A new Hoare-Calculus for Programs with Recursive Parameterless Procedures

by

Kurt Sieber

A 81/02

February 1981

Fachbereich 10 - Informatik
Universität des Saarlandes
6600 Saarbrücken
West Germany

1. Introduction

In this paper a programming language, consisting of while-programs with block structure and mutually recursive parameterless procedures is presented. Two different denotational semantics are defined for this language, viz. a dynamic and a static scope semantics. For each of these semantics a Hoare-style axiom system is presented and proved sound and relatively complete.

A similar approach to a Hoare-like system with recursive parameterless procedures was given in [APT 78], but Apt's proofs were rather complex. He proposed himself in [APT 79] to replace the denotational semantics by an operational one in order to overcome the difficulties in his proofs. This was done e. g. in [OLD] for a class of even more complex programming languages. In the present report we have unsed another way out, which essentially consists in

a) avoiding addresses by defining a nondeterministic semantics for blocks with variable declarations and

b) introducing algebraic variables, i. e. variables which do not occur in the programming language but only in the assertion language.

The advantage of a) is a rather technical one:
Using addresses, one needs environments, i. e. partial functions ε from the set of variables to the set of addresses, and states σ, i. e. total functions from the set of addresses to the set of data. A program is then usually interpreted as a partial function, which maps pairs (ε, σ) to states σ. Hence, in order to define the semantics of Hoare formulas, a first order predicate formula p must also be evaluated w. r. t. pairs (ε, σ). This leads to difficulties, because p cannot contain enough information about (ε, σ): p only contains information about the values $\sigma(\varepsilon(x))$ for finitely many variables x and e. g. does say nothing about the domain of ε.

For this reason we have avoided addresses by defining a non-deterministic semantics for blocks with variable declarations. For more details the reader is referred to section 3.

The advantage of b) is more essential:
With the aid of algebraic variables we get a Hoare calculus which seems to be more natural than Apt's or Olderog's versions. For a detailed discussion the reader is referred to section 4.2.

The paper is structured as follows:
In section 2 basic definitions and notations are introduced. In section 3 the programming language is presented and both semantics - for dynamic and static scope - are defined. In section 4 Hoare phrases are introduced. Section 5 and 6 consist of the proof of soundness and relative completeness of an axiom system for dynamic scope, and in section 7 it is explained, how the axiom system must be modified in order to get static scope. In the conclusion (section 8) the relevance of our results is shortly discussed.

2. Definitions and notations

2.1. First order predicate logic

A first order predicate language $\underline{L}$ is built up from three sets:
- a set $\underline{F}$ of *function symbols*,
- a set $\underline{P}$ of *predicate symbols*,
- an infinite set $\underline{V}$ of *variables*.

The *terms* and *formulas* of $\underline{L}$ are built up in the usual way. $\mathcal{T}(\underline{L})$ denotes the set of all terms and $\mathcal{E}xp(\underline{L})$ the set of all quantifier free formulas of $\underline{L}$. $\underline{L}$ itself denotes the set of all its formulas.

An *interpretation* for $\underline{L}$ is a pair $I = (\underline{D}, I_0)$, where
- $\underline{D}$ is a nonempty set (the *domain* of I)
- I_0 is a function which assigns
  to each k-ary $f \in \underline{F}$ a function $I_0(f) : \underline{D}^k \to \underline{D}$,
  to each k-ary $p \in \underline{P}$ a predicate $I_0(p) : \underline{D}^k \to \{\underline{true}, \underline{false}\}$

A *state* σ (for $\underline{L}$ and I) is defined to be a function $\sigma : \underline{V} \to \underline{D}$. $\underline{\Sigma}$ denotes the set of all states (for $\underline{L}$ and I). For every term t of $\underline{L}$ a value $I(t)(\sigma) \in \underline{D}$ and for every formula p of $\underline{L}$ a truth value $I(p)(\sigma) \in \{\underline{true}, \underline{false}\}$ are defined as usual.

Instead of $I(p)(\sigma) = \underline{true}$ we also write $\models_{I,\sigma} p$ · p is called *valid in* I (or: I-*valid*, notation: $\models_I p$), if $\models_{I,\sigma} p$ for all $\sigma \in \underline{\Sigma}$. p is called *valid* (notation: $\models p$), if $\models_I p$ for all interpretations I of $\underline{L}$.

For arbitrary sets $\underline{M}$, $\underline{N}$ and a (partial) function $g : \underline{M} \rightsquigarrow \underline{N}$ the following notations will be used:
- dom (g) denotes the domain of g
- if $m \in \underline{M}$ and $n \in \underline{N}$, then g [n/m] denotes the function g' with:
  dom (g') = dom (g) $\cup$ {m}
  g' (m) = n
  g' (m') = g (m') for every m' $\neq$ m
- if $\underline{S} \subseteq$ dom (g), then $g|_{\underline{S}}$ denotes the restriction of g to $\underline{S}$.

For $p \in \underline{L}$, $x \in \underline{V}$ and a term t of $\underline{L}$, p_x^t denotes the formula, obtained by substituting t for every free occurence of x (being understood that bound variables of p have been renamed previously, if necessary). The substitution lemma of first order predicate logic then states that

$$I\,(p_x^t)\,(\sigma) = I\,(p)\,(\sigma[I(t)(\sigma)/x]).$$

The definition and the lemma may be generalized in the usual way for simultaneous substitution (denoted by $p_{x_1,\ldots,x_n}^{t_1,\ldots,t_n}$)

2.2. Partial orders

Let $(\underline{P}, \sqsubseteq)$ be a (reflexive) partial order.
If $\underline{S} \subseteq \underline{P}$, then $\bigsqcup \underline{S}$ denotes the lowest upper bound of $\underline{S}$ (if it exists) and for $a, b \in \underline{P}$ we write $a \sqcup b$ instead of $\bigsqcup \{a, b\}$
Furtheron let $\underline{M}$ be an arbitrary set.

Then the set $(\underline{M} \to \underline{P})$ of total functions from $\underline{M}$ to $\underline{P}$ is made into a partial order by defining for $f, g \in (\underline{M} \to \underline{P})$:

$$f \sqsubseteq g \quad \Leftrightarrow \quad f(m) \sqsubseteq g(m) \quad \text{for every } m \in \underline{M}$$

If $\underline{S} \subseteq (\underline{M} \to \underline{P})$, then $\bigsqcup \underline{S}$ exists if $\bigsqcup \{f(m) \mid f \in \underline{S}\}$ exists for every $m \in \underline{M}$, and then: $(\bigsqcup \underline{S})(m) = \bigsqcup \{f(m) \mid f \in \underline{S}\}$ for every $m \in \underline{M}$.

Let $(\underline{P}_1, \sqsubseteq), \ldots, (\underline{P}_n, \sqsubseteq)$ be partial orders.
Then $\underline{P}_1 \times \ldots \times \underline{P}_n$ is made into a partial order by defining
$(a_1,\ldots,a_n) \sqsubseteq (b_1,\ldots,b_n) \quad \Leftrightarrow \quad a_i \sqsubseteq b_i \quad \text{for } i = 1,\ldots, n$

If $\underline{S} \subseteq \underline{P}_1 \times \ldots \times \underline{P}_n$, then $\underline{S}$ exists iff $\bigsqcup \{pr_i(x) \mid x \in \underline{S}\}$ exists for $i = 1, \ldots, n$ (where $pr_i: \underline{P}_1 \times \ldots \times \underline{P}_n$ denotes the i-th projection), and then $pr_i\,(\bigsqcup \underline{S}) = \bigsqcup \{pr_i(x) \mid x \in \underline{S}\}$ for $i = 1, \ldots, n$.

A subset $\underline{S}$ of a partial order $\underline{P}$ is called a *Δ-set*, if $a \sqcup b$ exists for every $a, b \in \underline{S}$ and lies in $\underline{S}$.

Let $(\underline{P}_1, \sqsubseteq)$ and $(\underline{P}_2, \sqsubseteq)$ be partial orders and $f: \underline{P}_1 \to \underline{P}_2$.
f is called *monotonic*, if for every $a, b \in \underline{P}_1$:

$$a \sqsubseteq b \quad \text{implies} \quad f(a) \sqsubseteq f(b)$$

f is called *Δ-continuous*, if for every Δ-set $\underline{S} \subseteq \underline{P}_1$:
If $\bigsqcup \underline{S}$ exists, then $\bigsqcup f(\underline{S})$ exists and $\bigsqcup f(\underline{S}) = f(\bigsqcup \underline{S})$
(These definitions are taken from [ADJ].)

Let $(\underline{P}, \underline{\sqsubseteq})$ be a partial order with a least element $\underline{\perp}$, and let $f : \underline{P} \to \underline{P}$ be Δ-continuous. Then the fixpoint theorem says:
If $\{f^i(\underline{\perp}) \mid i \in \underline{\mathbb{N}}\}$ exists, then it is the least fixpoint of f.

2.3. Programming languages

We define a *syntax* to be a function $\mathcal{S}$, which assigns to everey first order predicate language $\underline{L}$ a programming language $\mathcal{S}(\underline{L})$ and which is monotonic w. r. t. "$\underline{c}$". The elements of $\mathcal{S}(\underline{L})$ are called programs. In this paper only one syntax will be introduced, viz. $\mathcal{S}_{proc}$, the syntax of while-programs with mutually recursive parameterless procedures.

A *semantics* (for a syntax $\mathcal{S}$) is defined to be a function $\mathcal{M}$, which assigns to every pair (I, S) where
- I is an interpretation for a first order predicate language $\underline{L}$,
- $S \in \mathcal{S}(\underline{L})$ for the same $\underline{L}$,

a relation $\mathcal{M}(I, S) \underline{c} \underline{\Sigma} \times \underline{\Sigma}$, the "meaning" of S. Defining $\mathcal{M}(I, S)$ as a relation - and not necessarily as a partial function - allows to consider also nondeterministic programs. We shall introduce two semantics for $\mathcal{S}_{proc}$: $\mathcal{M}_{dyn}$ (dynamic scope) and $\mathcal{M}_{stat}$ (static scope). As both semantics are never considered simultaneously, $\mathcal{M}_{dyn}(I, S)$ and $\mathcal{M}_{stat}(I, S)$ will both be denoted by $\mathcal{M}_I(S)$.

2.4. Hoare logic

Let $\underline{L}_E$ and $\underline{L}_A$ be two first order predicate languages, such that:
- $\underline{L}_E \underline{c} \underline{L}_A$ and
- $\underline{V}_A \setminus \underline{V}_E$ is infinite

(where $\underline{V}_A$ and $\underline{V}_E$ denote the variable sets of $\underline{L}_A$ and $\underline{L}_E$)

The set of *Hoare formulas* for $\underline{L}_A$ and $\mathcal{S}(\underline{L}_E)$ is then defined as

$$\mathcal{H}(\underline{L}_A, \mathcal{S}(\underline{L}_E)) = \underline{L}_A \cup \{\{p\}S\{q\} \mid p, q \in \underline{L}_A, S \in \mathcal{S}(\underline{L}_E)\}$$

i. e. it is an extension of $\underline{L}_A$.

$\underline{L}_E$ is called the *expression language* and $\underline{L}_A$ the *assertion language* of $\mathcal{K}(\underline{L}_A, \mathcal{S}(\underline{L}_E))$, the variables of $\underline{V}_E$ are called *program variables*, the variables of $\underline{V}_A \setminus \underline{V}_E$ are called *algebraic variables*.

Let I be an interpretation for $\underline{L}_A$ and $\mathcal{M}$ a semantics for $\mathcal{S}$. Let $\underline{\Sigma}$ be the set of states for $\underline{L}_A$ and I. As $\mathcal{S}$ is monotonic, $\mathcal{S}(\underline{L}_E) \subseteq \mathcal{S}(\underline{L}_A)$, and so $\mathcal{M}_I(S) \subseteq \underline{\Sigma} \times \underline{\Sigma}$ is defined for every $S \in \mathcal{S}(\underline{L}_E)$. Hence a truth value $I(h)(\sigma)$ can be assigned to every $h \in \mathcal{K}(\underline{L}_A, \mathcal{S}(\underline{L}_E))$ and $\sigma \in \underline{\Sigma}$ by:

a) If $h = p \in \underline{L}_A$, then $I(p)(\sigma)$ is defined as usual.

b) If $h = \{p\}\ S\ \{q\}$, then

$$I(\{p\}\ S\ \{q\}, \sigma) = \underline{true}$$

$$\overset{\text{def.}}{\leftrightarrow} \begin{cases} \text{If } I(p)(\sigma) = \underline{true} \text{ and } (\sigma, \sigma') \in \mathcal{M}_I(S) \\ \qquad \text{then } I(q)(\sigma') = \underline{true} \end{cases}$$

$$\leftrightarrow \begin{cases} I(p)(\sigma) = \underline{false} \text{ or:} \\ I(p)(\sigma) = \underline{true} \text{ and } I(q)(\sigma') = \underline{true} \text{ for every } \sigma' \\ \qquad\qquad\qquad\qquad \text{with } (\sigma,\sigma') \in \mathcal{M}_I(S) \end{cases}$$

Because of a), Hoare logic is an extension of first order predicate logic. The notations $\models_{I,\sigma} \ldots, \models_I \ldots, \models \ldots$ and the according definitions are also generalized from predicate logic to Hoare logic, in particular $\models_I \{p\}\ S\ \{q\}$ means:

"For every $\sigma \in \underline{\Sigma}$: If $I(p)(\sigma) = \underline{true}$ then $I(q)(\sigma') = \underline{true}$ for every σ' with $(\sigma, \sigma') \in \mathcal{M}_I(S)$"

In this case $\mathcal{M}_I(S)$ is called partially correct with respect to p and q. More generally, an arbitrary relation $\underline{R}$ is called *partially correct* with respect to p and q, if for every $\sigma \in \underline{\Sigma}$: $I(p)(\sigma) = \underline{true}$ and $(\sigma, \sigma') \in \underline{R}$ implie: $I(q)(\sigma') = \underline{true}$.

Note that all the definitions also depend on the semantics $\mathcal{M}$, but as $\mathcal{M}$ will always be known from the context, it is omitted in the notations.

2.5. Hoare axiom systems

A *Hoare axiom system* AX for a syntax $\mathcal{S}$ is a set of axiom schemes and inference rules which allow, for arbitrary languages $\underline{L}_E$ and $\underline{L}_A$, to derive Hoare formulas of $\mathcal{H}(\underline{L}_A, \mathcal{S}(\underline{L}_E))$ from formulas of $\underline{L}_A$. We do not require that AX works with Hoare formulas only: We shall introduce two axiom systems, AXDYN (dynamic scope) and AXSTAT (static scope) for $\mathcal{S}_{proc}$, which will both use more general constructs for the derivation of Hoare formulas, so-called Hoare phrases (see [APT 78]).

We omit a more precise definition of "axiom scheme", "inference rule" and related notions, but only introduce the notation "$\vdash_I h$" for $h \in \mathcal{H}(\underline{L}_A, \mathcal{S}(\underline{L}_E))$ to stand for "h is derivable from I-valid wff's of $\underline{L}_A$" (by an axiom system, which will always be known from the context).

An axiom system AX for $\mathcal{S}$ is said to be *sound* w. r. t. a semantics $\mathcal{M}$, if for every expression language $\underline{L}_E$, assertion language $\underline{L}_A$ and interpretation I for $\underline{L}_A$:

If $\vdash_I h$ for some $h \in \mathcal{H}(\underline{L}_A, \mathcal{S}(\underline{L}_E))$, then $\models_I h$.

We shall prove AXDYN to be sound w. r. t. $\mathcal{M}_{dyn}$ and AXSTAT w. r. t. $\mathcal{M}_{stat}$.

In order to examine questions of completeness of Hoare axiom systems the following notions have been introduced by Cook (see [COOK]):

a) Let $\underline{L}$ be a first order predicate language, I an interpretation for $\underline{L}$ and $\underline{\Sigma}$ the set of states. If $p \in \underline{L}$ and $\underline{R} \subseteq \underline{\Sigma} \times \underline{\Sigma}$ is a relation, then $q \in \underline{L}$ is called a *strongest postcondition* of p and $\underline{R}$ (w. r. t. I) if

$I(q)(\sigma') = \underline{true} \leftrightarrow$ there exists $\sigma \in \underline{\Sigma}$ such that $I(p)(\sigma) = \underline{true}$ and $(\sigma, \sigma') \in \underline{R}$

q is uniquely determined up to equivalence in I, so that one usually speaks of <u>the</u> strongest postcondition.

b) Let $\underline{L}_E$, $\underline{L}_A$, I, $\mathcal{S}$ and $\mathcal{M}$ be as above.
$\underline{L}_A$ is called I-*expressive* w. r. t. $\mathcal{S}(\underline{L}_E)$ and $\mathcal{M}$ if

- $\underline{L}_E$ contains the equality symbol, which is interpreted by I as equality on $\underline{D}$.
- For every $p \in \underline{L}_A$ and $S \in \mathcal{S}(\underline{L}_E)$, $\underline{L}_A$ contains the strongest postcondition of p and $\mathcal{M}_I(S)$.

c) A Hoare axiom system AX for $\mathcal{S}$ is said to be *complete in the sense of Cook* w. r. t. a semantics $\mathcal{M}$ if for every $\underline{L}_E$, $\underline{L}_A$ and interpretation I for $\underline{L}_A$, such that $\underline{L}_A$ is I-expressive with respect to $\mathcal{S}(\underline{L}_E)$ and $\mathcal{M}$:

If $\models_I h$ for some $h \in \mathcal{S}(\underline{L}_E))$, then $\vdash_{AX} h$.

