# NONSTANDARD RUNS OF FLOYD-PROVABLE PROGRAMS

I. Németi

Mathematical Institute of the Hungarian Academy of Sciences

Budapest, Reáltanoda u. 13-15, H-1053 Hungary

The question is investigated: "exactly which programs are provable by the Floyd-Hoare inductive assertions method?".

Theorem 1 of this paper says that from any theory $T$ containing the Peano axioms exactly those programs are Floyd-Hoare provable which are partially correct in the models of $T$ w.r.t. continuous traces. Intuitively: the provable programs are the ones which are correct in every perhaps nonstandard machine functioning perhaps in a nonstandard time. Of course every nonstandard machine and time has to satisfy our axioms $T$. This result was first proved in Andréka-Németi [1] in Hungarian. It was announced in English in [2], [3] and was quoted in Salwicki [23], Csirmaz [13], Richter-Szabo [20] etc.

In section 5 concrete examples of simple nonstandard runs of programs are constructively defined and illustrated on figures. The emphasis in section 5 is on simplicity, with the aim to make nonstandard runs and nonstandard models less esoteric, less imaginary, easy to draw, easy to touch. In the proof of Proposition 3 it is demonstrated how ultraproducts can be used to test applicability of Floyd's method in concrete situations.

We are specifically interested in the behaviour of programs (or "program schemes") in first order *axiomatizable classes* of models (or "interpretations").

The central notion of the present paper is that of continuous traces. Properties of continuous traces were investigated in Csirmaz [13]. A simpler and much more natural notion was introduced in [5], [6], [21], [4], [7], [19]. This improved approach was used in Csirmaz-Paris [15], Sain [22], Csirmaz [14]. The quoted works use the general methodology elaborated in Dahn [16] and Sain [21] for investigating new logics.

The most readable introduction is [4] or if that is not available then [7]. Further important works in the present nonstandard direction are Hájek [17], Richter-Szabo [20]. For more uses of ultraproducts (cf. Proposition 3 here) see [19], [4] and a little in [7]. Copies of all the above quoted papers of Andréka, Csirmaz, Németi or Sain *are available* from the present author except [2] and [12].

## 1. SYNTAX

Let $t$ be a *similarity type* assigning arities to function symbols and relation symbols. $\omega$ denotes the set of *natural numbers*.

$Y \stackrel{d}{=} \{y_i : i\in\omega\}$ is called the set of *variable symbols* and is disjoint from everything we use. Logical symbols: $\{\wedge,\neg,\exists\}$. Other symbols: $\{\leftarrow$, IF, THEN, (,), :$\}$. The set of "label symbols" is $\omega$ itself.

$L_t$ denotes the set of all *first order formulas* of type $t$ possibly with free variables (elements of $Y$ of course), see e.g. [10] p.22. We shall refer to "*terms* of type $t$" as defined in e.g. [10] p. 22.

Now we define the set $P_t$ of *programs* of type $t$.

The set $U_t$ of *commands* of type $t$ is defined by:

$(j : y \leftarrow \tau) \in U_t$ if $j\in\omega$, $y\in Y$ and $\tau$ is a term of type $t$.

$(j :$ IF $\lambda$ THEN $v) \in U_t$ if $j,v\in\omega$, $\lambda\in L_t$ is a formula without quantifiers.

These are the only elements of $U_t$.

If $(i:u)\in U_t$ then $i$ is called the *label* of the command $(i:u)$.

By a *program* of type $t$ we understand a finite sequence of commands (elements of $U_t$) in which no two members have the same label. Formally, the set of programs is:

$$P_t \stackrel{d}{=} \{\langle (i_0:u_0),\ldots,(i_n:u_n)\rangle \;:\; n\in\omega,\;\; (\forall e\leq n)(i_e:u_e)\in U_t,\;\; (\forall e<k\leq n)\, i_k\neq i_e\}.$$

For every $p \stackrel{d}{=} \langle (i_0:u_0),\ldots,(i_n:u_n)\rangle \in P_t$ we shall use the notation $i_{n+1} \stackrel{d}{=} \min(\omega\setminus\{i_m : m\leq n\})$.

EXAMPLE: Let $t$ contain the function symbols "+,·,0,1" with arities "2,2,0,0 " respectively. Now the sequence

$$\langle (0:\ y_1 \leftarrow 0), (1:\ \text{IF } y_1=y_2 \text{ THEN } 4), (2:\ y_1 \leftarrow y_1+1), (3:\ \text{IF } y_2=y_2 \text{ THEN } 1)\rangle$$

is a program of type $t$. See Figure 1.

