A FORMALIZED PROOF SYSTEM FOR TOTAL CORRECTNESS OF WHILE PROGRAMS

J.A. Bergstra
Department of Computer Science
University of Leiden
Wassenaarseweg 80
2333 AL Leiden

J.W. Klop
Department of Computer Science
Mathematical Centre
Kruislaan 413
1098 SJ Amsterdam

ABSTRACT

We introduce datatype specifications based on schemes, a slight generalization of first order specifications. For a schematic specification $(\Sigma, \mathbb{E})$, Hoare's Logic $HL(\Sigma, \mathbb{E})$ for partial correctness is defined as usual and on top of it a proof system $(\Sigma, \mathbb{E}) \vdash p \rightarrow S \downarrow$ for termination assertions is defined. The system is first order in nature, but we prove it sound and complete w.r.t. a second order semantics. We provide a translation of a standard proof system $HL_T(A)$ for total correctness on a structure A into our format.

0. INTRODUCTION

In this note we will present a formalized proof system for total correctness of while-programs. Its merits should be first of all that it acts as a first order proof system (although we can, at this moment, only prove a soundness result w.r.t. a second order semantics which allows fewer models for a specification than the usual first order semantics would do). The advantage of having a formalized proof system $(\Sigma, \mathbb{E}) \vdash p \rightarrow S \downarrow$ for program termination which is just as first order as Hoare's logic $HL(\Sigma, \mathbb{E}) \vdash \{p\}S\{q\}$ for partial correctness is both the possibility of mechanisation and the effect of giving a firm basis for a logical (proof theoretic) investigation of the system.

An essential point is that we want to base our proof system on a specification $(\Sigma, \mathbb{E})$ rather than on a structure A, which is done by most authors. For Hoare's Logic there is no strict need either to consider HL(A) for a fixed datastructure A, and the more general case of $HL(\Sigma,E)$ is clearly of substantial importance.

In various fairly standard approaches to total correctness, such as in HAREL [7] and [8] for deterministic sequential processes and in APT & OLDEROG [1] and GRÜMBERG et al. [6] for fair parallel computation the essence of using a fixed domain A is in the assumption that certain parts of A, as a many-sorted algebra, are well-ordered. This gives rise to quite natural proof rules like the system $HL_T(A)$ that we explain in section 1.1 in order to compare it with our system.

Instead we will develop a device called *schemes* which constitutes a slight generalization of the first order predicate logic. For a specification with schemes we write $(\Sigma, \mathbb{E})$ (whereas (Σ,E) denotes a specification with $E \subseteq L(\Sigma)$). Using schemes we can work in quite a flexible way with signature extensions, a method that proved to be useful and to be of first order character in BERGSTRA & KLOP [2]. Thus we obtain a proof system for termination assertions $(\Sigma, \mathbb{E}) \vdash p \to S \downarrow$ on top of a logic for partial correctness, in the same way as in BERGSTRA & KLOP [2] proof systems for program inclusion are obtained from a partial correctness logic.

We will now sum up the main notations and results.

For a specification $(\Sigma, \mathbb{E})$ with $\mathbb{E}$ a set of schemes, the logic of partial correctness $HL(\Sigma, \mathbb{E})$ brings nothing new. A proof system $(\Sigma, \mathbb{E}) \vdash p \to S \downarrow$ is then defined such that soundness can be shown for a semantics $\models_S$ in Lemma 5.

As a relation of $(\Sigma, \mathbb{E})$, p and S, $\vdash$ is recursively enumerable, thus deserving its denotation as a proof system.

Given a fixed A let $\mathbb{E}_A$ be the set of all schemes Φ over Σ_A that are true in A in the sense of $\models_S$. There is the following completeness result:

THEOREM (9.2) $(\Sigma_A, \mathbb{E}_A) \vdash p \to S \downarrow \Longleftrightarrow A \models p \to S \downarrow$.

In order to compare our system with a usual formalism using well-ordered sets we take the notation $[p]\ S\ [q]$ for total correctness (i.e. $[p]\ S\ [q] \equiv \{p\}\ S\ \{q\}\ \&\ p \to S \downarrow$) and define a system $HL_T(A) \vdash [p]\ S\ [q]$ for datastructures A with a fixed well-ordering $\leq$ on it. Then we define a canonical specification $(\Sigma_A, \mathbb{E}^<_A)$ of such A and state the following result:

THEOREM (11.1) $HL_T(A) \vdash [p]\ S\ [q] \Rightarrow HL(\Sigma_A, \mathbb{E}^<_A) \vdash \{p\}\ S\ \{q\}$ and $(\Sigma_A, \mathbb{E}^<_A) \vdash p \to S \downarrow$.