We shall <u>not</u> prove completeness in the sense of Cook for our axiom systems, but relative completeness in a <u>new sense</u>: In the sequel "expressiveness" will be replaced by a notion called "strong expressiveness", and as this notion is indeed stronger, a weaker completeness result is obtained. The relations between these alternative notions of relative completeness are discussed in section 8.

3. The programming language $\mathcal{S}_{proc}(\underline{L})$

3.1. Syntax

Let PV be an infinite set of symbols, called *procedure variables*. For every first order predicate language L we first define the sets $\mathfrak{St}(\underline{L})$ of *statements* (with typical elements St, ...) and $\mathfrak{E}(\underline{L})$ of *enviroumenta* (with typical elements E, ...) by:

$$
\begin{array}{lll}
St ::= x := t & & (x \in \underline{V},\ t \in \mathcal{T}(\underline{L})) \\
\quad | St_1;\ St_2 & & \\
\quad | \underline{if}\ e\ \underline{then}\ St_1\ \underline{else}\ St_2\ \underline{fi} & & (e \in \mathcal{E}xp(\underline{L})) \\
\quad | \underline{while}\ e\ \underline{do}\ St\ \underline{od} & & (e \in \mathcal{E}xp(\underline{L})) \\
\quad | \underline{call}\ P & & (P \in \underline{PV}) \\
\quad | B & &
\end{array}
$$

B ::= begin D E St end

D is a (possibly empty) list of variable declarations:
new x_1; ... ; new x_n; where x_1, ..., x_n are different variables.

E is a (possibly empty) list of procedure declarations:
proc P_1 : B_1; ...; proc P_m : B_m; where P_1, ..., P_m are different procedure variables. ε denotes the empty list.

E will also be considered as a partial function from PV to $\mathfrak{St}(\underline{L})$ with domain dom (E) = $\{P_1, \ldots, P_m\}$ and with $E(P_i) = B_i$ for i = 1, ..., m. Note that a declaration proc P : B may occur in E (as a substring) without P being an element of dom (E).

An occurence of $P \in \underline{PV}$ (resp. $x \in \underline{V}$) in a statement St is said to be *bound* , if it is contained in a substring "begin D E St' end" of St, for which $P \in$ dom (E) (resp. new x occurs in D). Otherwise the occurence of P (resp.x) is called *free*. St is said to be *closed*, if every occurence of a procedure variable in St is bound.

Now the programs of $\mathcal{S}_{proc}(\underline{L})$ are defined to be the closed statements of $\mathfrak{St}(\underline{L})$.

For the definition of the semantics we shall need another syntactical construct, the so-called units:
Let $\underline{L}$ be a first order predicate language. Then a *unit* of $\underline{L}$ is a pair (E | St) where E $\in$ $\mathfrak{E}(\underline{L})$ and St $\in$ $\mathfrak{St}(\underline{L})$. (We use "|" instead of "," for better readability.) The set of all units of $\underline{L}$ is denoted by $\mathcal{U}(\underline{L})$.

The notions "occurs free in", "closed", ... are transferred from statements to units by defining a property of (E | St) as the corresponding property of the statement "begin E St end", e. g.: (E | St) is called *closed*, if "begin E St end" is closed, ...

$\mathcal{U}(\underline{L})$ is made to a well-founded set (WFS) by defining:

$(E_1|St_1) \prec (E_2|St_2) \overset{\leftrightarrow}{\text{def.}} l(E_1) + l(St_1) < l(E_2) + l(St_2)$

or: $l(E_1) + l(St_1) = l(E_2) + l(St_2)$ and $l(St_1) < l(St_2)$

(where $l(...)$ denotes the length of a string)

This well-founded-set-structure will be needed later for defining $\mathcal{M}_{dyn}$ and $\mathcal{M}_{stat}$ recursively.

3.2. Program relations and procedure assignments

We have defined the meaning of a program to be a relation on states. But not every such relation "makes sense" as meaning of a program, e. g. we want to exclude (meanings of) programs which change the values of infinitely many variables or which depend on the initial values of infinitely many variables. This is done by the following definitions:

Let $\underline{L}$ be a first order predicate language, I an interpretation for $\underline{L}$ and $\underline{\Sigma}$ the set of states for $\underline{L}$ and I.

Let $\underline{R} \subseteq \underline{\Sigma} \times \underline{\Sigma}$ be a relation and $\underline{F} \subseteq \underline{V}$.

a) The relation $\underline{R}^{\underline{F}}$ is defined by:

$$\underline{R}^{\underline{F}} = \{(\tau,\tau') \mid \text{there exists } (\sigma,\sigma') \in \underline{R} \text{ such that } \tau|_{\underline{F}} = \sigma|_{\underline{F}},\ \tau'|_{\underline{F}} = \sigma'|_{\underline{F}} \text{ and } \tau|_{\underline{V}\setminus\underline{F}} = \tau'|_{\underline{V}\setminus\underline{F}}\}$$

b) $\underline{R}$ is called $\underline{F}$-*bounded*, if $\underline{R} = \underline{R}^{\underline{F}}$, i. e. if for every $(\sigma, \sigma') \in \underline{R}$:

- $\sigma|_{\underline{V}\setminus\underline{F}} = \sigma'|_{\underline{V}\setminus\underline{F}}$ (i. e. $\underline{R} \subseteq \underline{R}^{\underline{F}}$) and
- if $\tau|_{\underline{F}} = \sigma|_{\underline{F}}$, $\tau'|_{\underline{F}} = \sigma'|_{\underline{F}}$ and $\tau|_{\underline{V}\setminus\underline{F}} = \tau'|_{\underline{V}\setminus\underline{F}}$, then (τ, τ') also lies in $\underline{R}$ (i. e. $\underline{R}^{\underline{F}} \subseteq \underline{R}$)

c) $\underline{R}$ is called a *program relation* if there is a <u>finite</u> set $\underline{F}$ such that $\underline{R}$ is $\underline{F}$-bounded.

Part c) of the definition is exactly what we wanted:
A program relation $\underline{R}$ only changes the values of the (finitely many) variables of $\underline{F}$ and only depends on the initial values of the variables of $\underline{F}$.

In order to work with program relations we need the following lemma:

<u>Lemma 1:</u>

a) $\underline{R}^{\underline{F}} = (\underline{R}^{\underline{F}})^{\underline{F}}$, i. e. $\underline{R}^{\underline{F}}$ is $\underline{F}$-bounded.

b) If $\underline{R}$ is $\underline{F}_1$-bounded and $\underline{F}_1 \subseteq \underline{F}_2$,
then $\underline{R}$ is also $\underline{F}_2$-bounded.

c) $\underline{R}$ is $\underline{F}_1$-bounded and $\underline{F}_2$-bounded
$\Leftrightarrow$ $\underline{R}$ is $(\underline{F}_1 \cap \underline{F}_2)$-bounded.

d) If $\underline{R}_1$ is $\underline{F}_1$-bounded and $\underline{R}_2$ is $\underline{F}_2$-bounded,
then $\underline{R}_1 \cup \underline{R}_2$ is $(\underline{F}_1 \cup \underline{F}_2)$-bounded.

e) If $(R_i)_{i\in I}$ is a family of relations, then $(\bigcup_{i\in I} \underline{R}_i)^{\underline{F}} =$

$$\bigcup_{i\in I} \underline{R}_i^{\underline{F}},$$

especially $\bigcup_{i\in I} \underline{R}_i$ is $\underline{F}$-bounded, if every $\underline{R}_i$ is $\underline{F}$-bounded.

Proof:

a), b), d) and e) directly result from the definitions.

c) "➡":

Let $\underline{R}$ be $\underline{F}_1$-bounded and $\underline{F}_2$-bounded and $(\sigma, \sigma') \in \underline{R}$.

Then σ and σ' coincide outside $\underline{F}_1$ and outside $\underline{F}_2$, hence outside $\underline{F}_1 \cap \underline{F}_2$. This proves $\underline{R} \subseteq \underline{R}^{(\underline{F}_1 \cap \underline{F}_2)}$.

Moreover let $\tau, \tau' \in \underline{\Sigma}$ with

$$\tau|_{\underline{F}_1 \cap \underline{F}_2} = \sigma|_{\underline{F}_1 \cap \underline{F}_2}, \quad \tau'|_{\underline{F}_1 \cap \underline{F}_2} = \sigma'|_{\underline{F}_1 \cap \underline{F}_2} \quad \text{and}$$

$$\tau|_{\underline{V}\setminus(\underline{F}_1 \cap \underline{F}_2)} = \tau'|_{\underline{V}\setminus(F_1 \cap F_2)}$$

Then choose $\rho, \rho' \in \underline{\Sigma}$ with

$$\rho|_{\underline{F}_1} = \sigma|_{\underline{F}_1}, \ \rho|_{\underline{F}_2} = \tau|_{\underline{F}_2}, \ \rho'|_{\underline{F}_1} = \sigma'|_{\underline{F}_1}, \ \rho'|_{\underline{F}_2} = \tau'|_{\underline{F}_2} \quad \text{and}$$

$\rho|_{\underline{V}\setminus(\underline{F}_1 \cap \underline{F}_2)} = \rho'|_{\underline{V}\setminus(\underline{F}_1 \cap \underline{F}_2)}$ (note that this is possible, because

- τ and σ resp. τ' and σ' coincide on $\underline{F}_1 \cap \underline{F}_2$
- σ and σ' resp. τ and τ' coincide on $\underline{V}\setminus(\underline{F}_1 \cap \underline{F}_2)$)

Now, as $\underline{R}$ is $\underline{F}_1$-bounded, $(\rho, \rho') \in \underline{R}$ and then, as $\underline{R}$ is $\underline{F}_2$-bounded, also $(\tau, \tau') \in \underline{R}$. This proves $\underline{R}^{\underline{F}_1 \cap \underline{F}_2} \subseteq \underline{R}$.

"⬅":

is an immediate consequence of b)

Note that part c) of the lemma can be generalized to finitely many sets $\underline{F}_1, \ldots, \underline{F}_n$ by a standard induction argument. This implies that for every program relation $\underline{R}$ there is a least set $\underline{F}$, sucht that $\underline{R}$ is $\underline{F}$-bounded. This set is called the variable set of $\underline{R}$, and is denoted by "var ($\underline{R}$)". Furtheron, part d) of the lemma implies, that the union of two - and hence of finitely many - program relations is again a program relation.

We denote the set of all program relations (for given $\underline{L}$ and I) by $\underline{\mathcal{R}}$. Then $\underline{\mathcal{R}}$ is a partial order w. r. t. "$\underline{c}$", in which every finite subset has a lowest upper bound.

A *procedure assignment* is defined to be an mapping $\gamma : \underline{PV} \to \underline{\mathcal{R}}$, and the set of all procedure assignments is denoted by $\underline{\Gamma}$. $\underline{\Gamma}$ is made into a partial order as defined in 2.2., hence also every finite subset of $\underline{\Gamma}$ has a lowest upper bound. For every $\gamma \in \underline{\Gamma}$ and $\underline{F} \underline{c} \underline{V}$ the procedure assignment $\gamma^{\underline{F}} : \underline{PV} \to \underline{\mathcal{R}}$ is defined by $\gamma^{\underline{F}}(P) = \gamma(P)^{\underline{F}}$ for every $P \in \underline{PV}$.

3.3. Dynamic scope semantics

Let $\underline{L}$ be a first order predicate language, I an interpretation for $\underline{L}$ and $(E \mid St) \in \mathcal{U}(\underline{L})$. Then we first define a function $\mathcal{M}_I(E|St) : \underline{\Gamma} \to \underline{\mathcal{R}}$ inductively by:

i) $(\sigma, \sigma') \in \mathcal{M}_I(E|x:=t)(\gamma) \leftrightarrow \sigma' = \sigma\,[I(t)(\sigma)/x]$

ii) $(\sigma, \sigma') \in \mathcal{M}_I(E|St_1; St_2)(\gamma) \leftrightarrow$

There exists $\sigma'' \in \underline{\Sigma}$ sucht that $(\sigma, \sigma'') \in \mathcal{M}_I(E|St_1)(\gamma)$ and $(\sigma'', \sigma') \in \mathcal{M}_I(E|St_2)(\gamma)$

iii) $(\sigma, \sigma') \in \mathcal{M}_I(E|\underline{\text{if}}\ e\ \underline{\text{then}}\ St_1\ \underline{\text{else}}\ St_2\ \underline{\text{fi}})(\gamma) \leftrightarrow$

$I(e)(\sigma) = \underline{\text{true}}$ and $(\sigma, \sigma') \in \mathcal{M}_I(E|St_1)(\gamma)$

or $I(e)(\sigma) = \underline{\text{false}}$ and $(\sigma, \sigma') \in \mathcal{M}_I(E|St_2)(\gamma)$

iv) $(\sigma, \sigma') \in \mathcal{M}_I(E|\ \underline{\text{while}}\ e\ \underline{\text{do}}\ St\ \underline{\text{od}})(\gamma) \leftrightarrow$

There exist $n \in \mathbb{N}$ and $\sigma_1, \ldots, \sigma_n \in \underline{\Sigma}$ such that:

$\sigma = \sigma_1$, $\sigma' = \sigma_n$,

$(\sigma_k, \sigma_{k+1}) \in \mathcal{M}_I(E|St)(\gamma)$ for $k = 1, \ldots, n-1$,

$I(e)(\sigma_k) = \underline{\text{true}}$ for $k = 1, \ldots, n-1$ and

$I(e)(\sigma_n) = \underline{\text{false}}$

v) $\mathcal{M}_I(E|\ \underline{\text{begin}}\ St\ \underline{\text{end}})(\gamma) = \mathcal{M}_I(E|St)(\gamma)$

vi) $(\sigma, \sigma') \in \mathcal{M}_I(E_1|\ \underline{\text{begin}}\ \underline{\text{new}}\ x;\ DE_2St\ \underline{\text{end}})(\gamma) \leftrightarrow$

There exist $d \in \underline{D}$ and $\sigma'' \in \underline{\Sigma}$ such that:

$(\sigma[d/x], \sigma'') \in \mathcal{M}_I(E_1|\ \underline{\text{begin}}\ DE_2St\ \underline{\text{end}})(\gamma)$

and $\sigma' = \sigma''\ [\sigma(x)/x]$

vii) $\mathcal{M}_I(E_1|\ \underline{\text{begin}}\ \underline{\text{proc}}\ P : B;\ E_2St\ \underline{\text{end}})(\gamma) =$

$\mathcal{M}_I(E_1\ [B/P]\ |\ \underline{\text{begin}}\ E_2St\ \underline{\text{end}})(\gamma)$

viii) $\mathcal{M}_I(E|\ \underline{\text{call}}\ P)(\gamma)$

$$= \begin{cases} \gamma(P) & \text{if } P \notin \text{dom}(E) \\ \mu_i^{E,\gamma} & \text{if } E = \underline{\text{proc}}\ P_1 : B_1;\ \ldots;\ \underline{\text{proc}}\ P_m : B_m;\ \text{and } P = P_i \end{cases}$$

where $(\mu_1^{E,\gamma}, \ldots, \mu_m^{E,\gamma})$ is the least fixpoint of

$$\Phi^{E,\gamma} : \underline{\mathcal{R}}^m \rightarrow \underline{\mathcal{R}}^m$$

$$(\underline{R}_1, \ldots, \underline{R}_m) \mapsto (\mathcal{M}_I(\varepsilon\ |\ B_1)(\gamma[\underline{R}_1/P_1]\ldots[\underline{R}_m/P_m]),$$

$$\vdots$$

$$\mathcal{M}_I(\varepsilon\ |\ B_m)(\gamma[\underline{R}_1/P_1]\ldots[\underline{R}_m/P_m]))$$

To see that $\mathcal{M}_I(E|St) : \underline{\Gamma} \to \underline{\mathcal{R}}$ is always well-defined, one needs the following lemma, which can be proved by structural induction (remember the WFS-structure, defined on $\mathcal{U}(\underline{L})$):

Lemma 2:

a) $\mathcal{M}_I(E|St): \underline{\Gamma} \to \underline{\mathcal{R}}$ is a well-defined, Δ-continuous mapping.

b) If $\underline{F}$ contains all variables, which occur free in $(E|St)$, and if for every $P \in \underline{PV}$, which occurs free in $(E|St)$ the relation $\gamma(P)$ is $\underline{F}$-bounded, then $\mathcal{M}_I(E|St)(\gamma)$ is $\underline{F}$-bounded.