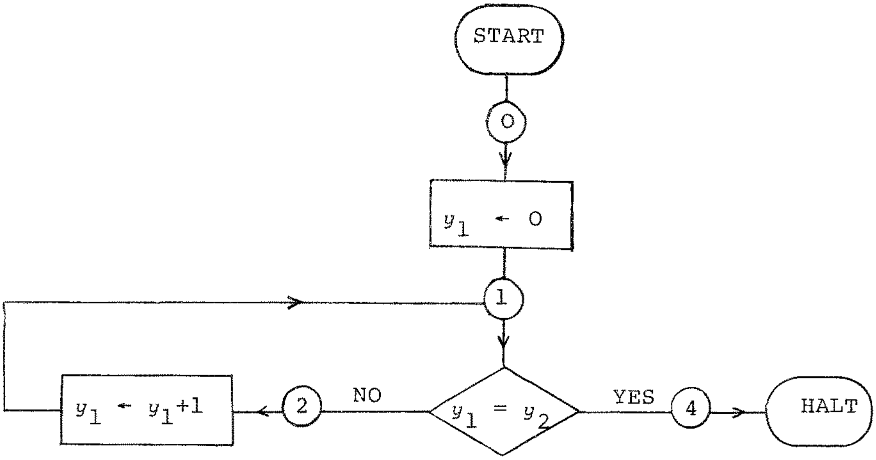


FIGURE 1

## 2. SEMANTICS

Let $p \in P_t$ be a program and $\mathfrak{A}$ be a structure or model of type $t$, see [10]p.20. The universe of a model denoted by $\mathfrak{A}$ will always be denoted by $A$.

$V_p$ denotes the variable symbols occurring in $p$. Note that $V_p$ is a finite subset of $Y$.

By a valuation (of the variables of $p$) in $\mathfrak{A}$ we understand a function $q : V_p \to A$ (cf. [8]p.55).

Let $\tau$ be a term occurring in $p$. Now $\tau[q]_{\mathfrak{A}}$ denotes the value of the term $\tau$ in the model $\mathfrak{A}$ at the valuation $q$ of the variable symbols, cf. [10]p.27 Def.1.3.13. We shall often write $\tau[q]$ i.e. we shall omit the subscript $\mathfrak{A}$.

From now on we work with the *similarity type of arithmetic*. I.e. $t$ is fixed to consist of "$+,\cdot,0,1$" with arities "2,2,0,0". We shall omit the index $t$ since it is fixed anyway.

$\mathfrak{N}$ denotes the standard model of arithmetic, that is $\mathfrak{N} \overset{d}{=} \langle \omega,+,\cdot,0,1 \rangle$ where $+,\cdot,0,1$ are the usual.

EXAMPLE: Let $V_p=\{y_1,y_2\}$, $q(y_1)=2$, $q(y_2)=3$, $\tau=((y_1+y_2)+y_2)$. Then $\tau[q]_{\mathfrak{N}}=8$.

$PA\subseteq L$ denotes the (recursive) set of the *Peano-axioms* (together with the induction axioms), see [10]p.42 Ex.1.4.11 (axioms 1-7). We shall only be concerned with models of the Peano-axioms.

We are going to define the *continuous traces* of a program in a model of $PA$.

DEFINITION 1 Let $\mathfrak{A}\models PA$ be an arbitrary model (of Peano arithmetic). Let $p\in P$ be a program with set $V_p$ of variables.

A *trace* of $p$ in $\mathfrak{A}$ is a sequence $s \overset{d}{=} \langle s_a\rangle_{a\in A}$ indexed by the elements of $A$ such that (i) and (ii) below are satisfied.

(i) $s_a : V_p\cup\{\lambda\} \to A$ is a valuation of the variables (of $p$) into $\mathfrak{A}$, where $\lambda\in Y\setminus V_p$ is a variable not occurring in $p$. $\lambda$ can be conceived of as the "control variable of $p$". (We could call $s_a$ a "*state*" of $p$ in the model $\mathfrak{A}$.)

