely related to the notion of a *nested sequence* introduced in APT [1].

We assumed that the sets $\mathcal{AV}$ and $\mathcal{SV}$ are well-ordered. If $F \subseteq \mathcal{SV}$ and F is finite then by $\overline{F}$ we denote the sequence of all elements of F ordered according to the well-ordering of $\mathcal{SV}$. In an analogous way we define $\overline{G}$ for $G \subseteq \mathcal{AV}$. If $F,G \subseteq \mathcal{SV}$ and $\overline{F} = (x_1,\ldots,x_k)$ and $\overline{G} = (z_1,\ldots,z_m)$, where $k \leq m$, then by $\overline{F} = \overline{G}$ we denote the formula $x_1 = z_1 \wedge \ldots \wedge x_k = z_k$. If $F,G \subseteq \mathcal{AV}$ and $\overline{F} = (a_1,\ldots,a_k)$, $\overline{G} = (c_1,\ldots,c_m)$, where $k \leq m$, then by $\overline{F} = \overline{G}$ we denote the formula

$$\forall x\,(a_1[x] = c_1[x] \wedge \ldots \wedge a_k[x] = c_k[x]).$$

Let $E_1 \cdot \ldots \cdot E_\ell \cdot E_{\ell+1} \cdot S$ be a stable sequence and let $\{Q^i_j\}_{1\leq i\leq \ell+1, 1\leq j\leq n_i}$ be different procedure variables. Denote $(Q^i_1,\ldots,Q^i_{n_i})$ for $i = 1,\ldots,\ell+1$ by $\overline{Q}_i$.

For $i = 0,\ldots,\ell+1$ and $S_0 \in S$ define $S_0[\overline{\tilde{Q}}_i \leftarrow \overline{\tilde{P}}_i]$ as $S_0[\overline{Q}_1/\overline{P}_1]\ldots[\overline{Q}_i/\overline{P}_i]$. If $i = 0$ then $S_0[\overline{\tilde{Q}}_i \leftarrow \overline{\tilde{P}}_i]$ is simply S_0. Similarly we define $E[\overline{\tilde{Q}}_i \leftarrow \overline{\tilde{P}}_i]$ for $E \in \mathcal{E}$.

Suppose now that

$$(*)\qquad \begin{array}{l} F \subseteq \mathcal{SV} \setminus \mathrm{var}(E_1,\ldots,E_{\ell+1},S),\ G \subseteq \mathcal{AV} \setminus \mathrm{array}(E_1,\ldots,E_{\ell+1},S), \\ \mathrm{card}(F) = \mathrm{card}\,(\mathrm{var}(E_1,\ldots,E_{\ell+1})),\ \mathrm{card}(G) = \mathrm{card}(\mathrm{array}(E_1,\ldots,E_{\ell+1})), \end{array}$$

where card(A) is the cardinality of the set A.

For $i = 0,\ldots,\ell + 1$ define a correctness formula $\Gamma(E_1,\ldots,E_i,\tilde{Q}_i,F,G)$ as follows:

If $i = 0$ then $\Gamma(E_1,\ldots,E_i,\tilde{Q}_i,F,G)$ is the empty set.

If $i > 0$ then

$$\Gamma(E_1,\ldots,E_i,\tilde{Q}_i,F,G) =$$

$$\Gamma(E_1,\ldots,E_{i-1},\tilde{Q}_{i-1},F,G),\ \{p_1^i\}Q_1^i\{q_1^i\},\ldots,\{p_{n_i}^i\}Q_{n_i}^i\{q_{n_i}^i\},$$

where for $j = 1,\ldots,n_i$

$$p_j^i \equiv \overline{\mathrm{var}(E_1,\ldots,E_i)} = \overline{F} \wedge \overline{\mathrm{array}(E_1,\ldots,E_i)} = \overline{G},$$

$$q_j^i \equiv \mathrm{sp}(p_j^i,E_1,\ldots,E_i|S_j^i).$$

The above correctness formulas correspond to the *most general formulas* introduced in GORELICK [7]. They can be viewed as a complete description of the procedures declared in $E_1,\ldots,E_\ell$.

Now we prove a lemma which trivially implies the completeness theorem.