(Part b) of the lemma is necessary for the proof of a) in the case St = call P, i. e. clause viii) of the definition of $\mathcal{M}_I(E|St)$:

As $(\varepsilon|B_i) \prec (E|\ \underline{\text{call}}\ P)$ for $i = 1, \ldots, m$,

the induction hypothesis holds for every $(\varepsilon|B_i)$, i. e. for every $\gamma \in \underline{\Gamma}$ the mapping $\Phi^{E,\gamma}$ is Δ-continuous, and if $\underline{F}$ is a set, which contains all variables of the B_i's and all $var(\gamma(P))$, for which P occurs free in one of the B_i's, then $\Phi^{E,\gamma}$ maps m- tuples of $\underline{F}$-bounded relations again on m-tuples of $\underline{F}$-bounded relations. Hence $\bigsqcup_{i \in \mathbb{N}} (\Phi^{E,\gamma})^i (\emptyset, \ldots, \emptyset)$ exists in $\underline{\mathcal{R}}^m$ and is the least fixpoint of $\Phi^{E,\gamma}$.

A detailed proof of lemma 2 is left to the reader.)

Now, again by structural induction, it can be easily seen that $\mathcal{M}_I(E|St)(\gamma)$ depends only on those values $\gamma(P)$ of γ, for which P occurs free in $(E|St)$, especially is independent of γ, if $(E|St)$ is closed.

As programs were defined to be the closed statements of $\mathcal{St}(\underline{L})$, this leads to the final definition of $\mathcal{M}_{dyn}$:

For every $S \in \mathcal{P}_{proc}(\underline{L})$ and interpretation I for $\underline{L}$

$$\mathcal{M}_I(S) = \mathcal{M}_I(\varepsilon|S)(\gamma)$$

for an arbitrary $\gamma \in \underline{\Gamma}$.

The definition of the semantics $\mathcal{M}_{dyn}$ is standard, except for the treatment of variable declarations: Clause vi) means that, by the declaration new x, the variable x is assigned a random value $d \in \underline{D}$, then the rest of the block is executed and afterwards x is assigned its old value $\sigma(x)$.

3.4. Static scope semantics

A static scope semantics $\mathcal{M}_{stat}$ for $\mathcal{S}_{proc}$ can be defined by modifying $\mathcal{M}_{dyn}$ slightly:

For every environment E and variables $x_1,\ldots,x_n,x_1',\ldots,x_n' \in \underline{V}$ let $E^{x_1',\ldots,x_n'}_{x_1,\ldots,x_n}$ denote the environment, obtained by substituting $x_1',\ldots,x_n'$ simultaneously for all occurences of $x_1,\ldots,x_n$. Similarly $E^{P_1',\ldots,P_n'}_{P_1,\ldots,P_n}$, $St^{x_1',\ldots,x_n'}_{x_1,\ldots,x_n}$ and $St^{P_1',\ldots,P_n'}_{P_1,\ldots,P_n}$ are defined for statements St and procedure variables $P_1,\ldots,P_n$, $P_1',\ldots,P_n'$.

Then $\mathcal{M}_{stat}$ can be obtained from $\mathcal{M}_{dyn}$ by

- changing vi) into:

  $(\sigma, \sigma') \in \mathcal{M}_I(E_1|\underline{\text{begin new}}\ x;\ D\ E_2\ St\ \underline{\text{end}})(\gamma) \Leftrightarrow$
  There exist $d \in \underline{D}$ and $\sigma'' \in \underline{\Sigma}$ such that:
  $(\sigma\ [d/x'],\ \sigma'') \in \mathcal{M}_I\ (E_1{}^{x'}_{x}|\underline{\text{begin}}\ D\ E_2{}^{x'}_{x}\ \ St^{x'}_{x}\ \underline{\text{end}})(\gamma)$
  and $\sigma' = \sigma''\ [\sigma(x')/x']$

  where $x' \in \underline{V}$ does neither eccur in E_1, D, E_2 or St nor in any var $(\gamma(P))$, for which P occurs free in $(E_1|\underline{\text{begin}}\ E_2 St\ \underline{\text{end}})$.

- changing vii) into:

  $\mathcal{M}_I(E_1\ \underline{\text{begin proc}}\ P_1 : B_1;\ldots;\ \underline{\text{proc}}\ P_m : B_m;\ St\ \underline{\text{end}})(\gamma)$
  $= \mathcal{M}_I(E_1\ \underline{\text{proc}}\ P_1' : (B_1)^{P_1',\ldots,P_m'}_{P_1,\ldots,P_m}\ ;\ \ldots.|$
  $\underline{\text{begin}}\ \ St^{P_1',\ldots,P_m'}_{P_1,\ldots,P_m}\ \underline{\text{end}})\ (\gamma)$

  where $P_1',\ldots,P_m' \in \underline{PV}$ do not occur in E_1, $B_1,\ldots,B_m$ or St.

Lemma 2 also holds for $\mathcal{M}_{stat}$, but in order to see that $\mathcal{M}_{stat}$ is well-defined, it must be additionally proved that the new clause vi) is independent on the choice of x' and the new clause vii) of the choice of $P_1', \ldots, P_m'$.

Note that this is the first time, that - in the new clause vi) - we have used the fact, that $\gamma(P)$ is a program relation for every $P \in \underline{PV}$. In the definition of dynamic scope semantics we could have worked with arbitrary relations as well. But in order to allow a uniform tratment of both semantics, program relations were used for $\mathcal{M}_{dyn}$, too.

3.5. A program example

As our semantics definitions differ from usual definitions by a nondeterministic interpretation of variable declarations, we want to give a small example which illustrates this feature of our semantics.

Let $\underline{L}$ be the language of Peano-arithmetic and I its usual interpretation. Let $St \in \mathfrak{St}(\underline{L})$ be the statement:
begin new x; if x = 0 then call P; y := x else y := x+1 fi end
and let γ be a procedure assignment with:

$$\gamma(P) = \{(\sigma, \sigma[2/x]) \mid \sigma \in \underline{\Sigma}\},$$

i. e. $\gamma(P)$ assigns the value 2 to the variable x and does not change the value of any other variable. Note that $var(\gamma(P)) = \{x\}$.

With these notations we get for $\mathcal{M}_{dyn}$:

$(\sigma, \sigma') \in \mathcal{M}_I(\varepsilon \mid St)(\gamma)$

$\overset{\text{vi)} \Leftrightarrow \text{v)}}{\Leftrightarrow}$ There exist $d \in \underline{D}$ and $\sigma'' \in \underline{\Sigma}$ such that:

$(\sigma[d/x], \sigma'') \in \mathcal{M}_I(\varepsilon \mid \underline{\text{if}}\ x = 0\ \underline{\text{then call}}\ P;\ y := x + 1\ \underline{\text{fi}})(\gamma)$

and $\sigma' = \sigma''[\sigma(x)/x]$

iii) There exists $\sigma'' \in \Sigma$ such that:

$(\sigma[0/x], \sigma'') \in \mathcal{M}_I\ (\varepsilon|\underline{\text{call}}\ P;\ y := x)\ (\gamma)$

or $(\sigma[d/x], \sigma'') \in \mathcal{M}_I(\varepsilon|y := x + 1)\ (\gamma)$ for some $d \neq 0$

and $\sigma' = \sigma''\ [\sigma(x)/x]$

ii),viii),i) $\sigma' = \sigma[0/x][2/x][2/y][\sigma(x)/x]$

or $\sigma' = \sigma[d/x][d+1/y][\sigma(x)/x]$ for some $d \neq 0$

$\leftrightarrow$ $\sigma' = \sigma[d/y]$ for some $d \geq 2$

and for $\mathcal{M}_{stat}$:

$(\sigma, \sigma') \in \mathcal{M}_I(\varepsilon|St)\ (\gamma)$

vi), v) There exist $d \in \underline{D}$ and $\sigma'' \in \Sigma$ such that:

$(\sigma[d/x'], \sigma'') \in \mathcal{M}_I(\varepsilon|\underline{\text{if}}\ x' = 0\ \underline{\text{then}}\ \underline{\text{call}}\ P;\ y := x'$

$\underline{\text{else}}\ y := x' + 1\ \underline{\text{fi}})(\gamma)$

and $\sigma' = \sigma''[\sigma(x')/x']$

iii) There exists $\sigma'' \in \Sigma$ such that:

$(\sigma[0/x'], \sigma'') \in \mathcal{M}_I\ (\varepsilon|\underline{\text{call}}\ P;\ y := x')(\gamma)$

or $(\sigma[d/x'], \sigma'') \in \mathcal{M}_I(\varepsilon|y := x' + 1)(\gamma)$ for some $d \neq 0$

and $\sigma' = \sigma''\ [\sigma(x')/x']$

ii),viii),i) $\sigma' = \sigma[0/x'][2/x][0/y][\sigma(x')/x']$

or $\sigma' = \sigma[d/x'][d + 1/y][\sigma(x')/x']$ for some $d \neq 0$

$\leftrightarrow$ $\sigma' = \sigma[2/x][0/y]$

or $\sigma' = \sigma[d/y]$ for some $d \geq 2$

4. Hoare phrases and strong expressiveness

4.1. Definitions

As was already mentioned in section 2.5. we shall use so-called Hoare phrases in our axiom systems. They will be introduced now:

Assume $\underline{L}_E$ and $\underline{L}_A$ to be given as in section 2.4.

A *generalized Hoare formula* (*gHf*) for $\underline{L}_A$ and $\mathcal{U}(\underline{L}_E)$ is

- either a formula of $\underline{L}_A$
- or a formula $\{p\}\ (E|St)\{q\}$ where $p, q \in \underline{L}_A$, $(E|St) \in \mathcal{U}(\underline{L}_E)$. Instead of $\{p\}\ (\varepsilon|St)\ \{q\}$ we write $\{p\}\ St\ \{q\}$; in this sense every Hoare formula for $\underline{L}_A$ and $\mathcal{S}_{proc}\ (\underline{L}_E)$ is a special kind of a gHf.

A *Hoare phrase* for $\underline{L}_A$ and $\mathcal{U}(\underline{L}_E)$ is a construct of the form $G \to H$, where $G = \langle g_1,\ldots,g_m\rangle$ is a (possibly empty) sequence of gHf's, each g_i being of the form $\{p_i\}$ call $P_i\{q_i\}$, and $H = \langle h_1,\ldots,h_n\rangle$ is a nonempty sequence of arbitrary gHf's. We write h instead of $\langle h\rangle$, and H instead of $\langle\rangle \to H$. In this sense every gHf and hence every Hoare formula is a special kind of a Hoare phrase. Sequences of gHf's will be also considered as sets and in this sense the symbols "$\subseteq$", "$\cup$",... will be used.

Let I be an interpretation of $\underline{L}_A$ and $\underline{\Sigma}$ its set of states. Let $\mathcal{R}_E$ denote the set of all program relations $\underline{R} \subseteq \underline{\Sigma} \times \underline{\Sigma}$ with var $(\underline{R}) \subseteq \underline{V}_E$ and $\underline{\Gamma}_E$ the set of all procedure assignments γ: $\underline{PV} \to \mathcal{R}$ with $\gamma(P) \in \mathcal{R}_E$ for every $P \in \underline{PV}$. Then for every $(E|St) \in \mathcal{U}(\underline{L}_E) \subseteq \mathcal{U}(\underline{L}_A)$ and $\gamma \in \underline{\Gamma}_E \subseteq \underline{\Gamma}$ the relation $\mathcal{M}_I(E|St)(\gamma)$ is defined and a truth value $I(h)(\gamma, \sigma)$ is assigned to every gHf h by:

a) If $h = p \in \underline{L}_A$, then $I(p)(\gamma, \sigma) = I(p)(\sigma)$

b) If $h = \{p\}\ (E|St)\{q\}$ then

$$I(\{p\}\ (E|St)\ \{q\})(\gamma, \sigma) = \underline{true}$$

$$\Leftrightarrow \begin{cases} \text{If } I(p)(\sigma) = \underline{true} \text{ and } (\sigma, \sigma') \in \mathcal{M}_I(E|St)(\gamma) \\ \qquad \text{then } I(q)(\sigma') = \underline{true} \end{cases}$$

Instead of $I(h)(\gamma, \sigma) = \underline{\text{true}}$, we also write $\models_{I,\gamma,\sigma}{}^{h}$, $\models_{I,\gamma}{}^{h}$ stands for: $\models_{I,\gamma,\sigma}{}^{h}$ for all $\sigma \in \underline{\Sigma}$, and for a sequence H of gHf's $\models_{I,\gamma} H$ means: $\models_{I,\gamma}{}^{h}$ for every $h \in H$.

A Hoare phrase is called *I-valid* (notation: $\models_I G \to H$) if for every $\gamma \in \underline{\Gamma}_E$: $\models_{I,\gamma} G$ implies $\models_{I,\gamma} H$, and *valid* (notation: $\models G \to H$) if $\models_I G \to H$ for every interpretation I for $\underline{L}_A$. Of course the definitions depend on the semantics $\mathcal{M}$. Note that for Hoare formulas (which are Hoare phrases of the special form $<> \to \{p\}\ S\ \{q\}$, where S is a program) these definitions coincide with those of section 2.4., because $\mathcal{M}_I(S) = \mathcal{M}_I(\varepsilon|S)(\gamma)$ for every γ, in both semantics.

With the new definitions, strong expressiveness can now be defined:

Let $\underline{L}_E$, $\underline{L}_A$ and I be as usual.
Then $\underline{L}_A$ is called *strongly I-expressive* with respect to $\mathcal{U}(\underline{L}_E)$ and $\mathcal{M}$ if:

- $\underline{L}_E$ contains the equality symbol, which is interpreted by I as equality on $\underline{D}$ and
- for every $p \in \underline{L}_A$, $(E|St) \in \mathcal{U}(\underline{L}_E)$ and list G of gHf's (as in a Hoare phrase), the strongest postcondition of p and the relation $\bigcup_{\gamma \in \Gamma_E, \vdash_{I,\gamma} G} \mathcal{M}_I(E|St)(\gamma)$ is contained in $\underline{L}_A$.

An axiom system AX for $\mathcal{P}_{proc}$ (working with Hoare phrases) is said to be *relatively complete in the new sense* with respect to $\mathcal{M}$ if for every $\underline{L}_E$, $\underline{L}_A$ and interpretation I of $\underline{L}_A$, such that $\underline{L}_A$ is strongly I-expressive with respect to $\mathcal{U}(\underline{L}_E)$ and $\mathcal{M}$:

If $\models_I G \to H$ for some Hoare phrase, then $\vdash_I G \to H$.

4.2. The role played by algebraic variables

Having defined our programming language in section 3 and the semantics of Hoare phrases in section 4.1., we are now ready to discuss, which role is played by algebraic variables:

Let P be a recursive procedure, declared by proc P : B. In order to prove partial correctness of P w. r. t. some formulas p and q, we have to apply Scott's induction principle, i. e. we have to prove an assertion of the form:

" If $\gamma(P') \subseteq R$, then $\mathcal{M}_I(\varepsilon | B_p^{P'})(\gamma) \subseteq R$ ",

where R is a suitable relation and P' is a "dummy procedure variable". Such an assertion must be translated into Hoare logic by:

$$\{r\}\ \underline{call}\ P'\{s\} \rightarrow \{r\}\ B_p^{P'}\ \{s\}$$

where {r}... {s} expresses the relation R.

Let e. g. P be a procedure which calculates the factorial function, then r will be : x = a and s : x = a!, where $a \in \underline{V}_A \setminus \underline{V}_E$ and, as a typical step in the derivation of such a Hoare phrase, e. g.

$$\{x = a\}\ \underline{call}\ P'\ \{x = a!\} \rightarrow \{x = a - 1\}\ \underline{call}\ P'\ \{x = (a - 1)!\}$$

will occur. This Hoare phrase is I-valid, as we have defined the semantics of Hoare phrases by quantifying over $\gamma \in \underline{\Gamma}_E$:

Let $\models_{I,\gamma} \{x=a\}\ \underline{call}\ P'\{x=a!\}$ for some $\gamma \in \underline{\Gamma}_E$ and let $(\sigma, \sigma') \in \gamma(P')$.
Then, as $a \notin var(\gamma(P'))$, also $(\sigma[\sigma(x)/a], \sigma'[\sigma(x)/a]) \in \gamma(P')$.