(ii) To formulate this condition, let $p = \langle (i_0:u_0),\dots,(i_n:u_n)\rangle$ and recall the notation $i_{n+1} \overset{d}{=} \min(\omega\setminus\{i_m : m\le n\})$. Now we demand

$s_0(\lambda) = i_0$ and for any $a\in A$,

if $s_a(\lambda)\notin\{i_m : m\le n\}$ then $s_{a+1}=s_a$ else

for all $m\le n$ such that $s_a(\lambda)=i_m$, conditions a) and b) below hold.

a) if $u_m$ = "$y_w\leftarrow\tau$" then

$s_{a+1}(\lambda)=i_{m+1}$ and for any $x\in V_p$,

$$s_{a+1}(x) \doteq \begin{cases}\tau[s_a]_{\mathfrak{A}} & \text{if } x=y_w\\ s_a(x) & \text{otherwise.}\end{cases}$$

b) if $u_m$ = "IF $\chi$ THEN $v$" then

$$s_{a+1}(\lambda) = \begin{cases} v & \text{if } \mathfrak{A}\models\chi[s_a]\\ i_{m+1} & \text{otherwise, and}\end{cases}$$

$s_{a+1}(x) = s_a(x)$, for every $x\in V_p$.

By this we have defined traces of a program in $\mathfrak{A}$ as sequences

$\langle s_a \rangle_{a\in A}$ "respecting the structure" of the program. End of Def.1.

DEFINITION 2 The sequence $\langle s_a \rangle_{a\in A}$ is continuous in $\mathfrak{A}$ if $\langle s_a \rangle_{a\in A}$ satisfies the induction axioms, that is if for any $\varphi\in L$ with free variables in $V_p$ we have

$$\mathfrak{A} \models ((\varphi[s_0] \wedge \bigwedge_{a\in A} (\varphi[s_a] \rightarrow \varphi[s_{a+1}])) \rightarrow \bigwedge_{a\in A} \varphi[s_a]).$$

By a *continuous trace* of $p$ in $\mathfrak{A}$ we understand a trace $\langle s_a \rangle_{a\in A}$ of $p$ which is continuous. End of Def.2.

Note that in the standard model $\mathfrak{N}$ every trace is continuous.

Intuitively, a trace $\langle s_a \rangle_{a\in A}$ is continuous if whenever a first order property $\varphi\in L$ changes during time $(A)$, then there exists a *point of time* $(a\in A)$ when this change is just happening:

$\mathfrak{A} \models \varphi[s_0]$ and $(\exists a\in A)\ \mathfrak{A} \not\models \varphi[s_a]$ together imply that

$(\exists a\in A)(\mathfrak{A} \models \varphi[s_a]$ and $\mathfrak{A} \not\models \varphi[s_{a+1}])$.

DEFINITION 3 Let $p = \langle (i_0:u_0),\dots,(i_n:u_n) \rangle \in P$ and $\psi\in L$ be such that the free variables of $\psi$ are in $V_p$. Let $\mathfrak{A} \models PA$.

The pair $(p,\psi)$ is said to be *partially correct* in $\mathfrak{A}$ w.r.t. continuous traces if for any continuous trace $\langle s_a \rangle_{a\in A}$ of $p$ in $\mathfrak{A}$ and for any $a\in A$ $s_a(\lambda) \notin \{i_m : m\leq n\}$ implies $\mathfrak{A} \models \psi[s_a]$.

$\mathfrak{A} \overset{pc}{\models} (p,\psi)$ denotes that the pair $(p,\psi)$ is partially correct in $\mathfrak{A}$ w.r.t. continuous traces. End of Def.3.

## 3. DERIVATION SYSTEM (rules of inference)

In the following definition we recall the so called Floyd-Hoare derivation system. This system serves to derive pairs $(p,\psi)$ (where $p\in P$ and $\psi\in L$) from theories $T\subseteq L$.

DEFINITION 4 Let $p = \langle (i_0,u_0),\dots,(i_n:u_n) \rangle \in P$, let $\psi\in L$ and let $T\subseteq L$. The set of labels of $p$ is defined as

$$\mathrm{lab}(p) \overset{d}{=} \{i_m : m\leq n+1\} \cup \{v : (\exists m\leq n)u_m = \text{"IF } \chi \text{ THEN } v\text{"}\}.$$

Note that lab($p$) is finite.

A *Floyd-Hoare derivation* of $(p,\psi)$ from $T$ consists of a mapping $\Phi : \text{lab}(p) \to L$ together with classical first-order derivations listed in (i)-(iv) below.