LEMMA 19. *Assume that the assertion language is expressive. Suppose that for some stable sequence* $E_1\cdot\ldots\cdot E_\ell\cdot E_{\ell+1}\cdot S$ $(\ell\geq 0)$ *and assertions* p *and* q

(4) $\quad <E_1,\ldots,E_{\ell+1}|\{p\}S\{q\}>$ *is valid.*

Then

(5) $\quad \vdash \Gamma(E_1,\ldots,E_\ell,\tilde{Q}_\ell,F,G) \rightarrow <E_1,E_2[\tilde{Q}_1 \leftarrow \tilde{P}_1],\ldots,E_{\ell+1}[\tilde{Q}_\ell \leftarrow \tilde{P}_\ell]|\{p\}S[\tilde{Q}_\ell \leftarrow \tilde{P}_\ell]\{q\}>$

where $\{Q_j^i\}_{1\leq i\leq \ell+1,\,1\leq j\leq n_i}$ *are different procedure variables which do not occur in* $E_1,\ldots,E_{\ell+1}$ *or* S *and* F *and* G *satisfy* (*).

PROOF. We proceed by $\prec_\ell$-induction with respect to $c(E_{\ell+1}|S)$. So assume that the lemma holds for all stable sequences $E_1'\cdot\ldots\cdot E_k'\cdot E_{k+1}'\cdot S'$ $(k\geq 0)$ such that $c(E_{k+1}'|S') \prec_\ell c(E_{\ell+1}|S)$. Denote $\Gamma(E_1,\ldots,E_\ell,\tilde{Q}_\ell,F,G)$ by Γ_0 and

$E_1, E_2[\tilde{Q}_1 \leftarrow \tilde{P}_1], \ldots, E_\ell[\tilde{Q}_{\ell-1} \leftarrow \tilde{P}_{\ell-1}]$ by E.

Case I. S is v:=t.

By the results of DE BAKKER [3] (4) implies that the assertion $p \to q[t/v]$ is true. Thus by the axiom (A4), assignment axiom and the consequence rule (5) holds.

Case II. S is $R_1^3; R_2^3$.

(4) implies that $\langle E_1, \ldots, E_{\ell+1} | \{p\} R_1^3 \{r\} \rangle$ and $\langle E_1, \ldots, E_{\ell+1} | \{r\} R_2^3 \{q\} \rangle$ are valid, where $r \equiv sp(p, E_1, \ldots, E_{\ell+1} | R_1^3)$.
$c(E_{\ell+1} | R_1^3) \prec_\ell c(E_{\ell+1} | S)$, $c(E_{\ell+1} | R_2^3) \prec_\ell c(E_{\ell+1} | S)$ and both $E_1 \cdot \ldots \cdot E_\ell \cdot E_{\ell+1} \cdot R_1^3$ and $E_1 \cdot \ldots \cdot E_\ell \cdot E_{\ell+1} \cdot R_2^3$ are stable sequences, so by the induction hypothesis we have

$$\vdash \Gamma_0 \to \langle E, E_{\ell+1}[\tilde{Q}_\ell \leftarrow \tilde{P}_\ell] | \{p\} R_1^3 [\tilde{Q}_\ell \leftarrow \tilde{P}_\ell] \{r\} \rangle$$

and

$$\vdash \Gamma_0 \to \langle E, E_{\ell+1}[\tilde{Q}_\ell \leftarrow \tilde{P}_\ell] | \{r\} R_2^3 [\tilde{Q}_\ell \leftarrow \tilde{P}_\ell] \{q\} \rangle.$$

Now by the rule of composition we get (5).

Case III. S is <u>if</u> e <u>then</u> R_1^3 <u>else</u> R_2^3 <u>fi</u>.

(4) implies that $\langle E_1, \ldots, E_{\ell+1} | \{p \wedge e\} R_1^3 \{q\} \rangle$ and $\langle E_1, \ldots, E_{\ell+1} | \{p \wedge \neg e\} R_2^3 \{q\} \rangle$ are valid. Analogously to the case II by the induction hypothesis and the rule of conditional statements (5) holds.

Case IV. S is $\langle P_i \Leftarrow \langle S_i \rangle \rangle_{i=1}^n ; R^3$.