As $I(x=a)(\sigma[\sigma(x)/a]) = \underline{true}$, the assumption implies:

$I(x=a!)(\sigma'[\sigma(x)/a]) = \underline{true}$, i. e. $\sigma'(x) = \sigma(x)!$

As this holds for every $(\sigma, \sigma') \in \gamma(P')$, one gets $\models_{I,\gamma} \{x=a-1\}\ \underline{call}\ P'\{x=(a-1)!\}$

Note that, if we had used a program variable y instead of the algebraic variable a, the corresponding Hoare phrase would not be I-valid, as e. g. if

$$\gamma(P') = \{(\sigma, \sigma[2/x][2/y]) \mid \sigma \in \Sigma\} \quad \text{for some } \gamma \in \underline{\Gamma}_E,$$

then $\models_{I,\gamma} \{x = y\}\ \underline{call}\ P'\ \{x = y!\}$ holds,
but $\models_{I,\gamma} \{x = y - 1\}\ \underline{call}\ P'\ \{x = (y - 1)!\}$ does not.

This is the reason, why in [APT 78] or [OLD], where no algebraic variables are used, the syntax and semantics of Hoare phrases is more complicated:

Both authors add the declaration proc P : B to the Hoare phrase and - roughly spoken - their interpretation of Hoare phrases differs from ours as follows:

a) Apt quantifies over all $\gamma \in \Gamma$, for which var $(\gamma(P'))$ contains only variables occuring in B. Hence
{x=y}call P'{x=y!}→<proc P:B|{x=y-1}call P'{x=(y-1)!}>
is I-valid in Apt's sense, if y does not occur in B.

b) Olderog quantifies over all $\gamma \in \Gamma$, for which $\gamma(P')$ is an "approximation" of the meaning of P, i. e. in the above example: $\gamma(P') = (\Phi^{E,\gamma})^n (\emptyset)$ with E = proc P : B;
and $n \in IN$.

As for these approximations, var $(\gamma(P'))$ contains only variables occuring in B, all Hoare phrases which are I-valid in Apt's sense, are also I-valid in Olderog's sense (but not vice versa).

Our concept, based on algebraic variables, seems to be more natural than a) and b) and - moreover - has another advantage:
Let e. g. St be a statement, calculating $\binom{m}{n} = \frac{m!}{n!(m-n)!}$
with a free procedure variable P, which is assumed to calculate the factorial function. Then we are interested in assertions like: "If P calculates the factorial function, then St calculates $\binom{m}{n}$, which can be written as a Hoare phrase by:

$$\{x = a\}\ \underline{call}\ P\ \{x = a!\} \rightarrow \{x = a \wedge y = b\}\ St\ \{z = \binom{a}{b}\} \quad (*)$$

Again this Hoare phrase will not be I-valid, if we replace a by a program variable y.

Moreover we are interested in an inference rule, which allows us to derive e. g.

$$\{x = a\}\ (\underline{proc}\ P : B \mid St)\quad \{z = \binom{a}{b}\}$$

from (*) and {x = a} (proc P : B | call P) {x = a!},
because such a derivation is closely related to the way of thinking in structured programming. While in [APT 78] or [OLD]

such a rule does not exist and also cannot be introduced without difficulties , our axiom systems contain such rules (see (D 11) in section 5.1. resp. (S 11) in section 7.).

4.3. A property of strongest postconditions

Lemma 3:

For both semantics, $\mathcal{M}_{dyn}$ and $\mathcal{M}_{stat}$, the following assertion holds:

If $\underline{F} \subseteq \underline{V}$ contains all variables of E and St, then
$\mathcal{M}_I\ (E|St)\ (\gamma)^{\underline{F}} \subseteq \mathcal{M}_I\ (E|St)\ (\gamma^{\underline{F}})$ for every $\gamma \in \underline{\Gamma}$.

The proof (again by structural induction) is left to the reader.

Lemma 4:

If q is the strongest postcondition of p and $\bigcup_{\gamma \in \underline{\Gamma}_E, \models_{I,\gamma} G} \mathcal{M}_I(E|St)(\gamma)$ then:

a) $\models_I G \to \{p\}\ (E|St)\ \{r\}$ iff $\models_I q \supset r$,
in particular $\models_I G \to \{p\}\ (E|St)\ \{q\}$ holds.

b) For both semantics, $\mathcal{M}_{dyn}$ and $\mathcal{M}_{stat}$, q depends only on the variables of G, p, E and St.

Proof:

a) is easy and is left to the reader.

b) Let $\underline{F}$ be the set of variables of G, p, E and St and let $\sigma'|_{\underline{F}} = \tau'|_{\underline{F}}$.

Then it must be proved that

$$I(q)(\sigma') = \underline{\text{true}} \quad \leftrightarrow \quad I(q)(\tau') = \underline{\text{true}}$$

Because of the symmetry of this assertion it is sufficient to prove one direction of the equivalence:

Let $I(q)(\sigma') = \underline{\text{true}}$.

Then there exist $\gamma \in \underline{\Gamma}_E$ and $\sigma \in \underline{\Sigma}$ such that

$\models_{I,\gamma} G$, $I(p)(\sigma) = \underline{true}$ and $(\sigma, \sigma') \in \mathcal{M}_I(E|St)(\gamma)$.

Let $\tau \in \underline{\Sigma}$ be defined by: $\tau|_{\underline{F}} = \sigma|_{\underline{F}}$ and $\tau|_{\underline{V}_A \setminus \underline{F}} = \tau'|_{\underline{V}_A \setminus \underline{F}}$.

Then $(\tau, \tau') \in \mathcal{M}_I(E|St)(\gamma)^{\underline{F}}$ and, as $\underline{F}$ contains all variables of E and St, lemma 3 implies:

$(\tau, \tau') \in \mathcal{M}_I(E|St)(\gamma^{\underline{F}})$.

As $\models_{I,\gamma} G$ holds, and $\underline{F}$ contains all variables of G,

$\models_{I,\gamma^{\underline{F}}} G$ holds too.

This implies, by part a) of the lemma: $\models_{I,\gamma^{\underline{F}}} \{p\}(E|St)\ \{q\}$

Finally, as $\underline{F}$ contains all variables of p:

$I(p)(\tau) = I(p)(\sigma) = \underline{true}$, hence $I(q)(\tau') = \underline{true}$ □

5. An axiom system for dynamic scope semantics

5.1. The axiom system

Here we present the axiom system AXDYN for $\mathcal{S}_{proc}$, which in the sequel will be proved sound and relatively complete in the new sense with respect to $\mathcal{M}_{dyn}$:

AXDYN:

(D0) $\{q^t_x\}\ (E|x := t)\ \{q\}$

(D1) $\dfrac{G \to h}{G' \to h}$ if $G \subseteq G'$

(D2) $\dfrac{G \to <\{p\}(E|St_1)\ \{q\},\ \{q\}\ (E|St_2)\ \{r\}>}{G \to \{p\}(E|St_1;\ St_2)\ \{r\}}$

(D3) $\dfrac{G \to <\{p \wedge e\}\ (E|St_1)\ \{q\},\ \{p \wedge \neg e\}\ (E|St_2)\ \{q\}>}{G \to \{p\}\ (E\ |\ \underline{if}\ e\ \underline{then}\ St_1\ \underline{else}\ St_2\ \underline{fi})\ \{q\}}$

(D4) $\dfrac{G \to \{p \wedge e\}\ (E|St)\ \{p\}}{G \to \{p\}\ (E\ |\ \underline{while}\ e\ \underline{do}\ St\ \underline{od})\ \{p \wedge \neg e\}}$

(D5) $\dfrac{G \to <p \supset q,\ \{q\}\ (E|St)\ \{r\},\ r \supset s>}{G \to \{p\}\ (E|St)\ \{s\}}$

(D6) $\dfrac{G \to \{p\}\ (E|St)\ \{q\}}{G \to \{p\}\ (E|\ \underline{begin}\ St\ \underline{end})\{q\}}$

(D7) $\dfrac{G \to \{p^a_x\}\ (E_1|\ \underline{begin}\ DE_2St\ \underline{end})\ \{q^a_x\}}{G \to \{p\}(E_1|\underline{begin}\ \underline{new}\ x;\ DE_2St\ \underline{end})\{q\}}$

if $a \in \underline{V}_A \setminus \underline{V}_E$ and does neither occur in p nor in q.

(D8) $\dfrac{G \to \{p\}(E_1\ [B/P]\ |\underline{begin}\ E_2St\ \underline{end})\ \{q\}}{G \to \{p\}(E_1|\ \underline{begin}\ \underline{proc}\ P : B;\ E_2St\ \underline{end})\ \{q\}}$

(D9) $\dfrac{\forall \bar{a}.\ (r_{1\bar{x}}^{\bar{y}} \supset s_1 \wedge \ldots \wedge r_{n\bar{x}}^{\bar{y}} \supset s_n) \supset (p_{\bar{x}}^{\bar{y}} \supset q)}{<\{r_1\}\ \underline{call}\ P\ \{s_1\},\ldots,\{r_n\}\ \underline{call}\ P\ \{s_n\}> \to \{p\}\ (E|\underline{call}\ P)\ \{q\}}$

if $n \geq 0$, $P \notin dom(E)$ and:

a) $\bar{x}$ is the vector of all program variables of p, $r_1,\ldots,r_n$.

b) $\bar{y}$ is a vector of variables, which do not occur in p, q, $r_1,\ldots r_n$, $s_1,\ldots, s_n$, of the same length as $\bar{x}$.

c) $\bar{a}$ is the vector of all algebraic variables of $r_1,\ldots,r_n$, $s_1,\ldots, s_n$.

(D10) $$\frac{G \cup \langle \{p_1\}\ \underline{\text{call}}\ P'_1\{q_1\},\ldots,\{p_m\}\ \underline{\text{call}}\ P'_m\ \{q_m\}\rangle \rightarrow \langle \{p_1\}\ (B_1)^{P'_1,\ldots,P'_m}_{P_1,\ldots,P_m}\ \{q_1\},\ldots,\ \{p_m\}(B_m)^{P'_1,\ldots,P'_m}_{P_1,\ldots,P_m}\ \{q_m\}\rangle}{G \rightarrow \{p_i\}(\underline{\text{proc}}\ P_1{:}B_1;\ldots;\ \underline{\text{proc}}\ P_m{:}B_m;\ |\underline{\text{call}}\ P_i)\ \{q_i\}}$$

for every i = 1, ..., m,
where $P'_1,\ldots,P'_m$ are different procedure varibales, which don't occur in G.

(D11) $$\frac{G \rightarrow \langle \{r_1\}(E_1|\underline{\text{call}}\ P_1)\{s_1\},\ldots,\{r_n\}(E_1|\underline{\text{call}}\ P_n)\{s_n\}\rangle,\ \langle \{r_1\}\ \underline{\text{call}}\ P_1\{s_1\},\ldots,\{r_n\}\ \underline{\text{call}}\ P_n\{s_n\}\rangle \rightarrow \{p\}(E_2|St)\ \{q\}}{G \rightarrow \{p\}\ (E_1E_2|St)\ \{q\}}$$

if no procedure variable, which occurs in E_1, occurs bound in $(E_2|St)$.

(D12) $$\frac{G \rightarrow h_1,\ldots,\ G \rightarrow h_n}{G \rightarrow\ \langle h_1,\ldots,\ h_n\rangle}$$

The axiom scheme (D0) and the inference rules (D2) - (D5) are slight variations of the usual Hoare rules for while-programs and (D1), (D6), (D8), (D12) are trivial. The other rules are shortly explained now:

(D7): The value of x neither has an influence on the execution of the block-statement "begin new x; DESt end" (as x is assigned a random value at the beginning), nor does it change (as x is assigned its old value again at the and), i. e. x behaves like an algebraic variable.

(D9): This rule is intuitively clear, if one thinks of the new varibales $\bar{y}$ as denotations of the values of $\bar{x}$ before the procedure call. (D9) is the basic rule for arguing about procedures.

(D10): This rule expresses Scott's induction principle.

(D11): This rule may be called the "principle of structured programming": One deduces $\{p\}\ (E_2|St)\ \{q\}$ under the assumption that $P_1,\ldots,P_n$ (which may occur free in $(E_2|St)$) have certain properties. Then a new environment E_1 is added, in which $P_1,\ldots,P_n$ have indeed these properties, and as a result one receives $\{p\}(E_1\ E_2|St)\{q\}$. The restriction, that no procedure variable, which occurs in E_1, occurs bound in $(E_2|St)$, is necessary because of dynamic scope.

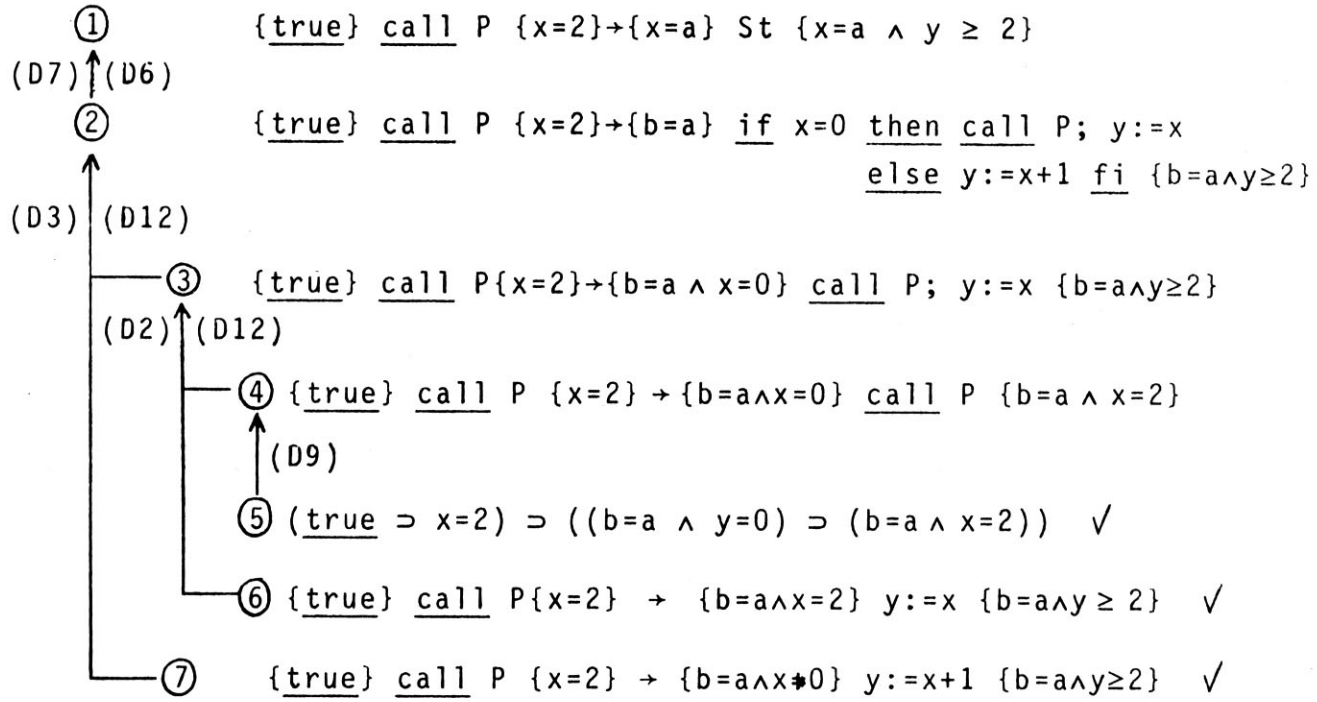
Note that (D11) is not only useful, but necessary, because Scott's induction principle, i. e. (D10) does not succeed for arbitrary assertions p_i, q_i, hence one needs (D11) afterwards (see section 5.3.).

5.2. Example of a derivation

Let L, I and St $\in \mathfrak{St}$(L) be as in section 3.5.
Then - according to the example in 3.5. - it must be possible to derive in AXDYN the Hoare phrase

$$\{\underline{\text{true}}\}\ \ \underline{\text{call}}\ P\ \{x = 2\} \rightarrow \{x = a\}\ St\ \{x = a \wedge y \geq 2\}$$

We describe the most important part of this derivation:



⑤ is an I-valid formula of $\underline{L}$ and the derivations of ⑥ and ⑦ are standard.

5.3. Soundness and relative completeness of AXDYN

Theorem 1:

AXDYN is sound and relatively complete in the new sense with respect to $\mathcal{M}_{dyn}$.

Proof:

We must prove, that for every interpretation I of $\underline{L}_A$:

a) $\vdash_I G \to H$ implies $\models_I G \to H$

and for every interpretation I of $\underline{L}_A$, such that $\underline{L}_A$ is strongly I-expressive with respect to $\mathcal{S}_{proc}$ ($\underline{L}_E$) and $\mathcal{M}_{dyn}$:

b) $\models_I G \to H$ implies $\vdash_I G \to H$

For this purpose the meaning of each inference rule, resp. of the axiom scheme is now described by a proposition. The proofs of the propositions are given in section 6.

Proposition D0:

$\models_I \; p \supset q_x^t \quad \Leftrightarrow \models_I \{p\}(E|x := t)\{q\}$,

especially every instace of (D0) is valid (as $\models q_x^t \supset q_x^t$).