Notation: When $z\in\text{lab}(p)$ we write $\Phi_z$ instead of $\Phi(z)$.

(i) A derivation $T \vdash \Phi_{i_0}$ .

(ii) To each command $(i_m : y_j \leftarrow \tau)$ occurring in $p$ a derivation $T \vdash (\Phi_{i_m} \to \Phi_{i_{m+1}}(y_j/\tau))$, where $\varphi(y/\tau)$ denotes the formula obtained from $\varphi$ by substituting $\tau$ in place of $y$ in the usual way, cf. [8]p.61.

(iii) To each command $(i_m$ : IF $\chi$ THEN $v)$ occurring in $p$ derivations $T \vdash ((\chi\wedge\Phi_{i_m}) \to \Phi_v)$ and $T \vdash ((\neg\chi\wedge\Phi_{i_m}) \to \Phi_{i_{m+1}})$.

(iv) To each $z \in (\text{lab}(p)\setminus\{i_m : m\leq n\})$ a derivation $T \vdash (\Phi_z \to \psi)$.

The existence of a Floyd-Hoare derivation of $(p,\psi)$ from $T$ is denoted by $T \overset{FH}{\vdash} (p,\psi)$. <u>End of Def.4.</u>

<u>REMARKS:</u> If $T$ is decidable then the set of Floyd-Hoare derivations (of pairs $(p,\psi)$ where $p\in P$ and $\psi\in L$, from $T$) is also decidable. If $T$ is recursively enumerable then the Floyd-Hoare derivable pairs are also recursively enumerable, i.e. $\{(p,\psi) : T \overset{FH}{\vdash} (p,\psi)\}$ is recursively enumerable.

## 4. COMPLETENESS

Notation: $\text{Mod}(T) \overset{d}{=} \{\mathfrak{A} : \mathfrak{A} \models T\}$ for any $T\subseteq L$.

<u>THEOREM 1</u> Let $T\supseteq PA$ be arbitrary. Let further $p\in P$ and $\psi\in L$ be also arbitrary. Then $T \overset{FH}{\vdash} (p,\psi)$ if and only if $(p,\psi)$ is partially correct in every model of $T$ w.r.t. continuous traces.

In concise form:

$$T \overset{FH}{\vdash} (p,\psi) \iff (\forall \mathfrak{A}\in\text{Mod}(T))\, \mathfrak{A} \overset{pc}{\models} (p,\psi).$$

<u>Proof.</u> The proof can be found in [3] which appeared in MFCS'81 pp. 162-171. <u>QED</u>

The condition $T \supseteq PA$ can be eliminated from the above theorem. Moreover, the restriction that $t$ is the similarity type of $PA$ can be eliminated, too. This generalization of Thm.1 above is Thm.9 of [4] on p.56 there (see also Prop.12 there), and it is also stated in Part II of [7] which is available in the literature. A somewhat modified version of this general theorem is Thm.3.3 of [13].

A drawback of our present approach is that the meanings of programs in $\mathfrak{A}$ are continuous traces and that these continuous traces are not elements of $\mathfrak{A}$, they are just functions $s : A \to A$ satisfying certain axioms formulated in the metalanguage (and not in $L$). This drawback is completely eliminated in the approach of [4], and of [7]. There the meanings of programs in a model $\mathfrak{M}$ are elements of $\mathfrak{M}$ and all requirements are formulated in the subject language $L$, e.g. continuity of traces is formulated by a set $IA^q$ of formulas in $L$.

The present approach is also extended to treat total correctness in the quoted papers, see e.g. Thm.7 on p.51 of [4]. The generality of that approach enables one to investigate the lattice of logics of programs (or dynamic logics), see the figure on p.109 of [4], and for more results and detailed proofs in this direction see [19]. The proof methods in the quoted general works are similar to the model theoretic proofs in the book Henkin-Monk-Tarski-Andréka-Németi [18]. The algebraization of our general dynamic logic (of programs) yields $\mathrm{Crs}_\alpha$ -s defined in the quoted book.

## 5. AN EXAMPLE FOR NONSTANDARD TRACES

So far we restricted ourselves to models of Peano's arithmetic $PA$. Specially, our similarity type $t$ was required to contain the symbols $+,\cdot,0,1$ with arities $2,2,0,0$ respectively. However, all what we really used in our definitions, e.g. in the definition of continuous traces, was $0$ and succ where succ is the successor function.