This result says that the proposed formalism can be used to represent methods using well-ordered sets.

Some final remarks should be made. First of all it would be nice to have a logic for total correctness which is of a first order nature and which is sound and complete for a semantics of specifications and programs which is of first order nature as well. For partial correctness the corresponding problem was solved in BERGSTRA & TUCKER [5]. There a so called axiomatic semantics for while-programs is given such that HL is sound and complete for it in a most general and first order way. It is not clear to us whether or not a similar result can be obtained for total correctness. Anyhow, if we consider simultaneously first order semantics for specifications and the operational semantics (which is not first order) for programs, a proof system $\mid\!\sim$ for $(\Sigma,E) \mid\!\sim p \to S \downarrow$

is either not sound or trivial. This follows immediately from the Compactness Theorem.

Secondly it should be noticed that in principle it is possible to produce a sophisticated proof theory of $(\Sigma, \mathbb{E}) \vdash p \rightarrow S \downarrow$. Indeed, for one structure A already many different and plausible specifications $(\Sigma, \mathbb{E}_i)$ can be found which have different proof theoretic properties. Of course a similar line of investigation is possible for methods using well-ordered sets, but that will require replacing the well-ordering by a better one from time to time. Essentially this involves a modification of the datastructure which seems less attractive from a theoretical point of view.

1. SCHEMES

A scheme will be a generalization of an assertion. Next to the usual predicate-logical symbols a scheme may also contain symbols φ_i^n. The φ_i^n function syntactically as n-ary relation symbols (although their semantics is quite different); the n will mostly be omitted. Formally:

DEFINITION 1.1. The set $Sch(\Sigma)$ of *schemes over the signature* Σ, with typical variable Φ , is inductively defined by:

$$\Phi ::= P_i^n(t_1,\ldots,t_n),\ t_1 = t_2,\ \varphi_i^n(t_1,\ldots,t_n)\ (\text{all } n,i) |$$
$$\Phi_1 \vee \Phi_2,\ \Phi_1 \wedge \Phi_2,\ \neg\Phi\ ,\ \forall x\Phi,\ \exists x\Phi.$$

Here the P_i^n are n-ary predicate symbols from Σ, $t_i \in Ter(\Sigma)$ (the set of Σ-terms) and the φ_i^n are *scheme variables*. The latter are not part of Σ, but will be considered to be standardly included in the language (as logical symbols), just like the ordinary variables x,y,... . Note that $Ass(\Sigma) \subseteq Sch(\Sigma)$, where $Ass(\Sigma)$ is the set of assertions over Σ.

EXAMPLE 1.2. (i) The induction scheme IND $\equiv [\varphi(0) \wedge \forall x(\varphi(x) \rightarrow \varphi(Sx))] \rightarrow \forall x\varphi(x)$.
(ii) $\varphi_1 \rightarrow (\varphi_2 \rightarrow \varphi_1)$, a scheme with 0-ary scheme variables.

NOTATION 1.3. If Φ is a scheme containing precisely the scheme variables $\varphi_1,\ldots,\varphi_n$, we write $\Phi \equiv \Phi(\varphi_1,\ldots,\varphi_n)$.

2. SUBSTITUTION IN SCHEMES

The intended meaning of the scheme variables is that one may substitute assertions for them. For technical reasons it is convenient to allow even substitution of schemes for the scheme variables.

DEFINITION 2.1. Let $\Phi, \Psi \in \mathit{Sch}$. Then $\Phi[\Psi/\varphi(x_1,\ldots,x_n)]$ is the result of replacing each occurrence of the form $\varphi(t_1,\ldots,t_n)$ ($t_i \in \mathit{Ter}$) in Φ, by $\Psi[t_1,\ldots,t_n/x_1,\ldots,x_n]$. ('Ordinary' substitution $[\vec{t}/\vec{x}]$ in a scheme is defined just as for assertions.)

EXAMPLE 2.2. (i) Let $\Phi \equiv \mathrm{IND}$ and $\Psi \equiv x+y = y+x$. Then $\mathrm{IND}[\Psi/\varphi(x)] \equiv \Psi[0/x] \wedge \forall x(\Psi[x/x] \rightarrow \Psi[Sx/x]) \rightarrow \forall x \Psi[x/x] \equiv 0+y = y+0 \wedge \forall x(x+y = y+x \rightarrow Sx+y = y + Sx) \rightarrow \forall x\ x+y = y+x$.
(ii) Let $\Phi \equiv \varphi_1 \rightarrow (\varphi_2 \rightarrow \varphi_1)$. Then $\Phi[\varphi(x)/\varphi_1][\varphi(x)/\varphi_2] \equiv \varphi(x) \rightarrow (\varphi(x) \rightarrow \varphi(x))$.