Let $P_1', \ldots, P_n'$ be some procedure variables which do not occur in $E_1, \ldots, E_{\ell+1}, S$ and are different from all Q_j^i-s. Denote $(P_1, \ldots, P_n)$ by $\overline{P}$ and $(P_1', \ldots, P_n')$ by $\overline{P}'$. Let $\overline{P}_{k_1}, \ldots, \overline{P}_{k_m}$ be the list of those sequences $\overline{P}_i$-s which do not contain any P_j as an element and let $\overline{Q}_{k_1}, \ldots, \overline{Q}_{k_m}$ be the corresponding list of $\overline{Q}_i$-s. For any statement S_0 denote $S_0[\overline{Q}_{k_1}/\overline{P}_{k_1}] \ldots [\overline{Q}_{k_m}/\overline{P}_{k_m}]$ by $\tilde{S}_0$. Observe that by the definition of substitution and our assumptions about procedure variables

$$S[\tilde{Q}_\ell \leftarrow \tilde{P}_\ell] \equiv \langle P_i \Leftarrow \langle \tilde{S}_i \rangle\rangle_{i=1}^{n}; \tilde{R}^3,$$

$$S_i[\overline{P}'/\overline{P}][\tilde{Q}_\ell \leftarrow \tilde{P}_\ell] \equiv \tilde{S}_i[\overline{P}'/\overline{P}],$$

$$R_i^3[\overline{P}'/\overline{P}][\tilde{Q}_\ell \leftarrow \tilde{P}_\ell] \equiv \tilde{R}^3[\overline{P}'/\overline{P}].$$

By (4) and lemma 2

$$\langle E_1,\ldots,E_{\ell+1},\langle P_i' \Leftarrow \langle S_i[\overline{P}'/\overline{P}]\rangle\rangle_{i=1}^{n} \mid \{p\}R^3[\overline{P}'/\overline{P}]\{q\}\rangle$$

is valid. $E_1 \cdot \ldots \cdot E_{\ell+1}, \langle P_i' \Leftarrow \langle S_i[\overline{P}'/\overline{P}]\rangle\rangle_{i=1}^{n} \cdot R^3[\overline{P}'/\overline{P}]$ is a stable sequence and $c(E_{\ell+1},\langle P_i' \Leftarrow \langle S_i[\overline{P}'/\overline{P}]\rangle\rangle_{i=1}^{n} \mid R^3[\overline{P}'/\overline{P}]) \prec_\ell c(E_{\ell+1}\mid S)$, so by the induction hypothesis and the above identities

$$\vdash \Gamma_0 \rightarrow \langle E, E_{\ell+1}[\tilde{Q}_\ell \leftarrow \tilde{P}_\ell], \langle P_i' \Leftarrow \langle \tilde{S}_i[\overline{P}'/\overline{P}]\rangle\rangle_{i=1}^{n} \mid \{p\}\tilde{R}^3[\overline{P}'/\overline{P}]\{q\}\rangle.$$

Now by the rule of procedure declarations

$$\vdash \Gamma_0 \rightarrow \langle E, E_{\ell+1}[\tilde{Q}_\ell \leftarrow \tilde{P}_\ell] \mid \{p\}\langle P_i \Leftarrow \langle \tilde{S}_i\rangle\rangle_{i=1}^{n}; \tilde{R}^3\{q\}\rangle$$

which due to the above identities proves (5).

Case V. S is $\underline{\text{var}}\ x; R^1$.

Let y be a simple variable which does not occur in $E_1,\ldots,E_{\ell+1},S,p$ or q. Since (4) holds, by lemma 16 $\langle E_1,\ldots,E_{\ell+1} \mid \{p\}R^1[y/x]\{q\}\rangle$ is valid. $c(E_{\ell+1}\mid R^1[y/x]) \prec_\ell c(E_{\ell+1}\mid S)$, so by the inductive assumption

$$\vdash \Gamma_0 \rightarrow \langle E, E_{\ell+1}[\tilde{Q}_\ell \leftarrow \tilde{P}_\ell] \mid \{p\}R^1[y/x][\tilde{Q}_\ell \leftarrow \tilde{P}_\ell]\{q\}\rangle.$$

Now by the rule of variable declarations we get (5).

Case VI. S is $\underline{\text{array}}\ a; R^2$.

Analogous to the case V using lemma 17 and the rule of array declarations.

Case VII. S is P.