Proposition D1:

$\models_I G \to h$ implies $\models_I G' \to h$.

Proposition D2:

a) $\models_I G \to \langle\{p\}(E|St_1)\{q\}, \{q\}(E|St_2)\{r\}\rangle$

implies $\models_I G \to \{p\}(E|St_1; St_2)\{r\}$

b) In the case of strong expressiveness one gets:

If $\models_I G \to \{p\}(E\ St_1; St_2)\{q\}$, then there exists $r \in \underline{L}_A$ such that:

$\models_I G \to \{p\}(E|St_1)\{r\}$ and $\models_I G \to \{r\}(E|St_2)\{q\}$

Proposition D3:

$\models_I G \to \{p \wedge e\}(E|St_1)\{q\}$ and $\models_I G \to \{p \wedge \neg e\}(E|St_2)\{q\}$

$\Leftrightarrow \models_I G \to \{p\}(E|\underline{\text{if}}\ e\ \underline{\text{then}}\ St_1\ \underline{\text{else}}\ St_2\ \underline{\text{fi}})\{q\}$

Proposition D4:

a) $\models_I G \to \{p \wedge e\}(E|St)\{p\}$

implies $\models_I G \to \{p\}(E|\underline{\text{while}}\ e\ \underline{\text{do}}\ St\ \underline{\text{od}})\{p \wedge \neg e\}$

b) In the case of strong expressiveness one gets:

If $\models_I G \to \{p\}(E|\underline{\text{while}}\ e\ \underline{\text{do}}\ St\ \underline{\text{od}})\{q\}$, then there exists $r \in \underline{L}_A$ sucht that:

$\models_I p \supset r$, $\models_I G \to \{r \wedge e\}(E|St)\{r\}$ and $\models_I (r \wedge \neg e) \supset q$

Proposition D5:

$\models_I$ G → <p ⊃ q, {q} (E|St) {r}, r ⊃ s>

implies $\models_I$ G → {p}(E|St) {s}

Proposition D6:

$\models_I$ G → {p}(E|St){q}

$\Leftrightarrow \models_I$ G → {p}(E|<u>begin</u> St <u>end</u>) {q}

Proposition D7:

$\models_I$ G → $\{p_x^a\}$ (E_1|<u>begin</u> DE_2St <u>end</u>) $\{q_x^a\}$ for some a ∈ $\underline{V}_A \setminus \underline{V}_E$,

which does not occur in p or q ⇔

$\models_I$ G → {p}(E_1|<u>begin</u> <u>new</u> x; DE_2St <u>end</u>) {q}

Proposition D8:

$\models_I$ G → {p} (E_1[B/P]| <u>begin</u> E_2St <u>end</u>) {q}

$\Leftrightarrow \models_I$ G → {p} (E_1| <u>begin</u> <u>proc</u> P : B; E_2St <u>end</u>) {q}

Proposition D9:

$\models_I$ $\forall \bar{a}.\ (r_1\frac{\bar{y}}{\bar{x}} \supset s_1 \wedge \ldots \wedge r_n\frac{\bar{y}}{\bar{x}} \supset s_n) \supset (p\frac{\bar{y}}{\bar{x}} \supset q)$ ⇔

$\models_I$ <{r_1} <u>call</u> P {s_1}, ..., {r_n} <u>call</u> P {s_n}>↣{p}(E|<u>call</u> P) {q}

Proposition D10:

a) $\models_I$ G ∪ {p_1} <u>call</u> P'$_1${q_1},...> → < {p_1}$(B_1)^{P'_1,\ldots,P'_m}_{P_1,\ldots,P_m}$ {q_1},...>implies

$\models_I$ G → {p_i} (<u>proc</u> P_1 : B_1;...; <u>proc</u> P_m :B_m;|<u>call</u> P_i) {q_i}

b) If $\bar{x}$ contains all program variables of G, $B_1,\ldots,B_m$, $\bar{a}$ is a vector of algebraic variables of the same length and r_i is the strongest postcondition of $\bar{x} = \bar{a}$ and

$$\bigcup_{\gamma \in \Gamma_E, \models_{I,\gamma} G} \mathcal{M}_I\,(\underline{\text{proc}}\ P_1 : B_1; \ldots; \underline{\text{proc}}\ P_m : B_m; \mid \underline{\text{call}}\ P_i)(\gamma)$$
$$\text{for } i = 1, \ldots, m$$

(r_i exists in the case of strong expressiveness)

then one gets:

$$\models_I G \cup \langle \{\bar{x} = \bar{a}\}\ \underline{\text{call}}\ P'_1\{r_1\}, \ldots \rangle \rightarrow \langle \{\bar{x} = \bar{a}\}(B_1)^{P'_1, \ldots, P'_m}_{P_1, \ldots, P_m}\ \{r_1\}, \ldots \rangle$$

c) If additionally

$$\models_I G \rightarrow \{p\}(\underline{\text{proc}}\ P_1 : B_1; \ldots; \underline{\text{proc}}\ P_m : B_m; \mid \underline{\text{call}}\ P_i)\ \{q\}$$
for some $i \in \{1, \ldots, m\}$,

then one gets:

$$\models_I \{\bar{x} = \bar{a}\}\ \underline{\text{call}}\ P_i\ \{r_i\} \rightarrow \{p\}\ \underline{\text{call}}\ P_i\ \{q\}$$

Proposition D11:

If $\models_I G \rightarrow \langle \{r_1\}(E_1 \mid \underline{\text{call}}\ P_1)\{s_1\}, \ldots, \{r_n\}(E_1 \mid \underline{\text{call}}\ P_n)\{s_n\} \rangle$

and $\models_I \langle \{r_1\}\ \underline{\text{call}}\ P_1\{s_1\}, \ldots, \{r_n\}\underline{\text{call}}\ P_n\{s_n\} \rangle \rightarrow \{p\}(E_2 \mid St)\{q\}$

then $\models_I G \rightarrow \{p\}(E_1\ E_2 \mid St)\ \{q\}$

Proposition D12:

(coincides with the definition of $\models_I G \rightarrow H$)

$\models_I G \rightarrow h_1, \ldots,\ _I G \rightarrow h_n \quad \Leftrightarrow \quad \models_I G \rightarrow \langle h_1, \ldots, h_n \rangle$

Now theorem 1 will be proved with the aid of the propositions:

a) Every instance of the axiom scheme (D0) is valid, and every inference rule, when applied to I-valid premises, yields an I-valid conclusion (see the propositions). Now by a standard induction argument on the length of the derivation of $\vdash_I G \rightarrow H$ one receives:

$\vdash_I G \rightarrow H$ implies $\models_I G \rightarrow H$

b) Because of (D12) and the definition of $\models_I G \to H$, it is sufficient to prove that in the case of strong expressiveness:

$\models_I G \to h$ implies $\vdash_I G \to h$ for every gHf h.

For this purpose, the set of gHf's is made to a wellfounded set by defining:

$$h_1 \prec h_2 \overset{\text{def.}}{\Leftrightarrow} h_1 \in \underline{L}_A \text{ and } h_2 = \{p\}(E|St)\{q\}$$

$$\text{or: } h_1 = \{p_1\}(E_1|St_1)\{q_1\},\ h_2 = \{p_2\}(E_2|St_2)\{q_2\}$$
$$\text{and } (E_1|St_1) \prec (E_2|St_2)$$

Now the assertion can be proved by structural induction:

$h = p \in \underline{L}_A$:

$\models_I G \to p$ means $\models_I p$, which by definition is equivalent to $\vdash_I p$ or $\vdash_I \langle\,\rangle \to p$. Now one gets by (D1):

$$\vdash_I G \to p$$

$h = \{p\}(E|St)\{q\}$:

As an example we consider the case St = call P, which is the most difficult. The other cases are left to the reader.

Assume $\models_I G \to \{p\}(E|\underline{\text{call}}\ P)\{q\}$.

(1^o) If $P \notin \text{dom}(E)$, then it is sufficient to prove

$$\left.\begin{array}{l} \models_I G' \to \{p\}(E|\underline{\text{call}}\ P)\{q\} \\ \text{where } G' = \langle \{r_1\}\underline{\text{call}}\ P\{s_1\},\ldots,\{r_n\}\ \underline{\text{call}}\ P\ \{s_n\}\rangle \\ \text{is the set of all gHf's of G, which contain } \underline{\text{call}}\ P. \end{array}\right\} \quad (*)$$

From (*) one gets by proposition D9:

$$\models_I \forall \bar{a}.\ (r_1 \tfrac{\bar{y}}{\bar{x}} \supset s_1 \wedge \ldots \wedge r_n \tfrac{\bar{y}}{\bar{x}} \supset s_n) \supset (p \tfrac{\bar{y}}{\bar{x}} \supset q)$$

which is equivalent to:

$\vdash_I \quad \forall \bar{a}.\ (r_1 \frac{\bar{y}}{\bar{x}} \supset s_1 \wedge \ldots \wedge r_n \frac{\bar{y}}{\bar{x}} \supset s_n) \supset (p \frac{\bar{y}}{\bar{x}} \supset q)$

Then (D9) yields:

$\vdash_I \quad G' \to \{p\}(E|\underline{\text{call}}\ P)\{q\}$

and (D1):

$\vdash_I G \to \{p\}(E|\underline{\text{call}}\ P)\ \{q\}$

Proof of (*):

If $\models_{I,\gamma} G'$ for some $\gamma \in \Gamma_E$,
then let γ' be defined by:

$\gamma'(P) = \gamma(P)$ and $\gamma'(Q) = \emptyset$ for every $Q \neq P$

Then $\models_{I,\gamma'} G$ holds, which implies $\models_{I,\gamma'} \{p\}(E|\underline{\text{call}}P)\{q\}$.
Hence also $\models_{I,\gamma} \{p\}(E|\underline{\text{call}}\ P)\{q\}$, and (*) is proved.

(2^o) If = = $\underline{\text{proc}}\ P_1 : B_1; \ldots; \underline{\text{proc}}\ P_m : B_m$; and $P = P_i$,
then let $\bar{x}$ be the vector of all program variables of p, q,
G and $B_1, \ldots, B_m$ and let $\bar{a}$, r_j and P'_j ($j = 1, \ldots, m$) as defined in proposition D 10 b) resp. in rule (D 10).

Then proposition D 10 b) yields:

$\models_I G \cup \langle\{\bar{x}=\bar{a}\}\ \underline{\text{call}}\ P'_1\{r_1\},\ldots\rangle \to \langle\{\bar{x}=\bar{a}\}(B_1)^{P'_1,\ldots,P'_m}_{P_1,\ldots,P_m}\ \{r_1\},\ldots\rangle$

As $(\varepsilon|(B_j)^{P'_1\ \ldots,P'_m}_{P_1,\ldots,P_m}) \prec (E|\underline{\text{call}}\ P)$ for every $j = 1,\ldots, m$,

one gets by the induction hypothesis and by (D12):

$\vdash_I G\ \ \langle\{\bar{x}=\bar{a}\}\underline{\text{call}}\ P_1'\{r_1\},\ldots\rangle \to \langle\{\bar{x}=\bar{a}\}(B_1)^{P'_1,\ldots,P'_m}_{P_1,\ldots,P_m}\ \{r_1\},\ldots\rangle$

and (D10) yields:

$\vdash_I G \to \{\bar{x} = \bar{a}\}\ (E|\underline{\text{call}}\ P_i)\ \{r_i\}$

Furtheron by proposition D 10 c) one gets:

$\vdash_I \{\bar{x} = \bar{a}\}\ \underline{\text{call}}\ P_i\{r_i\} \rightarrow \{p\}\ \underline{\text{call}}\ P_i\ \{q\}$

wich by (1^o) implies:

$\vdash_I \{\bar{x} = \bar{a}\}\ \underline{\text{call}}\ P_i\{r_i\} \rightarrow \{p\}\ \underline{\text{call}}\ P_i\ \{q\}$

Finally by (D11)

$\vdash_I G \rightarrow \{p\}\ (E|\underline{\text{call}}\ P_i)\ \{q\}$

can be derived. □

This concludes the proof of theorem 1 up to the proofs of the particular propositions, which are given in the next section.

6. Proofs of the propositions of theorem 1

The proofs of the propositions D0, D1, D2a), D3, D4a) D5, D6, D8 and D12 are easy and are left to the reader. Some of them are given in [SIE] for while-programs without procedures.

Proof of proposition D2b):

Assume $\models_I G \to \{p\}(E|St_1; St_2)\{q\}$.

Let r be the strongest postcondition of p and

$$\bigcup_{\gamma \in \underline{\Gamma}_E, \models_{I,\gamma} G} \mathcal{M}_I(E|St_1)(\gamma)$$

(r exists because of strong expressiveness)

Then, by lemma 4a) one gets $\models_I G \to \{p\}(E|St_1)\{r\}$ and only $\models_I G \to \{r\}(E|St_2)\{q\}$ remains to be proved:

Let $\gamma_1 \in \underline{\Gamma}_E$ such that $\models_{I,\gamma_1} G$, $I(r)(\sigma) = \underline{true}$ and $(\sigma, \sigma') \in \mathcal{M}_I(E|St_2)(\gamma_1)$.

Then $I(q)(\sigma') = \underline{true}$ must be proved.
Because of the definition of r, there exist $\gamma_2 \in \underline{\Gamma}_E$ and $\tau \in \underline{\Sigma}$ such that $\models_{I,\gamma_2} G$ and $(\tau, \sigma) \in \mathcal{M}_I(E|St_1)(\gamma_2)$.

Let now $\gamma = \gamma_1 \sqcup \gamma_2 \in \underline{\Gamma}_E$. Then $\models_{I,\gamma} G$ holds, and because of the monotonicity of each $\mathcal{M}_I(E|St)$ one gets:
$(\tau, \sigma) \in \mathcal{M}_I(E|St_1)(\gamma)$ and $(\sigma, \sigma') \in \mathcal{M}_I(E|St_2)(\gamma)$,
hence $(\tau, \sigma') \in \mathcal{M}_I(E|St_1; St_2)(\gamma)$.

Finally, as $I(p)(\tau) = \underline{true}$,
the assumption implies $I(q)(\sigma') = \underline{true}$. □

Proof of proposition D4b):

Assume $\models_I G \rightarrow \{p\}(E|\underline{\text{while}}\ e\ \underline{\text{do}}\ St\ \underline{\text{od}})\ \{q\}$.

Let $\bar{y} = (y_1,\ldots,y_n)$ be the vector of all program variables, occuring in G, p, E, e, St and q, and $\bar{z} = (z_1,\ldots,z_n)$ a vector of programming variables, different from $y_1,\ldots,y_n$.

Let s be the strongest postcondition of p and

$$\bigcup_{\gamma \in \underline{\Gamma}_E, \models_{I,\gamma} G} \mathcal{M}_I(E|\underline{\text{while}}\ e \wedge \neg \bar{y} = \bar{z}\ \underline{\text{do}}\ St\ \underline{\text{od}})\ (\gamma)$$

(which exists because of strong expressiveness)
and let r be the formula $\exists\, \bar{z}.s$.

(1^o) Let $I(p)(\sigma) = \underline{\text{true}}$.
It must be proved that $I(r)(\sigma) = \underline{\text{true}}$,

i. e. $I(s)(\sigma[\bar{d}/\bar{z}]) = \underline{\text{true}}$ for some $\bar{d} \in \underline{D}^n$.

Let $\bar{d} = \sigma(\bar{y})$. Then $I(p)(\sigma[\bar{d}/\bar{z}]) = I(p)(\sigma) = \underline{\text{true}}$ and

$$(\sigma[\bar{d}/\bar{z}],\ \sigma[\bar{d}/\bar{z}]) \in \bigcup_{\gamma \in \underline{\Gamma}_E, \models_{I,\gamma} G} \mathcal{M}_I(E|\underline{\text{while}}\ e \wedge \neg \bar{y}=\bar{z}\ \underline{\text{do}}\ St\ \underline{\text{od}})(\gamma)$$

Hence, by the definition of s : $I(s)(\sigma[\bar{d}/\bar{z}]) = \underline{\text{true}}$.

(2^o) Let $\gamma_1 \in \underline{\Gamma}_E$ such that $\models_{I,\gamma_1} G$, $I(r \wedge e)(\sigma) = \underline{\text{true}}$ and
$(\sigma,\ \sigma') \in \mathcal{M}_I(E|St)(\gamma_1)$.

It must be proved, that $I(r)(\sigma') = \underline{\text{true}}$.