3. SEMANTICS OF SCHEMES

The most important one of the definitions below is no. (iii) where $A \models_s \Phi$ is defined: A is a standard model of Φ.

DEFINITION 3.1. (i) Let $\Phi \in \mathit{Sch}(\Sigma)$ and let $\Phi \equiv \Phi(\vec{\varphi})$. Then $\Phi \restriction \Sigma = \{\Phi[\vec{p}/\vec{\varphi}] \mid \vec{p} \in \mathit{Ass}(\Sigma)\}$. (E.g., $\mathrm{IND} \restriction \Sigma_{PA}$ is the set of all induction axioms over the signature of Peano's Arithmetic.)
(ii) Let $A \in \mathrm{Alg}$. Then $A \models \Phi$ abbreviates $A \models \Phi \restriction \Sigma_A$. (E.g. we have $\mathbf{A} \models \mathrm{IND}$ for every model $\mathbf{A}$ of PA.)
(iii) $A \models_s \Phi \Longleftrightarrow \forall A' \geq A : A' \models \Phi$. Here $A' \geq A$ means: A' is an *expansion* of A (i.e. A plus added 'structure'). In words: Φ is schematically true in A. (E.g. $N \models_s \mathrm{IND}$. As a contrast, consider a nonstandard model $N^{\#}$ of PA. Then $N^{\#} \models \mathrm{IND}$, but not $N^{\#} \models_s \mathrm{IND}$.)
(iv) If $\mathbb{E} \subseteq \mathit{Sch}(\Sigma)$, we call $(\Sigma, \mathbb{E})$ a *scheme specification*. (Cf. an ordinary specification (Σ, E) where $E \subseteq \mathit{Ass}(\Sigma)$.) Example: $(\Sigma_{PA}, \mathbb{P}\mathbb{A})$, i.e. Peano plus the scheme IND.
(v) Let $\Sigma' \geq \Sigma$. Then $(\Sigma, \mathbb{E})_{\Sigma'} = (\Sigma', \mathbb{E} \restriction \Sigma')$. Here $\mathbb{E} \restriction \Sigma' = \{\Phi[\vec{p}/\vec{\varphi}] \mid p \in \mathit{Ass}(\Sigma'), \Phi(\vec{\varphi}) \in \mathbb{E}\}$. (So, by attaching Σ' as subscript the scheme specification is transformed to an ordinary specification.)
(vi) Let $A \in \mathrm{Alg}(\Sigma)$. Then $A \models (\Sigma, \mathbb{E})$ abbreviates $A \models (\Sigma, \mathbb{E})_{\Sigma}$.
(vii) Let $A \in \mathrm{Alg}(\Sigma)$. Then: $A \models_s (\Sigma, \mathbb{E}) \Longleftrightarrow A \models_s \Phi, \forall \Phi \in \mathbb{E}$.
(viii) $\mathrm{Alg}_s(\Sigma, \mathbb{E}) = \{A \in \mathrm{Alg}(\Sigma) \mid A \models_s (\Sigma, \mathbb{E})\}$. (E.g. $\mathrm{Alg}_s(\Sigma_{PA}, \mathbb{P}\mathbb{A}) = \{N\}$.)
(ix) $\mathrm{Alg}_s(\Sigma, \mathbb{E}) \models_s \Phi \Longleftrightarrow \forall A \in \mathrm{Alg}_s(\Sigma, \mathbb{E})\ A \models_s \Phi$. Instead of the LHS we will also write simply $(\Sigma, \mathbb{E}) \models_s \Phi$.

4. DERIVABILITY OF SCHEMES

DEFINITION 4.1. $(\Sigma, \mathbb{E}) \vdash \Phi$ is defined as the usual derivability of an assertion from a specification (to this end the φ_i^n are treated as n-ary predicate symbols) plus the *substitution rule:*

$$\frac{\Phi_1}{\Phi_1[\Phi_2/\varphi(\vec{x})]}$$

for all $\Phi_1, \Phi_2 \in \mathit{Sch}(\Sigma)$ and all scheme variables φ.

PROPOSITION 4.2. $(\Sigma, \mathbb{E}) \vdash p \Leftrightarrow (\Sigma, \mathbb{E})_\Sigma \vdash p$, *for all* $p \in \mathit{Ass}(\Sigma)$.

PROOF. ($\Leftarrow$) trivial; ($\Rightarrow$) induction on the length of the proof of $(\Sigma, \mathbb{E}) \vdash p$. (This amounts to commutativity of substitution and derivability in the usual sense.) □

The following lemma presents a useful soundness result:

LEMMA 5. $(\Sigma, \mathbb{E}) \vdash \Phi \Rightarrow (\Sigma, \mathbb{E}) \models_s \Phi$.