Since $E_1 \cdot \ldots \cdot E_\ell \cdot E_{\ell+1} \cdot S$ is a stable sequence, P is declared in

$E_1,\ldots,E_{\ell+1}$.

Subcase 1^o. P is declared in $E_{\ell+1}$.

Let ϕ denote the empty system of procedure declarations. (4) implies that $\langle E_1,\ldots,E_{\ell+1},\phi|\{p\}P\{q\}\rangle$ is valid. $E_1\cdot\ldots\cdot E_{\ell+1}\cdot\phi\cdot P$ is a stable sequence and $c(\phi|P)\prec_\ell c(E_{\ell+1}|S)$ since $E_{\ell+1}$ is not empty. Hence by the induction hypothesis

$$\vdash \Gamma(E_1,\ldots,E_{\ell+1},\tilde{Q}_{\ell+1},F,G) \rightarrow$$

$$\langle E,E_{\ell+1}[\tilde{Q}_\ell \leftarrow \tilde{P}_\ell],\phi[\tilde{Q}_{\ell+1} \leftarrow \tilde{P}_{\ell+1}]|\{p\}P[\tilde{Q}_{\ell+1} \leftarrow \tilde{P}_{\ell+1}]\{q\}\rangle,$$

i.e.

$$\vdash \Gamma_0,\Gamma_1 \rightarrow \langle E,E_{\ell+1}[\tilde{Q}_\ell \leftarrow \tilde{P}_\ell]|\{p\}P[\tilde{Q}_\ell \leftarrow \tilde{P}_\ell][\bar{Q}_{\ell+1}/\bar{P}_{\ell+1}]\{q\}\rangle, \tag{6}$$

where

$$\Gamma_1 = \{p_1^{\ell+1}\}Q_1^{\ell+1}\{q_1^{\ell+1}\},\ldots,\{p_{n_{\ell+1}}^{\ell+1}\}Q_{n_{\ell+1}}^{\ell+1}\{q_{n_{\ell+1}}^{\ell+1}\}.$$

By the definition of $p_i^{\ell+1}$-s and $q_i^{\ell+1}$-s for $i = 1,\ldots,n_{\ell+1}$ $\langle E_1,\ldots,E_{\ell+1},\phi|\{p_i^{\ell+1}\}S_i^{\ell+1}\{q_i^{\ell+1}\}\rangle$ is valid. $c(\phi|S_i^{\ell+1})\prec_\ell c(E_{\ell+1}|S)$, so by the inductive assumption for $i = 1,\ldots,n_{\ell+1}$

$$\vdash \Gamma_0,\Gamma_1 \rightarrow \langle E,E_{\ell+1}[\tilde{Q}_\ell \leftarrow \tilde{P}_\ell]|\{p_i^{\ell+1}\}S_i^{\ell+1}[\tilde{Q}_{\ell+1} \leftarrow \tilde{P}_{\ell+1}]\{q_i^{\ell+1}\}\rangle,$$

i.e.

$$\vdash \Gamma_0,\Gamma_1 \rightarrow \langle E,E_{\ell+1}[\tilde{Q}_\ell \leftarrow \tilde{P}_\ell]|\{p_i^{\ell+1}\}S_i^{\ell+1}[\tilde{Q}_\ell \leftarrow \tilde{P}_\ell][\bar{Q}_{\ell+1}/\bar{P}_{\ell+1}]\{q_i^{\ell+1}\}\rangle. \tag{7}$$

Now by (6) and (7) using the rule of the procedure calls we get

$$\vdash \Gamma_0 \leftarrow \langle E,E_{\ell+1}[\tilde{Q}_\ell \leftarrow \tilde{P}_\ell]|\{p\}P[\tilde{Q}_\ell \leftarrow \tilde{P}_\ell]\{q\}\rangle$$

what was to be proved.

Subcase 2^o. P is declared in $E_1,\ldots,E_\ell$.