As $I(r)(\sigma) = \underline{\text{true}}$, there exists $\bar{d} \in \underline{D}^n$ such that

$$I(s)(\sigma[\bar{d}/\bar{z}]) = \underline{\text{true}}$$

Then, because of the definition of s, there exist $\gamma_2 \in \underline{\Gamma}_E$ and $\tau \in \underline{\Sigma}$ such that $\models_{I,\gamma_2} G$, $I(p)(\tau) = \underline{\text{true}}$ and

$$(\tau,\ \sigma[\bar{d}/\bar{z}]) \in \mathcal{M}_I\ (E|\underline{\text{while}}\ e \wedge \neg \bar{y} = \bar{z}\ \underline{\text{do}}\ St\ \underline{\text{od}})(\gamma_2)$$

i. e. there exist $\tau_1,\ldots,\tau_m \in \underline{\Sigma}$ such that:

$\tau_1 = \tau$, $\tau_m = \sigma[\bar{d}/\bar{z}]$,

$(\tau_k, \tau_{k+1}) \in \mathcal{M}_I(E|St)(\gamma_2)$ for $k = 1, \ldots, m-1$,

$I(e \wedge \neg\, \bar{y} = \bar{z})(\tau_k) = \underline{true}$ for $k = 1, \ldots, m-1$ and

$I(e \wedge \neg\, \bar{y} = \bar{z})(\tau_m) = \underline{false}$.

Now let $\underline{F} = \{y_1, \ldots, y_n\}$ and $\rho_1, \ldots, \rho_m \in \underline{\Sigma}$ such that:

$\rho_k|_{\underline{F}} = \tau_k|_{\underline{F}}$ for $k = 1, \ldots, m$ (especially: $\rho_m|_{\underline{F}} = \sigma|_{\underline{F}}$),

$\rho_{m+1}|_{\underline{F}} = \sigma'|_{\underline{F}}$,

$\rho_k(\bar{z}) = \sigma'(\bar{y})$ for $k = 1, \ldots, m+1$

and $\rho_1, \ldots, \rho_{m+1}$ coincide on all other variables.

Then one gets:

$(\rho_k, \rho_{k+1}) \in \mathcal{M}_I(E|St)(\gamma_2)^{\underline{F}} \subseteq \mathcal{M}_I(E|St)(\gamma_2^{\underline{F}})$ for $k=1, \ldots, m-1$

and $(\rho_m, \rho_{m+1}) \in \mathcal{M}_I(E|St)(\gamma_1)^{\underline{F}} \subseteq \mathcal{M}_I(E|St)(\gamma_1^{\underline{F}})$.

Define $\gamma = \gamma_1^{\underline{F}} \sqcup \gamma_2^{\underline{F}} \in \underline{\Gamma}_E$.

Then $(\rho_k, \rho_{k+1}) \in \mathcal{M}_I(E|St)(\gamma)$ for $k = 1, \ldots, m$.

As $\underline{F}$ contains all variables of e,

$I(e)(\rho_k) = I(e)(\tau_k) = \underline{true}$ for $k = 1, \ldots, m-1$

and also

$I(e)(\rho_m) = I(e)(\sigma) = \underline{true}$

Now let $l \in \mathbb{N}$ be the least number with $\rho_l(\bar{y}) = \rho_l(\bar{z})$

($l \leq m+1$ exists, because $\rho_{m+1}(\bar{y}) = \sigma'(\bar{y}) = \rho_{m+1}(\bar{z})$).

Then $(\rho_1, \rho_l) \in \mathcal{M}_I(E|\underline{while}\ e \wedge \neg\, \bar{y}=\bar{z}\ \underline{do}\ St\ \underline{od})(\gamma)$.

As $\underline{F}$ contains all variables of p, $I(p)(\rho_1) = I(p)(\tau_1) = \underline{true}$,
and as it contains all variables of G; $\models_{I,\gamma} G$ holds.

Hence $I(s)(\rho_l) = \underline{true}$ (by definition of s).

But $\rho_l(\bar{y}) = \rho_l(\bar{z}) = \sigma'(\bar{y}) = \rho_{m+1}(\bar{z}) = \rho_{m+1}(\bar{y})$ and, moreover,
ρ_l and ρ_{m+1} coincide on all other variables, i. e.
$\rho_l = \rho_{m+1}$, $I(s)(\rho_{m+1}) = \underline{true}$ and finally $I(r)(\rho_{m+1}) = \underline{true}$.

Now lemma 4b) yields, that s depends only on the variables of $\underline{F} \cup \{z_1,\ldots,z_n\}$, hence r depends only on the variables of $\underline{F}$. As ρ_{m+1} and σ' coincide on $\underline{F}$, one gets

$I(r)(\sigma') = \underline{true}$.

(3^0) Let $I(r \wedge \neg e)(\sigma) = \underline{true}$.
It must be proved that $I(q)(\sigma) = \underline{true}$.

As $I(r)(\sigma) = \underline{true}$, $I(s)(\sigma[\bar{d}/\bar{z}]) = \underline{true}$ for some $\bar{d} \in \underline{D}^n$.
Because of the definition of s there exist $\gamma \in \underline{\Gamma}_E$, $\tau \in \underline{\Sigma}$
such that $\models_{I,\gamma} G$, $I(p)(\tau) = \underline{true}$ and

$(\tau, \sigma[\bar{d}/\bar{z}]) \in \mathcal{M}_I(E|\underline{while}\ e \wedge \neg\, \bar{y}=\bar{z}\ \underline{do}\ St\ \underline{od})(\gamma)$.

As $I(\neg e)(\sigma[\bar{d}/\bar{z}]) = I(\neg e)(\sigma) = \underline{true}$, even

$(\tau, \sigma[\bar{d}/\bar{z}]) \in \mathcal{M}_I(E|\underline{while}\ e\ \underline{do}\ St\ \underline{od})(\gamma)$ holds.

But then the assumption implies $I(q)(\sigma[\bar{d}/\bar{z}]) = \underline{true}$,
hence $I(q)(\sigma) = \underline{true}$. □

Proof of proposition D 7:

It is sufficient to prove for every $\gamma \in \underline{\Gamma}_E$ that:

$\models_{I,\gamma} \{p_x^a\}\ (E_1|\underline{begin}\ D\ E_2\ St\ \underline{end})\ \{q_x^a\}$

for some $a \in \underline{V}_A \setminus \underline{V}_E$ which does not occur in p or q

$\Leftrightarrow \models_{I,\gamma} \{p\}(E_1|\underline{begin}\ \underline{new}\ x;\ DE_2St\ \underline{end})\ \{q\}$

"$\Rightarrow$": Assume $\models_{I,\gamma} \{p_x^a\}(E_1|\underline{begin}\ DE_2St\ \underline{end})\{q_x^a\}$,

$I(p)(\sigma) = \underline{true}$ and $(\sigma, \sigma') \in \mathcal{M}_I(E_1|\underline{begin}\ \underline{new}\ x;\ DE_2St\ \underline{end})(\gamma)$.

Then there exist $d \in \underline{D}$ and $\sigma'' \in \underline{\Sigma}$ such that:

$(\sigma[d/x], \sigma'') \in \mathcal{M}_I(E_1|\underline{begin}\ DE_2St\ \underline{end})(\gamma)$ and $\sigma' = \sigma''[\sigma(x)/x]$

As a is an algebraic variable and $\gamma \in \underline{\Gamma}_E$, also

$(\underbrace{\sigma[d/x][\sigma(x)/a]}_{\tau},\ \underbrace{\sigma''\ [\sigma(x)/a]}_{\tau'}) \in \mathcal{M}_I(E_1|\underline{begin}\ DE_2St\ \underline{end})(\gamma)$

By the substitution lemma

$$I(p^a_x)(\tau) = I(p)(\tau[\tau(a)/x]) = I(p)(\tau\ [\sigma(x)/x])$$
$$= I(p)(\sigma[\sigma(x)/a]) \underset{(*)}{=} I(p)(\sigma) = \underline{\text{true}}$$

where (*) holds, because a does not occur in p.

The assumption implies $I(q^a_x)(\tau') = \underline{\text{true}}$

and another application of the substitution lemma yields

$$I(q)(\sigma') = I(q)(\sigma''\ [\sigma(x)/x]) \underset{(*)}{=} I(q)(\sigma''[\sigma(x)/a]\ [\sigma(x)/x])$$
$$= I(q)(\tau'[\sigma(x)/x]) = I(q)(\tau'\ [\tau'(a)/x])$$
$$= I(q^a_x)(\tau') = \underline{\text{true}}$$

where (*) holds, because a does not occur in q.

"⇐": Assume $\models_{I,\gamma} \{p\}(E_1 | \underline{\text{begin}}\ \underline{\text{new}}\ x;\ DE_2 St\ \underline{\text{end}})\{q\}$,

$I(p^a_x)(\sigma) = \underline{\text{true}}$ and $(\sigma, \sigma') \in \mathcal{M}_I(E_1 | \underline{\text{begin}}\ DE_2 St\ \underline{\text{end}})(\gamma)$.

Let $\tau = \sigma[\sigma(a)/x]$ and $\tau' = \sigma'\ [\sigma(a)/x]$

Then $(\tau, \tau') \in \mathcal{M}_I(E_1 | \underline{\text{begin}}\ \underline{\text{new}}\ x;\ DE_2 St\ \underline{\text{end}})(\gamma)$,

because $\sigma = \tau[\sigma(x)/x]$ and $\tau' = \sigma'\ [\tau(x)/x]$.

Now by the substitution lemma

$$I(p)(\tau) = I(p)(\sigma\ [\sigma(a)/x]) = I(p^a_x)(\sigma) = \underline{\text{true}}$$

and the assumption implies

$I(q)(\tau') = \underline{\text{true}}$.

As a is an algebraic varibale, $\sigma(a) = \sigma'(a)$ and another application of the substitution lemma yields

$$I(q^a_x)(\sigma') = I(q)(\sigma'[\sigma'(a)/x]) = I(q)(\sigma'[\sigma(a)/x])$$
$$= I(q)(\tau') = \underline{\text{true}}.$$

□

Proof of proposition D 9:

"→": Assume $\models_I \forall \bar{a}.(r_1{}^{\bar{y}}_{\bar{x}} \supset s_1 \wedge \ldots \wedge r_n{}^{\bar{y}}_{\bar{x}} \supset s_n) \supset (p^{\bar{y}}_{\bar{x}} \supset q)$

and $\models_{I,\gamma} \{r_i\}\underline{\text{call}}\ P\{s_i\}$ for $i = 1,\ldots, n$ for some $\gamma \in \underline{\Gamma}_E$.

Then $\models_{I,\gamma} \{p\}(E|\underline{\text{call}}\ P)\{q\}$ must be proved.

Let $I(p)(\sigma) = \underline{\text{true}}$ and $(\sigma, \sigma') \in \mathcal{M}_I(E|\underline{\text{call}}\ P)(\gamma) = \gamma(P)$.

Then we get

$I(p^{\bar{y}}_{\bar{x}})(\sigma'[\sigma(\bar{x})/\bar{y}])$

$= I(p^{\bar{y}}_{\bar{x}})(\sigma\ [\sigma(\bar{x})/\bar{y}])$

(as the two states coincide on all algebraic variables and on the variables of $\bar{y}$)

$= I(p)(\sigma[\sigma(\bar{x})/\bar{y}]\ [\sigma(\bar{x})/\bar{x}])$

(by the substitution lemma)

$= I(p)(\sigma)$

(as the variables of $\bar{y}$ do not occur in p)

Hence $I(p^{\bar{y}}_{\bar{x}})(\sigma'[\sigma(\bar{x})/\bar{y}]) = \underline{\text{true}}$. (1)

Similarly, we get for every $i = 1,\ldots, n$ and vector $\bar{d}$ of elements of $\underline{D}$:

$I(r_i{}^{\bar{y}}_{\bar{x}})(\sigma'\ [\sigma(\bar{x})/\bar{y}][\bar{d}/\bar{a}]) = I(r_i)(\sigma[\bar{d}/\bar{a}])$ (2)

and

$I(s_i)(\sigma'\ [\sigma(\bar{x})/\bar{y}][\bar{d}/\bar{a}]) = I(s_i)(\sigma'[\bar{d}/\bar{a}])$. (3)

Because of

$\models_{I,\gamma} \{r_i\}\ \underline{\text{call}}\ P\ \{s_i\}$ for $i = 1,\ldots, n$ and

$(\sigma[\bar{d}/\bar{a}], \sigma'[\bar{d}/\bar{a}]) \in \gamma(P)$ (as $\gamma \in \underline{\Gamma}_E$)

(2) and (3) yield:

$I(\forall \bar{a}.(r_1{}^{\bar{y}}_{\bar{x}} \supset s_1 \wedge \ldots \wedge r_n{}^{\bar{y}}_{\bar{x}} \supset s_n))(\sigma'[\sigma(\bar{x})/\bar{y}]) = \underline{\text{true}}$ (4)

Finally (1), (4) and the assumption implie:

$I(q)(\sigma'\ [\sigma(\bar{x})/\bar{y}]) = \underline{true}$

i. e. $I(q)(\sigma') = \underline{true}$

(as no variable of $\bar{y}$ occurs in q)

"$\Leftarrow$" Let G be $\langle\{r_1\}\ \underline{call}\ P\ \{s_1\},\ldots,\{r_n\}\ \underline{call}\ P\ \{s_n\}\rangle$

and assume $\models_I\ G \rightarrow \{p\}\ (E|\underline{call}\ P)\ \{q\}$,

i. e. $\models_{I,\gamma} \{p\}\ (E|\underline{call}\ P)\{q\}$ for every $\gamma \in \underline{\Gamma}_E$ with $\models_{I,\gamma} G$.

Let $I(p\frac{\bar{y}}{\bar{x}})(\sigma) = \underline{true}$

and $I(\forall\ \bar{a}.\ (r_1\frac{\bar{y}}{\bar{x}} \supset s_1 \wedge\ldots\wedge r_n\frac{\bar{y}}{\bar{x}} \supset s_n))(\sigma) = true.$

Then, by the substitution lemma:

$$I(p)(\sigma\ [\sigma(\bar{y})/\bar{x}]) = \underline{true} \qquad (1)$$

and for every vector $\bar{d}$ of elements of $\underline{D}$ and every i:

$$I(r_i)(\sigma[\sigma(\bar{y})/\bar{x}][\bar{d}/\bar{a}]) = \underline{true} \text{ implies } I(s_i)(\sigma[\bar{d}/\bar{a}]) = \underline{true} \qquad (2)$$

Let now $\gamma \in \underline{\Gamma}_E$ be a procedure assignment with

$\gamma(P) = \{(\tau, \tau')|$ τ and τ' differ only in $\bar{x}$ and for every $\bar{d}$ and every i: $I(r_i)(\tau[\bar{d}/\bar{a}]) = \underline{true}$ implies $I(s_i)(\tau'[\bar{d}/\bar{a}]) = \underline{true}\}$

(note that this relation is indeed in $\underline{\mathfrak{R}}_E$.)

Then $\models_{I,\gamma}$ G holds, and the assumption implies:

$$\models_{I,\gamma} \{p\}(E|\underline{call}\ P)\ \{q\} \qquad (3)$$

Now (2) means: $(\sigma\ [\sigma(\bar{y})/\bar{x}],\ \sigma) \in \gamma\ (P)$,

hence (1) and (3) implie: $I(q)(\sigma) = \underline{true}$

and $\models_I\ \forall\ \bar{a}.\ (r_1\frac{\bar{y}}{\bar{x}} \supset s_1 \wedge\ldots\wedge r_n\frac{\bar{y}}{\bar{x}} \supset s_n) \supset (p\frac{\bar{y}}{\bar{x}} \supset q)$ is proved.

□

Proof of proposition D 10 a):

For the proof we need

Lemma 5:

For every $(E|St) \in \mathcal{U}(\underline{L})$, $P_1,\ldots,P_m,P_1',\ldots,P_m' \in \underline{PV}$ and $\gamma \in \underline{\Gamma}$:

$$\mathcal{M}_I(E^{P_1',\ldots,P_m'}_{P_1,\ldots,P_m} \mid St^{P_1',\ldots,P_m'}_{P_1,\ldots,P_m})(\gamma) = \mathcal{M}_I\ (E|St)(\gamma[\gamma(P_1')/P_1]\ldots[\gamma(P_m')/P_m])$$

Lemma 5 is a "substitution lemma" for procedure variables, and is intuitively clear.
Its proof is left to the reader.