Let the similarity type $d$ consist of the symbols $0$, succ, pred of arities $0,1,1$ only. Here succ is the successor and pred is the predecessor, i.e. the standard model of type $d$ is $\underset{\sim}{\omega} \overset{d}{=}$ $\overset{d}{=} \langle \omega, 0, \mathrm{succ}, \mathrm{pred} \rangle$ where $(\forall n \in \omega)[\mathrm{succ}(n) = n+1$ and $\mathrm{pred}(n+1) = n]$ and $\mathrm{pred}(0) = 0$. Let $Pa \overset{d}{=} \{\varphi \in L_d : \underset{\sim}{\omega} \models \varphi\}$. It is well known that $Pa$ is decidable. Of course $PA \models Pa$.

In the present section we shall use $Pa$ instead of $PA$ and $d$

instead of $t$. Our aims with this change are simplicity and better understanding of the basic methods underlying the so called nonstandard time semantics approach.

DEFINITION 5 The definition of continuous traces of programs $p \in P_d$ in models of $Pa$ should be clear, namely replace in Definitions 1,2 the statements "Let $\mathfrak{A} \models PA$" by "Let $\mathfrak{A} \models Pa$" and replace $a+1$ everywhere by $\mathrm{succ}(a)$. End of Def.5.

PROPOSITION 2 Let $T \subseteq L_d$ and assume $T \supseteq Pa$. Let $p \in P_d$ and $\psi \in L_d$ be arbitrary. Assume $T \overset{FH}{\vdash} (p,\psi)$. Then $(p,\psi)$ is partially correct in every model of $T$ w.r.t. continuous traces. In concise form:

$T \overset{FH}{\vdash} (p,\psi) \Rightarrow (\forall \mathfrak{A} \in \mathrm{Mod}(T))\ \mathfrak{A} \overset{pc}{\models} (p,\psi)$.

Proof. Let $T \supseteq Pa$, $p = \langle (i_0 : u_0), \dots, (i_n : u_n) \rangle \in P_d$ and $V_p = \{y_1, \dots, y_k\}$. Assume $T \overset{FH}{\vdash} (p,\psi)$. We want to show partial correctness of $(p,\psi)$ w.r.t. continuous traces in models of $T$.

Let $\mathfrak{A} \models T$ and let $\langle s_a \rangle_{a \in A}$ be a continuous trace of $p$ in $\mathfrak{A}$. Let $\langle \Phi_z \rangle_{z \in \mathrm{lab}(p)} = \Phi : \mathrm{lab}(p) \to L$ belong to a Floyd-Hoare derivation of $(p,\psi)$ from $T$. Recall that $y_1, \dots, y_k$ are the variables occurring in $p$. Therefore we may use $y_0$ as "control variable" (i.e. for $\lambda$).

Define

$$\Phi(y_0, y_1, \dots, y_k) \overset{d}{=} \bigwedge_{m=1}^{n} (y_0 = i_m \to \Phi_{i_m}(y_1, \dots, y_k)) \wedge$$

$$\wedge \left( \left( \bigwedge_{m=1}^{n} y_0 \neq i_m \right) \to \psi(y_1, \dots, y_k) \right).$$

Now $\Phi \in L$ and $\mathfrak{A} \models \Phi[s_0] \wedge \bigwedge_{a \in A} (\Phi[s_a] \to \Phi[s_{\mathrm{succ}(a)}])$. (This is true because $\Phi : \mathrm{lab}(p) \to L$ belongs to a Floyd-Hoare derivation of $(p,\psi)$ and $\langle s_a \rangle_{a \in A}$ is a trace of $p$ in $\mathfrak{A}$.)

Since $\langle s_a \rangle_{a \in A}$ is, in addition, continuous, $\mathfrak{A} \models \bigwedge_{a \in A} \Phi[s_a]$. Let $a \in A$ be such that $s_a(\lambda) \notin \{i_m : m \leq n\}$. Then $\mathfrak{A} \models \Phi[s_a]$ implies $\mathfrak{A} \models \psi[s_a]$, by the definition of $\Phi$. This means $\mathfrak{A} \overset{pc}{\models} (p,\psi)$ since $\langle s_a \rangle_{a \in A}$ was an arbitrary continuous trace of $p$ in $\mathfrak{A}$.