PROOF. Assume $(\Sigma, \mathbb{E}) \vdash \Phi$ and consider a structure A with $A \models_s (\Sigma, \mathbb{E})$. We show that $A \models_s \Phi$. Therefore consider $A' \geq A$ with $A' \models (\Sigma, \mathbb{E})$ and $\Sigma' = \Sigma_{A'}$.
The following sequence of implications establishes $A' \models \Phi$:

$$\begin{array}{l}
(\Sigma, \mathbb{E}) \vdash \Phi(\vec{\varphi}) \Rightarrow \\
(\Sigma', \mathbb{E}) \vdash \Phi(\vec{\varphi}) \Rightarrow \\
(\Sigma', \mathbb{E}) \vdash \Phi(\vec{p}) \text{ for all } p \in \mathit{Ass}(\Sigma') \Rightarrow (4.2) \\
(\Sigma', \mathbb{E})_{\Sigma'} \vdash \Phi(\vec{p}) \text{ " " " " " } \Rightarrow \\
(\Sigma, \mathbb{E})_{\Sigma'} \vdash \Phi(\vec{p}) \text{ " " " " " } .
\end{array}$$

Of course $A' \models (\Sigma, \mathbb{E})$ implies $A' \models (\Sigma, \mathbb{E})_{\Sigma'}$ and consequently

$$A' \models \Phi(\vec{p}) \text{ for all } p \in \mathit{Ass}(\Sigma')$$

which is $A' \models \Phi$. □

REMARK 5.1. The corresponding completeness result fails. To see this let us consider the example $(\Sigma_{PA}, \mathbb{P}\mathbb{A})$. Completeness of $\vdash$ w.r.t. $\models_s$ would entail

$$(\Sigma_{PA}, \mathbb{P}\mathbb{A}) \vdash \Phi \Leftrightarrow (\Sigma_{PA}, \mathbb{P}\mathbb{A}) \models_s \Phi$$

for all Φ, and especially for all $p \in \mathit{Ass}(\Sigma_{PA})$: $(\Sigma_{PA}, \mathbb{P}\mathbb{A}) \vdash p \Leftrightarrow (\Sigma_{PA}, \mathbb{P}\mathbb{A}) \models_s p$.
Now $\mathrm{Alg}_s(\Sigma_{PA}, \mathbb{P}\mathbb{A}) = \{N\}$ and we find $(\Sigma_{PA}, \mathbb{P}\mathbb{A}) \vdash p \Leftrightarrow N \models_s p$.
From 4.2 and $(\Sigma_{PA}, \mathbb{P}\mathbb{A})_{\Sigma_{PA}} = (\Sigma_{PA}, PA)$ this leads to $PA \vdash p \Leftrightarrow N \models_s p$

which contradicts Gödel's incompleteness theorem.

DEFINITION 6. The schematic theory $\mathbb{E}_A$ of a structure A is defined as the set of all schemes $\Phi \in Sch(\Sigma_A)$ such that $A \models_S \Phi$.

LEMMA 6.1. *The following are equivalent:*

(i) $(\Sigma_A, \mathbb{E}_A) \vdash \Phi$

(ii) $(\Sigma_A, \mathbb{E}_A) \models_S \Phi$

(iii) $A \models_S \Phi$.

PROOF. (i) $\Rightarrow$ (ii) according to Lemma 5. (ii) $\Rightarrow$ (iii) $\Rightarrow$ (i) are evident from the definitions. □

DEFINITION 7. A^S is the maximal (full) expansion of A, i.e. A^S is a structure (with presumably an uncountable signature) which contains a name for each possible relation function or constant on it.

The following property follows easily:

PROPOSITION 7.1. $A \models_S \Phi \Longleftrightarrow A^S \models \Phi$.

A^S will be used in the proof of Theorem 9.2. Moreover, in sections 10 and 11 we will use the partial correctness logic $HL(\Sigma, \mathbb{E})$ for schematic specifications.

DEFINITION 7.2. $HL(\Sigma, \mathbb{E}) \vdash \{\varphi\}\ S\ \{\psi\}$ is Hoare's logic over $(\Sigma, \mathbb{E})$.

Syntactically one requires that $S \in WP(\Sigma)$ and φ , $\psi \in Sch(\Sigma)$. Its axioms and rules are exactly the same as usually for HL, the only difference being that schemes may occur at the position of assertions in the original system.