Now proposition D 10 a) can be proved:

Assume $\models_{I,\gamma} \{p_1\}(B_1)^{P_1',\ldots,P_m'}_{P_1,\ldots,P_m} \{q_1\},\ldots, \models_{I,\gamma} \{p_m\}(B_m)^{P_1',\ldots,P_m'}_{P_1,\ldots,P_m} \{q_m\}$

for every $\gamma \in \underline{\Gamma}_E$ with $\models_{I,\gamma} G \cup \langle \{p_1\}\ \underline{call}\ P_1'\ \{q_1\},\ldots\rangle$

Then we must prove

$$\models_{I,\gamma} \{p_i\}\ (E|\underline{call}\ P_i)\ \{q_i\} \text{ for } i = 1,\ldots, m \qquad (*)$$
$$\text{for every } \gamma \in \underline{\Gamma}_E \text{ with } \models_{I,\gamma} G,$$

where $E = \underline{proc}\ P_1 : B_1; \ldots; \underline{proc}\ P_m : B_m;$

Let $\gamma \in \underline{\Gamma}_E$ be a procedure assignment with $\models_{I,\gamma} G$.
Then, by definition of the semantics,
$\mathcal{M}_I\ (E|\underline{call}\ P_i)(\gamma) = \mu_i^{E,\gamma}$ for every i,
where $(\mu_1^{E,\gamma}, \ldots, \mu_m^{E,\gamma})$ is the least fixpoint of

$$\Phi^{E,\gamma} : \underline{\mathfrak{R}}^m \longrightarrow \underline{\mathfrak{R}}^m$$

$$(\underline{R}_1,\ldots,\underline{R}_m) \longmapsto (\mathcal{M}_I(\varepsilon|B_1)(\gamma[\underline{R}_1/P_1]\ldots[\underline{R}_m/P_m]),$$
$$\vdots$$
$$\mathcal{M}_I(\varepsilon|B_m)(\gamma[\underline{R}_1/P_1]\ldots[\underline{R}_m/P_m]))$$

With these notations, (*) is equivalent to

$$\models_{I,\gamma[\mu_1^{E,\gamma}/P_1']\ldots[\mu_m^{E,\gamma}/P_m']} \{p_i\} \underline{\text{call}}\ P_i' \{q_i\} \text{ for } i=1,\ldots,m \quad (**)$$

This last assertion will now be proved with Scott's induction principle, applied to n-tuples $(\underline{R}_1,\ldots,\underline{R}_m) \in \underline{\mathfrak{R}}^m$ with the predicate:

$$\models_{I,\gamma[\underline{R}_1/P_1']\ldots[\underline{R}_m/P_m']} \{p_i\} \underline{\text{call}}\ P_i' \{q_i\} \text{ for } i=1,\ldots,m \quad (***)$$

(1^o) (***) holds for $(\underline{R}_1,\ldots,\underline{R}_m) = (\emptyset,\ldots,\emptyset)$

(2^o) Assume (***) holds for $(\underline{R}_1,\ldots,\underline{R}_m)$

Then it must be proved, that (***) also holds for

$$(\underline{R}_1',\ldots,R_m') = \Phi^{E,\gamma}(\underline{R}_1,\ldots,\underline{R}_m)$$

As $\models_{I,\gamma} G$ and $P_1',\ldots,P_m'$ dot not occur in G, the induction hypothesis yields

$$\models_{I,\gamma[\underline{R}_1/P_1']\ldots[\underline{R}_m/P_m']} G \cup \langle\{p_1\} \underline{\text{call}}\ P_1' \{q_1\},\ldots\rangle$$

Now the assumption implies

$$\models_{I,\gamma[\underline{R}_1/P_1']\ldots[\underline{R}_m/P_m']} \{p_i\}(B_i)^{P_1',\ldots,P_m'}_{P_1',\ldots,P_m'} \{q_i\} \text{ for } i=1,\ldots,m,$$

and as by lemma 5

$$\mathcal{M}_I(\varepsilon|(B_i)^{P_1',\ldots,P_m'}_{P_1',\ldots,P_m'})(\gamma[\underline{R}_1/P_1']\ldots[\underline{R}_m/P_m'])$$

$= \mathcal{M}_I(\varepsilon|B_i)(\gamma[\underline{R}_1/P_1'] \ldots [\underline{R}_m/P_m'][\underline{R}_1/P_1] \ldots [\underline{R}_m/P_m])$

$= \mathcal{M}_I(\varepsilon|B_i)(\gamma[\underline{R}_1/P_1] \ldots [\underline{R}_m/P_m]) = \underline{R}_i'$

this is equivalent to

$\models_{I,\gamma[\underline{R}_1'/P_1'] \ldots [\underline{R}_m'/P_m']} \{p_i\}$ call P_i' $\{q_i\}$.

(3^o) As (***) is an admissible predicate (in the sense of [MAN]), (1^o) and (2^o) implie (**). □

Proof of proposition D 10 b):

Lemma 6:

a) Let $\underline{F} \subseteq \underline{V}_A$ be a set, which contains all program variables of p and q, and let $\underline{R}_1$, $\underline{R}_2 \subseteq \Sigma \times \Sigma$ be $\underline{V}_E$ - bounded relations (not necessarily program relations).

If $\underline{R}_1$ is partially correct with respect to p and q and $\underline{R}_2^F \subseteq \underline{R}_I^F$, then $\underline{R}_2$ is partially correct with respect to p and q, too.

b) Let $\bar{x} = (x_1, \ldots, x_n)$ be a vector of program variables, $\bar{a} = (a_1, \ldots, a_n)$ a vector of algebraic variables, and $\underline{R}_1$, $\underline{R}_2$ $\underline{V}_E$ - bounded relations.

If $\underline{F} = \{x_1, \ldots, x_n\}$, q is the strongest postcondition of $\bar{x} = \bar{a}$ and $\underline{R}_1$, and $\underline{R}_2$ is partially correct with respect to $\bar{x} = \bar{a}$ and q, then $\underline{R}_2^F \subseteq \underline{R}_I^F$.

Proof:

a) Let $(\sigma, \sigma') \in \underline{R}_2$ and $I(p)(\sigma) =$ true.

It must be proved that $I(q)(\sigma') =$ true

Let $\sigma'' \in \underline{\Sigma}$ be defined by: $\sigma''|_{\underline{F}} = \sigma'|_{\underline{F}}$, $\sigma''|_{\underline{V}_A \setminus \underline{F}} = \sigma|_{\underline{V}_A \setminus \underline{F}}$

Then $(\sigma, \sigma'') \in \underline{R}_2^{\underline{F}} \subseteq \underline{R}_1^{\underline{F}}$, i. e. there exists $(\tau, \tau') \in \underline{R}_1$ such that $\sigma|_{\underline{F}} = \tau|_{\underline{F}}$, $\sigma''|_{\underline{F}} = \tau'|_{\underline{F}}$.

Moreover, as $\underline{R}_2$ is $\underline{V}_E$ - bounded, σ, σ' and σ'' coincide on $\underline{V}_A \setminus \underline{V}_E$, and as $\underline{R}_1$ is $\underline{V}_E$ - bounded, one can choose (τ, τ') such that σ, σ'', τ and τ' coincide on $\underline{V}_A \setminus \underline{V}_E$, too.

Hence $I(p)(\tau) = I(p)(\sigma) = \underline{true}$ and the partial correctness of $\underline{R}_1$ with respect to p and q implies: $I(q)(\tau') = \underline{true}$.

But then also $I(q)(\sigma') = I(q)(\sigma'') = I(q)(\tau') = \underline{true}$.

b) Let $(\tau, \tau') \in \underline{R}_2^{\underline{F}}$.

Then there exists $(\sigma, \sigma') \in \underline{R}_2$ such that $\sigma(\bar{x}) = \tau(\bar{x})$, $\sigma'(\bar{x}) = \tau'(\bar{x})$ and $\tau|_{\underline{V}_A \setminus \underline{F}} = \tau'|_{\underline{V}_A \setminus \underline{F}}$.

Define $\rho = \sigma[\sigma(\bar{x})/\bar{a}]$, $\rho' = \sigma'[\sigma(\bar{x})/\bar{a}]$.
As $\underline{R}_2$ is $\underline{V}_E$ - bounded, also $(\rho, \rho') \in \underline{R}_2$.
Now $I(\bar{x} = \bar{a})(\rho) = \underline{true}$ yields $I(q)(\rho') = \underline{true}$,
i. e. there exists $\rho'' \in \underline{\Sigma}$ such that
$I(\bar{x} = \bar{a})(\rho'') = \underline{true}$ and $(\rho'', \rho') \in \underline{R}_1$.

As $\tau(\bar{x}) = \sigma(\bar{x}) = \rho(\bar{a}) = \rho'(\bar{a}) = \rho''(\bar{a}) = \rho''(\bar{x})$
and $\tau'(\bar{x}) = \sigma'(\bar{x}) = \rho'(\bar{x})$,
$(\tau, \tau') \in \underline{R}_2^{\underline{F}}$ too. □

Now the proposition can be proved:

Let $\bar{x} = (x_1, \ldots, x_n)$, $\bar{a}$ and $r_1, \ldots, r_m$ be as defined in the proposition, $E = \underline{proc}\ P_1 : B_1\ ; \ldots ;\ \underline{proc}\ P_m : B_m$ and $\gamma_1 \in \underline{\Gamma}_E$ such that

$\vdash_{I,\gamma_1} G$ and $\vDash_{I,\gamma_1} \{\bar{x}=\bar{a}\}\ \underline{call}\ P_i'\ \{r_i\}$ for $i = 1, \ldots, m$.

Then $\vDash_{I,\gamma_1} \{\bar{x} = \bar{a}\}(B_i)^{P_1', \ldots, P_m'}_{P_1, \ldots, P_m} \{r_i\}$ must be proved for $i = 1, \ldots, m$.

For $\underline{F} = \{x_1,\ldots,x_n\}$ one first gets by lemma 6 b):

$$\gamma_1(P_i')^{\underline{F}} \;\underline{c}\; \Big(\bigcup_{\gamma\in\underline{\Gamma}_E,\models_{I,\gamma}G} \mathcal{M}_I(E|\underline{\text{call}}\ P_i)(\gamma)\Big)^{\underline{F}}$$

$$= \bigcup_{\gamma\in\underline{\Gamma}_E,\models_{I,\gamma}G} \mathcal{M}_I(E|\underline{\text{call}}\ P_i)(\gamma)^{\underline{F}} \qquad \text{for } i=1,\ldots,m$$

and by lemma 6 a) it is sufficient to prove:

$$\mathcal{M}_I(\varepsilon|B_i)^{P_1',\ldots,P_m'}_{P_1,\ldots,P_m}(\gamma_1)^{\underline{F}} \;\underline{c}\; \Big(\bigcup_{\gamma\in\underline{\Gamma}_E,\models_{I,\gamma}G} \mathcal{M}_I(E|\underline{\text{call}}\ P_i)(\gamma)\Big)^{\underline{F}}$$

for $i=1,\ldots,m$.

Hence let $0 \le i \le m$ and denote $(B_i)^{P_1',\ldots,P_m'}_{P_1,\ldots,P_m}$ by B_i'.

Then

$$\mathcal{M}_I(\varepsilon \mid B_i')(\gamma_1)^{\underline{F}}$$

$$\underline{c}\quad \mathcal{M}_I(\varepsilon \mid B_i')(\gamma_1^{\underline{F}}) \qquad \text{(by lemma 3)}$$

$$\underline{c}\quad \mathcal{M}_I(\varepsilon \mid B_i')(\gamma_1^{\underline{F}}\,[\bigcup_{\gamma\in\underline{\Gamma}_E,\models_{I,\gamma}G} \mathcal{M}_I(E|\underline{\text{call}}\ P_1)(\gamma)^{\underline{F}}/P_1'] \ldots.)$$

(because of monotonicity of $\mathcal{M}_I(\varepsilon|B_i')$ and as

$$\gamma_I^{\underline{F}}(P_i') \;\underline{c}\; \bigcup_{\gamma\in\underline{\Gamma}_E,\models_{I,\gamma}G} \mathcal{M}_I(E|\underline{\text{call}}\ P_i)(\gamma)^{\underline{F}} \qquad \text{for } i=1,\ldots,m)$$

$$\underline{c}\ \mathcal{M}_I(\varepsilon|B_i')\Big(\Big(\bigsqcap_{\gamma\in\underline{\Gamma}_E,\models_{I,\gamma}G} \gamma^{\underline{F}}\Big)\Big[\bigcup_{\gamma\in\underline{\Gamma}_E,\models_{I,\gamma}G} \mathcal{M}_I(E|\underline{\text{call}}\ P_1)(\gamma)^{\underline{F}}/P_1'\Big]\ldots\Big)$$

(again by monotonicity and as $\models_{I,\gamma_1} G$ holds)

$$= \bigcup_{\gamma\in\underline{\Gamma}_E,\models_{I,\gamma}G} \mathcal{M}_I(\varepsilon|B_i')(\gamma^{\underline{F}}\,[\mathcal{M}_I(E|\underline{\text{call}}\ P_1)(\gamma)^{\underline{F}}\ /\ P_1']\ldots)$$

(by Δ - continuity of $\mathcal{M}_I(\varepsilon|B_i')$)

$$\underline{c} \bigcup_{\gamma \in \underline{\Gamma}_E, \models_{I,\gamma} G} \mathcal{M}_I(\varepsilon|B_i')(\gamma^{\underline{F}}[\mathcal{M}_I(E|\underline{\text{call}}\ P_1)(\gamma^{\underline{F}})/P_1'] \ldots)$$

(by monoticity of $\mathcal{M}_I(\varepsilon|B_i')$ together with lemma 3)

$$= \bigcup_{\gamma \in \underline{\Gamma}_E, \models_{I,\gamma} G} \mathcal{M}_I(\varepsilon|B_i)(\gamma^{\underline{F}}\ [\mathcal{M}_I(E|\underline{\text{call}}\ P_1)(\gamma^{\underline{F}})/P_1] \ldots)$$

$$= \bigcup_{\gamma \in \underline{\Gamma}_E, \models_{I,\gamma} G} \mathcal{M}_I(E|\underline{\text{call}}\ P_i)(\gamma^{\underline{F}})$$

(as $(\mathcal{M}_I(E|\underline{\text{call}}\ P_1)(\gamma^{\underline{F}}), \ldots, \mathcal{M}_I(E|\underline{\text{call}}\ P_m)(\gamma^{\underline{F}}))$ is a fixpoint of $\Phi^{E,\gamma^{\underline{F}}}$)

$$\underline{c} \bigcup_{\gamma \in \underline{\Gamma}_E, \models_{I,\gamma} G} \mathcal{M}_I(E|\underline{\text{call}}\ P_i)(\gamma)$$

(as $\models_{I,\gamma} G$ implies $\models_{I,\gamma^{\underline{F}}} G$)

Hence we have proved

$$\mathcal{M}_I(E|B_i')(\gamma)^{\underline{F}} \ \underline{c} \bigcup_{\gamma \in \underline{\Gamma}_E, \models_{I,\gamma} G} \mathcal{M}_I(E|\underline{\text{call}}\ P_i)(\gamma)$$

But then even

$$\mathcal{M}_I(E|B_i')(\gamma)^{\underline{F}} \ \underline{c} \ (\bigcup_{\gamma \in \underline{\Gamma}_E, \models_{I,\gamma} G} \mathcal{M}_I(E|\underline{\text{call}}\ P_i)(\gamma))^{\underline{F}} \text{ holds.}$$

□

Proof of proposition D 10 c):

Let $\bar{x} = (x_1, \ldots, x_n)$, $\bar{a}$ and $r_1, \ldots, r_m$ as defined in the proposition and E = $\underline{\text{proc}}\ P_1 : B_1; \ \ldots \ ; \ \underline{\text{proc}}\ P_m : B_m$

Assume $\models_I \ G \rightarrow \{p\}\ (E|\underline{\text{call}}\ P_i)\ \{q\}$.

Then $\models_I \{\bar{x} = \bar{a}\}$ call P_i $\{r_i\} \rightarrow \{p\}$ call P_i $\{q\}$

must be proved.

Let $F = \{x_1,\ldots,x_n\}$ and $\models_{I,\gamma} \{\bar{x} = \bar{a}\}$ call P_i $\{r_i\}$

Then, as r_i is the strongest postcondition of $\bar{x} = \bar{a}$ and

$$\underline{R} = \bigcup_{\gamma \in \underline{\Gamma}_E, \models_{I,\gamma} G} \mathcal{M}_I(E|\text{call } P_i)(\gamma) ,$$

lemma 6 b) yields: $\gamma(P_i)^{\underline{F}} \subseteq \underline{R}^{\underline{F}}$

(note that $\underline{R}$ is $\underline{V}_E$ - bounded)

Finally, as by the assumption $\underline{R}$ is partially correct with respect to p and q, lemma 6 a) implies, that

$\gamma(P_i)$ is partially correct with respect to p and q, i. e.

$$\models_{I,\gamma} \{p\} \text{ call } P_i \{q\}$$ □

Proof of proposition D 11:

Lemma 7:

If no procedure variable, which occurs in $E_1 =$ proc $P_1 : B_1$; ...; proc $P_m : B_m$; occurs bound in $(E_2|St)$, then:

$$\mathcal{M}_I(E_1;E_2|St)(\gamma) = \mathcal{M}_I(E_2|St)(\gamma[\mu_1^{E_1,\gamma}/P_1]\ldots[\mu_m^{E_1,\gamma}/P_m])$$

for every $\gamma \in \underline{\Gamma}$

Proof:

The restriction guarantees that the meaning of $P_1,\ldots, P_m$ is not "changed" during the program (which is possible because of dynamic scope). Hence they can be assigned their meaning at once. A detailed proof is left to the reader. □

Proposition D 11 is an immediate consequence of this lemma.