Observe that the proof from the previous subcase does not work here because $E_{\ell+1}$ can now be empty and then $c(\phi,P) = c(E_{\ell+1}|S)$. In fact,

we cannot use now any induction hypothesis. For some $i \in \{1,\ldots,\ell\}$ and $j \in \{1,\ldots,n_i\}$ $P \equiv P^i_j$. Since $E_1 \cdot \ldots \cdot E_\ell \cdot E_{\ell+1} \cdot P$ is a stable sequence, (4) in conjunction with lemma 3 implies that $\langle E_1,\ldots,E_i \mid \{p\}P^i_j\{q\}\rangle$ is valid. Let p' and q' be assertions obtained from respectively p and q by renaming all bound occurrences of variables from $var(E_1,\ldots,E_i)$ and F. Since $\models p' \leftrightarrow p$ and $\models q' \leftrightarrow q$, by the soundness of the consequence rule we get that

(8) $\langle E_1,\ldots,E_i \mid \{p'\}P^i_j\{q'\}\rangle$ is valid.

Let $\overline{F}_0$ be a sequence of completely new, different simple variables of the same length as $\overline{F}$ and let $\overline{G}_0$ be a sequence of completely new, different array variables of the same length as $\overline{G}$. Let $p_0 \equiv p'[\overline{F}_0/\overline{F}][\overline{G}_0/\overline{G}]$ and $q_0 \equiv q'[\overline{F}_0/\overline{F}][\overline{G}_0/\overline{G}]$. (8), the soundness of the substitution rule and lemma 18 imply that

(9) $\langle E_1,\ldots,E_i \mid \{p_0\}S^i_j\{q_0\}\rangle$ is valid.

To shorten notation let $\overline{z} = \overline{var(E_1,\ldots,E_i)}$ and $\overline{c} = \overline{array(E_1,\ldots,E_i)}$. By the definition $p^i_j \equiv \overline{z} = \overline{F} \wedge \overline{c} = \overline{G}$. We are to prove that

(10) $\vdash \Gamma_0 \rightarrow \langle E,E_{\ell+1}[\tilde{Q}_\ell \leftarrow \tilde{P}_\ell] \mid \{p\}Q^i_j\{q\}\rangle$.

By the invariance axiom

(11) $\vdash \Gamma_0 \rightarrow \langle E,E_{\ell+1}[\tilde{Q}_\ell \leftarrow \tilde{P}_\ell] \mid \{p_0[\overline{F}/\overline{z}][\overline{G}/\overline{c}]\}Q^i_j\{p_0[\overline{F}/\overline{z}][\overline{G}/\overline{c}]\}\rangle$.

By the selection axiom

(12) $\vdash \Gamma_0 \rightarrow \langle E,E_{\ell+1}[\tilde{Q}_\ell \leftarrow \tilde{P}_\ell] \mid \{p^i_j\}Q^i_j\{q^i_j\}\rangle$.

(11) and (12) imply by the consequence rule and conjunction rule that

$$\vdash \Gamma_0 \rightarrow \langle E,E_{\ell+1}[\tilde{Q}_\ell \leftarrow \tilde{P}_\ell] \mid \{p_0[\overline{F}/\overline{z}][\overline{G}/\overline{c}] \wedge p^i_j\}Q^i_j\{p_0[\overline{F}/\overline{z}][\overline{G}/\overline{c}] \wedge q^i_j\}\rangle.$$

Observe that $\models (p_0 \wedge p^i_j) \rightarrow (p_0[\overline{F}/\overline{z}][\overline{G}/\overline{c}] \wedge p^i_j)$, so by the rule of consequence

(13) $\vdash \Gamma_0 \rightarrow \langle E,E_{\ell+1}[\tilde{Q}_\ell \leftarrow \tilde{P}_\ell] \mid \{p_0 \wedge p^i_j\}Q^i_j\{p_0[\overline{F}/\overline{z}][\overline{G}/\overline{c}] \wedge q^i_j\}\rangle$.

We now show that

(14) $\quad \models (p_0[\overline{F}/\overline{z}][\overline{G}/\overline{c}] \wedge q^i_j) \to q_0.$

Let $\varepsilon \in Env$ be defined for all relevant variables and let $\sigma \in \Sigma$. Assume that $T(p_0[\overline{F}/\overline{z}][\overline{G}/\overline{c}] \wedge q^i_j)(\varepsilon,\sigma) = \underline{T}$. We are to show that