We did this proof for programs $p$ satisfying $V_p = \{y_1, \dots, y_k\}$. Note that this does not restrict generality. QED

Proposition 2 above shows that it is useful to construct continuous traces of programs in models of $Pa$, too, since if the output of a continuous trace of the program $p \in P_d$ in a possibly nonstandard model

$\mathfrak{A} \models Pa$ does not satisfy the output condition $\psi$ then $Pa \not\vdash^{FH} (p,\psi)$ i.e. then the partial correctness of $(p,\psi)$ is not Floyd provable from $Pa$.

EXAMPLE

Let the similarity type $d$ consist of the symbols 0,succ,pred of arities 0,1,1.

1.
We define the program $p \in P_d$ by the block-diagram on Figure 2.

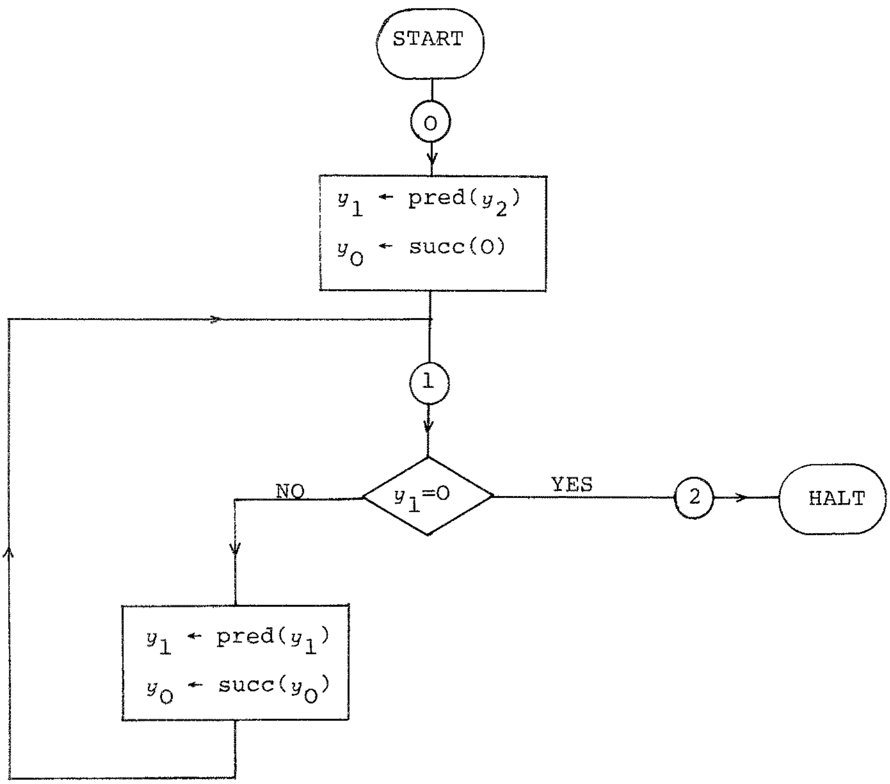


FIGURE 2

Clearly $p \in P_d$. Let $\psi$ be $y_0 = y_2$. Then $\psi \in L_d$ and the free variables of $\psi$ are in $V_p$. We shall call $\psi$ the *output condition,* because we shall consider partial correctness of the statement $(p,\psi)$.

2.

Next we construct a nonstandard model $\mathfrak{A}$ of our simplified number theory $Pa$.

$Z$ denotes the set of integers, i.e. $Z \stackrel{d}{=} \omega \cup \{-n : 0<n\in\omega\}$. We define $A \stackrel{d}{=} (\{0\}\times\omega)\cup(\{1\}\times Z)$. See Figure 3.