8. TERMINATION ASSERTIONS

DEFINITION 8.1. (i) Let $p \in Ass(\Sigma)$ and $S \in WP(\Sigma)$. Then $p \rightarrow S\downarrow$ is a termination assertion.

(ii) (Semantics:) If $A \in Alg(\Sigma)$ then: $A \models p \rightarrow S\downarrow \Longleftrightarrow S$ converges on every input $\vec{a} \in A$ such that $A \models p(\vec{\underline{a}})$.

The next definition is based on the concept of 'prototype proof' $\pi(S)$ as defined in BERGSTRA & KLOP [2]. This is roughly a scheme of which every ordinary proof of $\{p\}S\{q\}$ is a substitution instance. To this end we view a proof of $\{p\}S\{q\}$ as an 'interpolated statement', i.e. a statement in which assertions may occur; see Example 8.5 of a $\pi(S)$. For the precise details we refer to BERGSTRA & KLOP [2].

Simply speaking a prototype proof $\pi(S)$ is obtained by using scheme variables as precondition, postcondition, intermediate assertions and invariants. Let $\varphi \ldots \psi$ be the scheme variables occurring in $\pi(S)$ in linear order. Then the logical information that is required about $\varphi \ldots \psi$ is a scheme $\varphi \overset{S}{\rightsquigarrow} \psi$ which incorporates all implications that are used in applications of the rule of consequence.

DEFINITION 8.2. Let $S \in \mathit{WP}(\Sigma)$. Then $\varphi \overset{S}{\rightsquigarrow} \psi$ abbreviates the scheme $\forall(\bigwedge \kappa(\{\varphi\}\pi(S)\{\psi\}))$, where $\pi(S)$ is the prototype proof of S, κ denotes the set of implications used in $\{\varphi\}\pi(S)\{\psi\}$, and $\forall$ denotes the universal closure. Here φ,ψ are scheme variables different from those in $\pi(S)$. (As in BERGSTRA & KLOP [2] and in Example 8.5, we will denote the scheme variables in $\pi(S)$ by $r_1, r_2, \ldots$.)

Now we have the following proposition; the proof is routine and therefore omitted.

PROPOSITION 8.3. (i) $\varphi \overset{S_1;S_2}{\rightsquigarrow} \psi \vdash \varphi \overset{S_1}{\rightsquigarrow} r \wedge r \overset{S_2}{\rightsquigarrow} \psi$ *for some* r.
(ii) $\varphi_1 \overset{S}{\rightsquigarrow} \psi_1 \wedge \varphi_2 \overset{S}{\rightsquigarrow} \psi_2 \vdash \varphi_1 \wedge \varphi_2 \overset{S}{\rightsquigarrow} \psi_1 \wedge \psi_2$.
(iii) $HL(\Sigma, \mathbb{E}) \vdash \{\varphi\}S\{\psi\} \Longleftrightarrow (\Sigma, \mathbb{E}) \vdash \varphi \overset{S}{\rightsquigarrow} \psi$ *for some proof scheme* $\varphi \overset{S}{\rightsquigarrow} \psi$.

(*In fact we must write* $\varphi(\vec{x})$, $\psi(\vec{x})$ *etc. instead of* φ,ψ *where* $\vec{x}$ *is a list of the relevant variables.*)

The next definition is crucial.

DEFINITION 8.4. Let $p \to S\downarrow$ be a termination assertion. Then $\Phi(p \to S\downarrow)$ is the corresponding *termination scheme*, defined by:

$$\Phi(p \to S\downarrow) \equiv (\{p \wedge \varphi(\vec{x})\} \overset{S}{\rightsquigarrow} \{\underline{\text{false}}\}) \to \neg\, \exists\vec{x}(p \wedge \varphi(\vec{x})).$$

Here $\vec{x}$ is a list of the free variables in p and the variables in S.

EXAMPLE 8.5. Let $S \equiv$ while $x \neq 0$ do $x := P(x)$ od, in the signature of PA; P is the predecessor function.

```
Now   $\pi(S) \equiv$
        $\{r_0(x)\}$
        $\{r_1(x)\}$
while $x \neq 0$ do
          $\{r_1(x) \wedge x \neq 0\}$
          $\{r_2(Px)\}$
   $x := P(x)$
          $\{r_2(x)\}$
          $\{r_1(x)\}$
od
          $\{r_1(x) \wedge x = 0\}$
          $\{r_3(x)\}$.
```

Let us determine the termination scheme $\Phi(\underline{\mathrm{true}} \rightarrow S\downarrow)$.