(15) $\quad T(q_0)(\varepsilon,\sigma) = \underline{T}.$

Since $T(q^i_j)(\varepsilon,\sigma) = \underline{T}$, by the definition of q^i_j there exists $\sigma' \in \Sigma$ such that $M(E_1,\ldots,E_i|S^i_j)(\varepsilon,\sigma') = \sigma$ and $T(p^i_j)(\varepsilon,\sigma') = \underline{T}$. In view of (9) to show (15) it is enough to prove that $T(p_0)(\varepsilon,\sigma') = \underline{T}$. Observe that $\models (p_0[\overline{F}/\overline{z}][\overline{G}/\overline{c}] \wedge p^i_j) \to p_0$. Since $T(p^i_j)(\varepsilon,\sigma') = \underline{T}$, it is sufficient to show that $T(p_0[\overline{F}/\overline{z}][\overline{G}/\overline{c}])(\varepsilon,\sigma') = \underline{T}$. By lemma 13 for all $y \in dom(\varepsilon) \setminus var(E_1,\ldots,E_i)$ we have $\sigma'(\varepsilon(y)) = \sigma(\varepsilon(y))$. By lemma 15 for all $a \in AV \setminus array(E_1,\ldots,E_i)$ and $d \in D$ such that $(a,d) \in dom(\varepsilon)$ we have $\sigma'(\varepsilon(a,d)) = \sigma(\varepsilon(a,d))$. If $y \in SV$ is free in $p_0[\overline{F}/\overline{z}][\overline{G}/\overline{c}]$ then $y \notin var(E_1,\ldots,E_i)$. If $a \in AV$ is free in $p_0[\overline{F}/\overline{z}][\overline{G}/\overline{c}]$ then $a \notin array(E_1,\ldots,E_i)$. We assumed that $T(p_0[\overline{F}/\overline{z}][\overline{G}/\overline{c}])(\varepsilon,\sigma) = \underline{T}$, so the above in view of lemma 7 implies that $T(p_0[\overline{F}/\overline{z}][\overline{G}/\overline{c}])(\varepsilon,\sigma') = \underline{T}$, which proves (15). Thus (14) holds. (13) and (14) imply by the rule of consequence that

$$\vdash \Gamma_0 \to \langle E,E_{\ell+1}[\tilde{Q}_\ell \leftarrow \tilde{P}_\ell]|\{p_0 \wedge p^i_j\}Q^i_j\{q_0\}\rangle.$$

By the substitution rule

$$\vdash \Gamma_0 \to \langle E,E_{\ell+1}[\tilde{Q}_\ell \leftarrow \tilde{P}_\ell]|\{(p_0 \wedge p^i_j)[\overline{z}/\overline{F}][\overline{c}/\overline{G}]\}Q^i_j\{q_0\}\rangle$$

because the variables from $\overline{F}$ and $\overline{G}$ do not occur in q_0. But $(p_0 \wedge p^i_j)[\overline{z}/\overline{F}][\overline{c}/\overline{G}] \equiv p_0 \wedge \overline{z} = \overline{z} \wedge \overline{c} = \overline{c}$, so by the consequence rule

$$\vdash \Gamma_0 \to \langle E,E_{\ell+1}[\tilde{Q}_\ell \leftarrow \tilde{P}_\ell]|\{p_0\}Q^i_j\{q_0\}\rangle.$$

Since $p' \equiv p_0[\overline{F}/\overline{F}_0][\overline{G}/\overline{G}_0], q' \equiv q_0[\overline{F}/\overline{F}_0][\overline{G}/\overline{G}_0]$, $\models p \leftrightarrow p'$ and $\models q \leftrightarrow q'$, by the substitution rule and the consequence rule we get (10) what was to be proved. □

The proof of the completeness theorem is now immediate. Namely suppose

that $\langle E|\{p\}S\{q\}\rangle$ is normal. Then $E \cdot S$ is a stable sequence. Hence, if $\langle E|\{p\}S\{q\}\rangle$ is valid then by lemma 19 (taking $\ell = 0$) $\vdash \langle E|\{p\}S\{q\}$. So if $\langle E|\Gamma\rangle$ is a valid normal correctness phrase then by the above or by the axiom (A4) $\vdash \langle E|\gamma\rangle$ for each $\gamma \in \Gamma$. By the collection rule we get $\vdash \langle E|\Gamma\rangle$ what was to be proved.