This concludes the proof of theorem 1.

7. The axiom system AXSTAT

According to the modification of the semantics (compare section 3) an axiom system AXSTAT for the static scope semantics can be obtained from AXDYN by modifying only some of the inference rules:

(D7) is changed into (S7):

$$\frac{G \cup \langle \{x'=a\}\ \underline{call}\ P_1\ \{x'=a\}, \ldots, \{x'=a\}\ \underline{call}\ P_m\ \{x'=a\}\rangle \rightarrow \{p\}(E_{1x}^{x'} \mid \underline{begin}\ DE_{2x}^{x'}\ St_x^{x'}\ \underline{end})\ \{q\}}{G \rightarrow \{p\}(E_1 \mid \underline{begin}\ \underline{new}\ x;\ DE_2St\ \underline{end})\ \{q\}}$$

where x' does not occur in G, p, D, E_1, E_2, St or q, $a \in \underline{V}_A \setminus \underline{V}_E$ and $P_1, \ldots, P_m$ are all procedure variables, which occur free in $(E_1 \mid \underline{begin}\ E_2St\ \underline{end})$.

(D8) is changed into (S8):

$$\frac{G \rightarrow \{p\}(E_1;\ \underline{proc}\ P_1 : (B_1)_{P_1,\ldots,P_m}^{P'_1,\ldots,P'_m}; \ldots \mid \underline{begin}\ St_{P_1,\ldots,P_m}^{P'_1,\ldots,P'_m}\ \underline{end})\{q\}}{G \rightarrow \{p\}(E_1 \mid \underline{begin}\ \underline{proc}\ P_1 : B_1; \ldots;\ \underline{proc}\ P_m : B_m;\ St\ \underline{end})\ \{q\}}$$

(D11) is changed into (S11):

$$\frac{G \rightarrow \langle \{r_1\}(E_1 \mid \underline{call}\ P_1)\{s_1\}, \ldots, \{r_n\}(E_1 \mid \underline{call}\ P_n)\ \{s_n\}\rangle, \langle \{r_1\}\ \underline{call}\ P_1\ \{s_1\}, \ldots, \{r_n\}\ \underline{call}\ P_n\{s_n\}\rangle \rightarrow \{p\}(E_2 \mid St)\ \{q\}}{G \rightarrow \{p\}(E_1;\ E_{2Q_1,\ldots,Q_m}^{Q'_1,\ldots,Q'_m} \mid St_{Q_1,\ldots,Q_m}^{Q'_1,\ldots,Q'_m})\ \{q\}}$$

where dom $(E_2) = \{Q_1, \ldots, Q_m\}$ and $Q'_1, \ldots, Q'_m$ as usual.
(but without the restrictions of the old rule (D 11)).

For the proof of soundness and relative completeness, only the propositions D7, D8 and D11 must be changed, according to the rules, e. g.:

New proposition S7:

$\models_I G \cup \langle \{x'=a\}\underline{call}\ P_1\{x'=a\}, \ldots, \{x'=a\}\underline{call}\ P_m\{x'=a\}\rangle$

$\rightarrow \{p\}(E_{1x}^{x'} \mid \underline{begin}\ DE_{2x}^{x'}\ \underline{end})\ \{q\}$

holds for some x' which does not occur in G, p, D, E_1, E_2, St or q and some $a \in \underline{V}_A \setminus \underline{V}_E$

$\leftrightarrow \models_I G \rightarrow \{p\}(E_1 \mid \underline{begin}\ \underline{new}\ x;\ DE_2St\ \underline{end})\ \{q\}$

Similarly, propositions D8 and D11 are changed and with these modifications the proof of theorem 1 (section 5) can be used to prove

Theorem 2:
==========

AXSTAT is sound and relatively complete in the new sense with respect to $\mathcal{M}_{stat}$.

We now give the proof of proposition S 7, the only proposition, in which new ideas are needed. Note that $\models_I \{x' = a\}\ P_i\ \{x' = a\}$ does <u>not</u> express, that $x' \notin$ var $(\gamma(P))$, but nevertheless the rule is strong enough to get relative completeness of AXSTAT.

Proof of the new proposition S7:

"$\rightarrow$":

Assume $\models_I G \cup \langle \{x'=a\}\underline{call}\ P_1\{x'=a\}, \ldots, \{x'=a\}\underline{call}\ P_m\{x'=a\}\rangle$

$\rightarrow \{p\}(E_{1x}^{x'} \mid \underline{begin}\ DE_{2x}^{x'}\ St_x^{x'}\ \underline{end})\ \{q\}$

and let $\gamma \in \underline{\Gamma}_E$ with $\models_{I,\gamma} G$.

Then $\models_{I,\gamma}\{p\}(E_1|\ \underline{\text{begin}}\ \underline{\text{new}}\ x;\ DE_2St\ \underline{\text{end}})\{q\}$ must be proved.

Let $\underline{F}$ be the set of all variables of G, p, D, E_1, E_2, St and q.
Then $\models_{I,\gamma\underline{F}}$ G holds, and as $x' \notin \underline{F}$ also

$\models_{I,\gamma\underline{F}} \{x' = a\}\ \underline{\text{call}}\ P_i\ \{x' = a\}$ holds for i = 1,...,m,

and the assumption implies

$$\models_{I,\gamma\underline{F}} \{p\}(E_{1x}^{x'}|\underline{\text{begin}}\ DE_{2x}^{x'}\ St_x^{x'}\ \underline{\text{end}})\ \{q\}. \qquad (*)$$

Now let $I(p)(\sigma)$ = $\underline{\text{true}}$ and

$$(\sigma, \sigma') \in \mathcal{M}_I(E_1|\ \underline{\text{begin}}\ \underline{\text{new}}\ x;\ DE_2St\ \underline{\text{end}})(\gamma).$$

Choose $\tau, \tau' \in \underline{\Sigma}$ such that $\tau|_{\underline{F}}=\sigma|_{\underline{F}}$, $\tau'|_F=\sigma'|_{\underline{F}}$, $\tau|_{\underline{V_A\setminus F}} = \tau'|_{\underline{V_A\setminus F}}$.

Then $(\tau, \tau') \in \mathcal{M}_I(E_1|\underline{\text{begin}}\ \underline{\text{new}}\ x;\ DE_2St\ \underline{\text{end}})(\gamma)^{\underline{F}}$

$$\underline{c}\ \mathcal{M}_I(E_1|\underline{\text{begin}}\ \underline{\text{new}}\ x;\ DE_2St\ \underline{\text{end}})(\gamma^{\underline{F}})$$

As $x' \notin \underline{F}$, it does not occur in var $(\gamma^{\underline{F}}(P_i))$ for i = 1,..., m, and by the definition of the semantics there exist $d \in \underline{D}$ and $\tau'' \in \underline{\Sigma}$ such that

$$(\tau[d/x'],\ \tau'') \in \mathcal{M}_I(E_{1x}^{x'}|\underline{\text{begin}}\ DE_{2x}^{x'}St_x^{x'}\ \underline{\text{end}})(\gamma^{\underline{F}})$$

and $\tau' = \tau''[\tau(x')/x']$.

As $I(p)(\tau[d/x']) = I(p)(\tau) = I(p)(\sigma) = \underline{\text{true}}$,
(*) implies $I(q)(\tau'') = \underline{\text{true}}$.

But then also $I(q)(\sigma') = I(q)(\tau') = I(q)(\tau'') = \underline{\text{true}}$.

"$\Leftarrow$":

Assume $\models_I G \to \{p\}(E_1|\underline{\text{begin}}\ \underline{\text{new}}\ x;\ DE_2St\ \underline{\text{end}})\{q\}$

and let $\gamma \in \underline{\Gamma_E}$ sucht that

$$\models_{I,\gamma} G \cup \langle\{x'=a\}\underline{\text{call}}\ P_1\{x'=a\},\ldots,\{x'=a\}\underline{\text{call}}\ P_m\{x'=a\}\rangle$$

Then $\models_{I,\gamma}\{p\}(E_{1x}^{x'}|\underline{\text{begin}}\ DE_{2x}^{x'}\ St_x^{x'}\ \underline{\text{end}})\{q\}$ must be proved.

Let $\underline{F} = \underline{V}_A \setminus \{x'\}$.
Then, as x' does not occur in G, $\models_{I,\gamma^{\underline{F}}} G$ holds, and the assumption implies

$$\models_{I,\gamma^{\underline{F}}} \{p\}(E_1 | \underline{\text{begin}}\ \underline{\text{new}}\ x;\ DE_2 St\ \underline{\text{end}})\{q\} \qquad (*)$$

Let now $I(p)(\sigma) = \underline{\text{true}}$ and

$$(\sigma, \sigma') \in \mathcal{M}_I(E_{1x}^{x'} | \underline{\text{begin}}\ DE_{2x}^{x'}\ St_x^{x'}\ \underline{\text{end}})(\gamma)$$

Now $\models_{I,\gamma}\{x'=a\}\underline{\text{call}}\ P_i\{x'=a\}$ implies $\gamma(P_i)\ \underline{c}\ \gamma^{\underline{F}}(P_i)$

for i = 1, ..., m,

hence, as $P_1,\ldots,P_m$ are the only procedure variables, which occur free in $(E_{1x}^{x'} | \underline{\text{begin}}\ DE_{2x}^{x'}\ St_x^{x'}\ \underline{\text{end}})$, also

$$(\sigma, \sigma') \in \mathcal{M}_I(E_{1x}^{x'} | \underline{\text{begin}}\ DE_{2x}^{x'}\ St_x^{x'}\ \underline{\text{end}})(\gamma^{\underline{F}})$$

But then e. g.: $(\sigma, \sigma'[\sigma(x')/x']) \in \mathcal{M}_I(E_1 | \underline{\text{begin}}\ \underline{\text{new}}\ x;\ DE_2 St\ \underline{\text{end}})(\gamma^{\underline{F}})$
as $x' \notin \text{var}(\gamma^{\underline{F}}(P_i))$ for i = 1, ..., m.

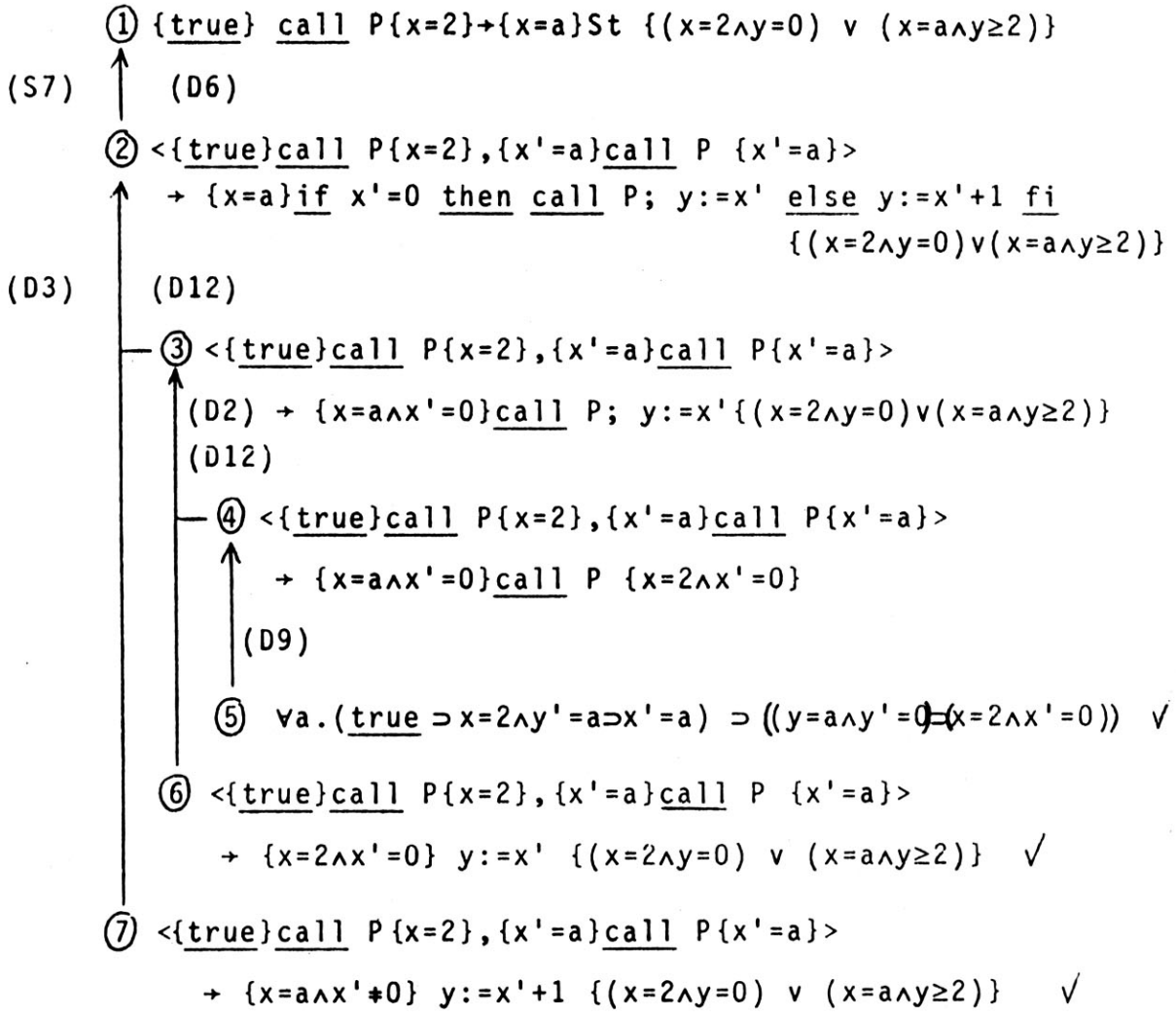
As $I(p) = \underline{\text{true}}$, (*) now implies $I(q)(\sigma[\sigma(x')/x']) = \underline{\text{true}}$
and finally $I(q)(\sigma') = \underline{\text{true}}$. □

Example of a derivation:

Let L, I and $St \in \mathcal{St}(\underline{L})$ be as in section 3.5.
Then - according to the static scope example in 3.5. - we prove in AXSTAT:

$\{\underline{\text{true}}\}\underline{\text{call}}\ P\ \{x=2\} \rightarrow \{x=a\}St\{(x=2 \wedge y=0) \vee (x=a \wedge y \geq 2)\}$

(See also the example for dynamic scope in section 5.2.)



⑤ is an I-valid formula of L and the derivations of ⑥ and ⑦ are standard.

8. Conclusion

As was already mentioned in section 2, our "strong expressiveness" is really stronger than Cook's expressiveness. Of corse, now the question arises, if "strong expressiveness" and hence "relative completeness in the new sense" is still a reasonable definition. A positive answer to this question can be found in [HAR]:

Harel proves, that for every pair ($\underline{L}$, I) which "contains the natural numbers", $\underline{L}$ is I-expressive with respect to an extension of his "context-free programs over $\underline{L}$". In our terms this means, that $\underline{L}$ is strongly I-expressive with respect to context-free programs over $\underline{L}$. This result can be transferred to our - slightly more complex - language $\mathcal{S}_{proc}(\underline{L})$. Hence our "relative completeness in the new sense" implies Harel's "arithmetical completeness" and, as Harel argues, citing a theorem of Lipton, this notion is not essentially weaker than Cook's relative completeness (only the finite interpretations are lost).

Acknoledgement: I am grateful to J. Loeckx for useful discussions concerning this paper.

References:

[ADJ] Goguen, Thatcher, Wagner and Wright, "Initial Algebra Semantics and Continuous Algebras", Journal of the ACM, vol. 24, no. 1, pp. 68 - 95 (1977).

[APT 78] Apt, K. R., "A Sound and Complete Hoare-like System for a fragment of PASCAL", Internal Report IW 96/78, Stichting Mathematisch Centrum, Amsterdam, 1978.

[APT 79] Apt, K. R., "Ten years of Hoare's logic, a survey", Faculty of Economics, Erasmus University, Rotterdam, April 1979.

[COOK] Cook, S. A., Soundness and Completeness of an Axiom System for Program Verification", SIAM Journal on Computing, vol. 7, no. 1, pp. 70 - 90 (1978).

[HAR] Harel, D., "First Order Dynamic Logik", Lecture Notes in Comp. Sc. 68 (1979).

[MAN] Manna, Z., "Mathematical Theory of Computation", Mc Graw-Hill, 1974.

[OLD] Olderog, E. R., "Sound and Complete Hoare Like Calculi Based on Copy Rules", Internal Report 7905, Christian-Albrechts-Universität, Kiel, 1979.

[SIE] Sieber, K., "Relative Completeness of a Hoare-Calculus for while-Programs", Internal Report A 80/01, Universität Saarbrücken, 1980.