$\varkappa(\{\underline{\mathrm{true}} \wedge \varphi(x)\}\ \pi(S)\{\underline{\mathrm{false}}\}) =$

$\{\ \underline{\mathrm{true}} \wedge \varphi(x) \rightarrow r_0(x),$
$\quad r_0(x) \rightarrow r_1(x),$
$\quad r_1(x) \wedge x \neq 0 \rightarrow r_2(Px),$
$\quad r_2(x) \rightarrow r_1(x),$
$\quad r_1(x) \wedge x = 0 \rightarrow r_3(x),$
$\quad r_3(x) \rightarrow \underline{\mathrm{false}}\}$.

Now $\Phi(\underline{\mathrm{true}} \rightarrow S\downarrow) = \sigma \rightarrow \neg\ \exists x\ \varphi(x)$, where σ is the universal closure of the conjunction of the six implications above.

Note that $\Phi \equiv \Phi(\underline{\mathrm{true}} \rightarrow S\downarrow)$ is none other than IND, to be precise: $(\Sigma_{PA}, \mathbb{P}\mathbb{A}) \vdash \Phi \longleftrightarrow$ IND. Here $\Phi \rightarrow$ IND follows by the substitution $\varphi(x) \equiv r_0(x) \equiv r_1(x) \equiv r_2(x)$ in Φ and by deriving from σ that $\neg\ \varphi(0) \wedge \forall x(\neg\varphi(x) \rightarrow \neg\ \varphi(Sx))$ (where S denotes the successor function).

NOTATION 8.6. We will write often $(\Sigma, \mathbb{E}) \vdash p \rightarrow S\downarrow$ instead of $(\Sigma, \mathbb{E}) \vdash \Phi(p \rightarrow S\downarrow)$.

9. Before formulating the main theorem we need the following proposition, whose routine proof is omitted.

PROPOSITION 9.1. $A^S \models \Phi(p \rightarrow S\downarrow) \Longleftrightarrow A^S \models p \rightarrow S\downarrow$.

THEOREM 9.2. *The following are equivalent:*
(i) $(\Sigma_A, \mathbb{E}_A) \vdash \Phi(p \rightarrow S\downarrow)$
(ii) $A \models_S \Phi(p \rightarrow S\downarrow)$
(iii) $A \models p \rightarrow S\downarrow$ . .

COMMENT: This result indicates the completeness of $(\Sigma, \mathbb{E}) \vdash \Phi(p \rightarrow S\downarrow)$ as a logic for total correctness.

PROOF: (i) $\Longleftrightarrow$ (ii) by Lemma 6.1. (ii) $\Longleftrightarrow$ (iii):

$A \models_S \Phi(p \rightarrow S\downarrow) \Longleftrightarrow$ (by Proposition 7.1)
$A^S \models \Phi(p \rightarrow S\downarrow) \Longleftrightarrow$ (by Proposition 9.1)
$A^S \models p \rightarrow S\downarrow \Longleftrightarrow$ (trivially)
$A \models p \rightarrow S\downarrow$. □

10. $(\Sigma_{PA}, \mathbb{P}\mathbb{A})$, AN EXAMPLE IN DETAIL

Let N be the structure $(\omega,+,\cdot,S,P,0)$ and let $\mathbb{P}\mathbb{A}$ be a suitable version of Peano's arithmetic on N with a scheme for induction as indicated in the example in 1.2

We will list here some properties of the partial and total correctness logics based on $(\Sigma, \mathbb{P}\mathbb{A}) = (\Sigma_{PA}, \mathbb{P}\mathbb{A})$.

As a matter of fact $(\Sigma, \mathbb{P}\mathbb{A}) \vdash p \to S\downarrow$ is incomplete for total correctness on N. This is easily seen from the fact that the set of programs S with $(\Sigma, \mathbb{P}\mathbb{A}) \vdash \underline{true} \to S\downarrow$ is Σ_1^0 whereas on the other hand $N \models \underline{true} \to S\downarrow$ is a complete Π_2^0 predicate of programs S. The example 8.5 shows, however, that $(\Sigma, \mathbb{P}\mathbb{A})$ proves the termination of nontrivial programs.

11. RELATIONS WITH A STANDARD PROOF METHOD

Let A be a data structure containing a binary relation $<$ which is in fact a well ordering of A with smallest element $o \in |A|$. For A we have a system of proving total correctness $HL_T(A)$ and a canonical specification $(\Sigma_A, \mathbb{E}_A^<)$. After detailed definitions we prove the following result which indicates that $HL_T(A)$ can be formalized via $(\Sigma_A, \mathbb{E}_A^<)$ and its total and partial correctness logic.