An inspection of the proof of lemma 19 (case VII. 1^{o}) shows that to prove the completeness theorem actually a weaker rule of procedure calls is sufficient:

$$\frac{\Gamma, \{p_1\}P'_1\{q_1\}, \ldots, \{p_n\}P'_n\{q_n\} \rightarrow \langle E, E'|\{p_0\}P'_j\{q_0\}, \{p_i\}S'_i\{q_i\}_{i=1,\ldots,n}\rangle}{\Gamma \rightarrow \langle E, E'|\{p_0\}P_j\{q_0\}\rangle}$$

where $j \in \{1,\ldots,n\}$,

with the former restrictions and notation.

14. CONCLUSIONS

Since the proofs of this paper are so long and complex it would be useful to find a way of simplifying them. One possibility of achieving this could be by choosing a semantics better suited for this type of proofs. The fact that we have used a variant of a denotational semantics led us to very tedious considerations concerning the class Θ^E of meanings of procedure variables. To avoid it one could use the operational semantics defined in COOK [5] but with suitable changes concerning declarations to get a static scope. We would have then to consider an appropriate class C^E of functions assigning bodies to procedure variables needed for the definition of validity. We suspect that this class can be defined in a very simple way, namely: $C^E = \{K : K \in PV \rightarrow S$ and for all $P \in PV$ all variables (including procedure variables) occurring in $K(P)$ occur in $E\}$.

Another candidate is the operational semantics defined in CLARKE [4]. The advantage of this semantics is that it does not use environments. They are needed in our framework to deal with variable declarations and the use of them adds significantly to the complexity of proofs.

It would be interesting to investigate whether the use of any of the

above two semantics would simplify the proof of the soundness theorem.

ACKNOWLEDGEMENT.

I am grateful to J.W. de Bakker for a number of useful comments.

REFERENCES

[1] APT, K.R., *Equivalence of operational and denotational semantics for a fragment of* PASCAL, in: Formal Description of Programming Concepts (edited by E.J. Neuhold), pp. 139-162, North Holland (1978).

[2] APT, K.R., & J.W. DE BAKKER, *Semantics and proof theory of* PASCAL *procedures,* in: Proc. of the Fourth Colloquium Automata, Languages and Programming, Lecture Notes in Computer Science 52, pp. 30-44, Springer (1977).

[3] DE BAKKER, J.W., *Correctness proof for assignment statements,* Report IW 55/76, Mathematisch Centrum (1976).

[4] CLARKE, E.M., *Programming language constructs for which it is impossible to obtain good Hoare-like axioms,* Proc. of the Fourth ACM Symposium on Principles of Programming Languages, pp. 10-20 (1977) (first version: Technical Report No. 76-287, Computer Science Department, Cornell University (1976)).

[5] COOK, S.A., *Soundness and completeness of an axiom system for program verification,* SIAM Journal on Computing, vol. 7, nr. 1, pp. 70-91 (1978).

[6] DONAHUE, J.E., *Complementary definitions of programming language semantics,* Lecture Notes in Computer Science 42, Springer (1976).

[7] GORELICK, G.A., *A complete axiomatic system for proving assertions about recursive and non-recursive programs,* Technical Report nr. 75, Department of Computer Science, University of Toronto (1975).

[8] HAREL, D., A. PNUELI & J. STAVI, *A complete axiomatic system for proving deductions about recursive programs*, in: Proc. of the Ninth ACM Symposium on Theory of Computing, pp. 249-260 (1977).

[9] HOARE, C.A.R., *An axiomatic basis for computer programming*, Comm. ACM 12, pp. 576-580 (1969).

[10] HOARE, C.A.R., *Procedures and parameters: an axiomatic approach*, in: Symposium on Semantics of Algorithmic Languages (edited by E. Engeler), Lecture Notes in Mathematics 188, pp. 102-116, Springer (1971).

[11] LIPTON, R.J., *A necessary and sufficient condition for the existence of Hoare Logics*, in: 18th Symposium on Foundations of Computer Science, pp. 1-6 (1977).

[12] SCHWARZ, J., *Generic commands - a tool for partial correctness formalisms*, The Computer Journal, vol. 20, nr. 2, pp. 151-155 (1977).

[13] SCOTT, D. & J.W. DE BAKKER, *A theory of programs*, Notes of an IBM Vienna seminar, unpublished (1969).

[14] WAND, M., *A new incompletenss result for Hoare's system*, Journal of ACM, vol. 25, nr. 1, pp. 168-175 (1978).