THEOREM 11.1. *If*

$$HL_T(A) \vdash [p]S[q]$$

then

$$HL(\Sigma_A, \mathbb{E}_A^<) \vdash \{p\}S\{q\}$$

and

$$(\Sigma_A, \mathbb{E}_A^<) \vdash p \to S\downarrow .$$

The system $HL_T(A)$ is nothing new, versions of it appeared in [1], [5], [6] and [7] and various other places. The intended meaning of $[p]S[q]$ is: $\{p\}S\{q\}$ & $p \to S\downarrow$.

DEFINITION 11.2. $HL_T(A)$ has the following rules:

(i) $$[p[t/x]]\ x := t\ \ [p]$$

(ii) $$\frac{[p]S_1[q]\quad [q]S_2[r]}{[p]S_1;S_2[r]}$$

(iii) $$\frac{[p \wedge b]S_1[q] \qquad [p \wedge \neg b]S_2[q]}{[p]\ \underline{if}\ b\ \underline{then}\ S_1\ \underline{else}\ S_2\ \underline{fi}\ [q]}$$

(iv) $$\frac{A \models p \to p' \quad [p']S[q'] \quad A \models q' \to q}{[p]S[q]}$$

(v) $$\frac{[I(\alpha) \wedge b]S[\exists\beta < \alpha\ I(\beta)] \quad A \models I(0) \to \neg b}{[I_0]\ \underline{while}\ b\ \underline{do}\ S\ \underline{od}\ [I_0 \wedge \neg b]}$$

where $I_0 \equiv \exists\alpha\ I(\alpha)$ and $\alpha,\beta \notin VAR(S)$.

11.3. $(\Sigma_A, \mathbb{E}^{<}_{A})$ consists of E_A, the theory of A in $A\mathit{ss}(\Sigma_A)$, and the scheme $\mathbb{E}^{<}$ of induction along $<$:

$$\forall\beta\,[(\forall\alpha(\alpha < \beta \to \varphi(\alpha))) \to \varphi(\beta)] \to \forall\alpha\ \varphi(\alpha).$$

11.4. We can now prove the theorem. The first part concerns partial correctness. This is a straightforward induction on program depth, except in the case of the <u>while</u> rule. We will consider this case.

Suppose that

$$[I_0]\ \underline{\text{while}}\ b\ \underline{\text{do}}\ S_0\ \underline{\text{od}}\ [I_0 \wedge \neg b]$$

has been deduced from

$$[I(\alpha) \wedge b]S_0[\exists\beta < \alpha\ I(\beta)],\ A \models I(0) \to \neg b$$

with $I_0 \equiv \exists\alpha\ I(\alpha)$.

From the induction hypothesis we find (in $HL(\Sigma_A, \mathbb{E}^{<}_{A})$):

$$\vdash \{I(\alpha) \wedge b\}\ S_0\ \{\exists\beta < \alpha\ I(\beta)\}$$

using the rule of consequence then

$$\vdash \{I(\alpha) \wedge b\}\ S_0\ \{I_0\}$$

and with existential generalization on the precondition

$$\vdash \{I_0 \wedge b\}\ S_0\ \{I_0\}$$

then with the <u>while</u> rule

$$\vdash \{I_0\}\ \underline{\text{while}}\ b\ \underline{\text{do}}\ S_0\ \underline{\text{od}}\ \{I_0 \wedge \neg b\}.$$

11.5. The second part of the proof involves showing $(\Sigma_A, \mathbb{E}_A) \vdash p \to S\downarrow$. We abbreviate $(\Sigma_A, \mathbb{E}^{<}_{A})$ to $(\Sigma, \mathbb{E})$ in this part of the proof. Of course we use induction on the structure of the proof of $[p]S[q]$. With X we denote the variables occurring free in p, S, q.

Suppose that $[p]S[q]$ was obtained by applying the rule of consequence to $[p']S[q']$, then by the induction hypothesis $(\Sigma, \mathbb{E}) \vdash p' \to S\downarrow$; an easy logical calculation then shows $(\Sigma, \mathbb{E}) \vdash p \to S$ because $\mathbb{E} \vdash p \to p'$.

For the case $S \equiv x := t$ we explain the argument in detail.

$$(\Sigma, \mathbb{E}) \vdash \forall X(p \wedge \varphi \to \underline{\text{false}}) \supset \neg\, \exists X\ p \wedge \varphi$$

because this is a tautology. Then

$$(\Sigma, \mathbb{E}) \vdash (\forall X(p \wedge \varphi \rightarrow r[t/x]) \wedge \forall x(r \rightarrow \underline{\text{false}})) \rightarrow \neg \exists X\ p \wedge \varphi\ .$$

thus

$$(\Sigma, \mathbb{E}) \vdash [p \wedge \varphi \overset{x:=t}{\rightsquigarrow} \underline{\text{false}}] \rightarrow \neg \exists X\ p \wedge \varphi$$
$$(\Sigma, \mathbb{E}) \vdash \varphi(p \rightarrow S\downarrow)$$
$$(\Sigma, \mathbb{E}) \vdash p \rightarrow S\downarrow .$$

The argument in case [p]S[q] was obtained from an application of the conditional rule 11.2 (iii) is entirely straightforward and is therefore omitted.

The harder cases of composition and iteration remain and we treat composition here. Let $S \equiv S_1;S_2$. Assume $HL_T(A) \vdash [p]S[q]$. Choose an assertion u with

$$HL_T(A) \vdash [p]S_1[u], HL_T(A) \vdash [u]S_2[q].$$

We show that $(\Sigma, \mathbb{E}) \vdash p \rightarrow S\downarrow$. It is sufficient to derive, working in $(\Sigma, \mathbb{E})$, $\neg\exists X\ p \wedge \varphi$ from $p \wedge \varphi \overset{S}{\rightsquigarrow} \text{false}$. So assume $p \wedge \varphi \overset{S}{\rightsquigarrow} \underline{\text{false}}$. Then for some r: $p \wedge \varphi \overset{S1}{\rightsquigarrow} r$ and $r \overset{S2}{\rightsquigarrow} \underline{\text{false}}$.
Because of $HL(\Sigma, \mathbb{E}) \vdash \{p\}S_1\{u\}$ (part (i) of this theorem) one obtains a proof scheme $p \overset{S_1}{\rightsquigarrow} r$, combining this one with $p \wedge \varphi \overset{S_1}{\rightsquigarrow} u$ one obtains using Proposition 8.3 $p \wedge \varphi \overset{S_1}{\rightsquigarrow} r \wedge u$; from $r \overset{S_2}{\rightsquigarrow} \underline{\text{false}}$ one immediately obtains $r \wedge u \overset{S_2}{\rightsquigarrow} \underline{\text{false}}$.

Now using the induction hypothesis on S_2 we know that $(\Sigma, \mathbb{E}) \vdash \Phi(r \rightarrow S_2\downarrow)$ thus $(\Sigma, \mathbb{E}) \vdash (r \wedge \Phi \overset{S_2}{\rightsquigarrow} \underline{\text{false}}) \rightarrow \neg\exists Xr \wedge \varphi$. Substituting u for φ and applying modus ponens we obtain $\neg\exists Xr \wedge u$. After applying the rule of consequence on $p \wedge \varphi \overset{S1}{\rightsquigarrow} r \wedge u$, $\forall X(r \wedge u) \rightarrow \underline{\text{false}}$ we find $p \wedge \varphi \overset{S1}{\rightsquigarrow} \underline{\text{false}}$. The induction hypothesis on S_1 then immediately yields $\neg\exists X\ p \wedge \varphi$.
The case that $S \equiv \underline{\text{while}}\ b\ \underline{\text{do}}\ S_0\ \underline{\text{od}}$ is similar but tedious and will be omitted. It occurs in detail in [3]. □

REFERENCES.

[1] APT, K.R. & E.R. OLDEROG, *Proof rules dealing with fairness*, Bericht Nr. 8104, March 1981, Institut für Informatik und Praktische Mathematik, Christian Albrechts-universität Kiel.

[2] BERGSTRA, J.A. & J.W. KLOP, *Proving program inclusion using Hoare's Logic*, Mathematical Centre, Department of Computer Science, Research Report IW 176, Amsterdam 1981.

[3] BERGSTRA, J.A. & J.W. KLOP, *A formalized proof system for total correctness of while programs*, Mathematical Centre, Department of Computer Science, Research Report IW 175, Amsterdam 1981.

[4] BERGSTRA, J.A. & J.V. TUCKER, *Hoare's Logic and Peano's Arithmetic*, Mathematical Centre, Department of Computer Science, Research Report IW 160, Amsterdam 1981.

[5] BERGSTRA, J.A. & J.V. TUCKER, *The axiomatic semantics of while programs using Hoare's Logic*, manuscript May 1981, definitive version in preparation.

[6] GRÜMBERG, O., N. FRANCEZ, J.A. MAKOWSKY & W.P. DE ROEVER, *A Proof Rule for fair Termination of Guarded Commands*, Report RUU-CS-81-2, Vakgroep Informatica Utrecht.

[7] HAREL, D., *First Order Dynamic Logic*, Springer Lecture Notes in Comp. Sc. 68, 1979.

[8] HAREL, D., *Proving the correctness of regular deterministic programs: a unifying survey using dynamic Logic*, T.C.S. 12(1) 61-83.