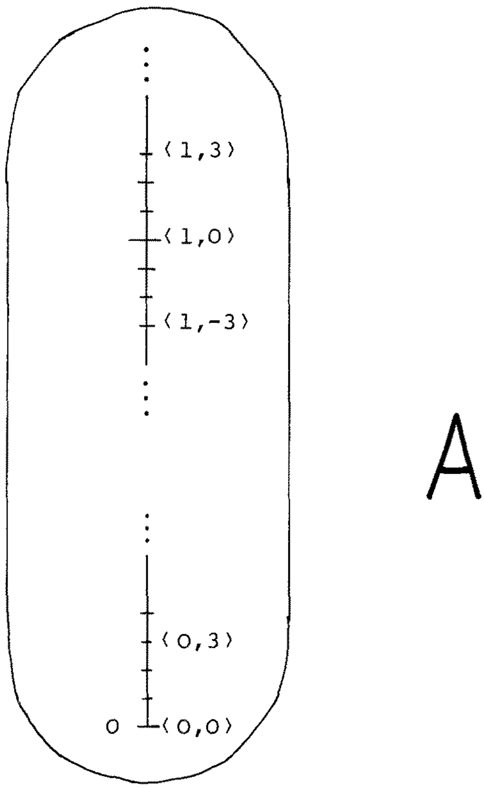


FIGURE 3

Now we define a model $\mathfrak{A}$ of similarity type $d$ such that the universe of $\mathfrak{A}$ is $A$ defined above, and the function symbols succ, pred,0 are defined on $A$ as follows:

Let $\langle a,b\rangle\in A$. Then $\mathrm{succ}(\langle a,b\rangle) \stackrel{d}{=} \langle a,b+1\rangle$, $\mathrm{pred}(\langle a,b+1\rangle) \stackrel{d}{=} \langle a,b\rangle$, $\mathrm{pred}(\langle 0,0\rangle) \stackrel{d}{=} \langle 0,0\rangle$. We define 0 of $\mathfrak{A}$ to be $\langle 0,0\rangle$.

We shall call $\langle 0,b\rangle\in A$ a standard number and $\langle 1,b\rangle$ a *nonstandard* number.

3.

Next we construct a continuous trace $f$ of $p$ in $\mathfrak{A}$.

Recall from Definition 1 that a trace of $p$ in $\mathfrak{A}$ is a sequence $s = \langle s_a \rangle_{a\in A}$ of valuations $s_a : \{y_0, y_1, y_2, \lambda\} \to A$ of the variables where $\lambda$ is the control variable. We shall identify this sequence $s$ with a 4-tuple $\langle f_0, f_1, f_2, f_\lambda \rangle$ of sequences $f_i : A \to A$ such that $f_0 \stackrel{d}{=} \langle s_a(y_0) : a\in A \rangle, \ldots, f_\lambda \stackrel{d}{=} \langle s_a(\lambda) : a\in A \rangle$. Clearly $f_i : A \to A$ for $i\in\{0,1,2,\lambda\}$.

Note that we can consider $f_i$ to be the history of the content of the program variable $y_i$ during execution of $p$ i.e. during time. For $a\in A$ we can say that $f_i(a)$ is the content of the variable $y_i$ at time point $a$.

Now let $f_0 : A \to A$ and $f_1 : A \to A$ be as indicated on Figure 4. See also Figure 5.
That is:

$$f_0(\langle a_0, a_1 \rangle) \stackrel{d}{=} \begin{cases} \langle a_0, a_1 \rangle & \text{if } a_0=0 \text{ or } a_1<0 \\ \langle 1,0 \rangle & \text{otherwise,} \end{cases}$$

$$f_1(\langle a_0, a_1 \rangle) \stackrel{d}{=} \begin{cases} \langle 1, -a_1 \rangle & \text{if } a_0=0 \\ \langle 0, -a_1 \rangle & \text{if } a_0=1 \text{ and } a_1<0 \\ \langle 0,0 \rangle & \text{if } a_0=1 \text{ and } a_1\geq 0. \end{cases}$$

We define

$f_2(a) \stackrel{d}{=} \langle 1,0 \rangle$ for every $a\in A$, and

$$f_\lambda(\langle a_0, a_1 \rangle) \stackrel{d}{=} \begin{cases} \langle 0,0 \rangle & \text{if } a_0=a_1=0 \\ \langle 0,1 \rangle & \text{if } (a_0=0,\ a_1>0) \text{ or } (a_0=1,\ a_1<0) \\ \langle 0,2 \rangle & \text{if } a_0=1 \text{ and } a_1\geq 0. \end{cases}$$

Now it is easy to see that $f \stackrel{d}{=} \langle f_0, f_1, f_2, f_\lambda \rangle$ is a trace of $p$ in $\mathfrak{A}$, continuity of which will be proved below.

4.
PROPOSITION 3 Let $d$, $p\in P_d$, $\mathfrak{A}$, $f$ be as defined above. Then $f$ is a continuous trace of $p$ in $\mathfrak{A}$.

Proof. For every $a\in A$ let $s_a : V_p \cup \{\lambda\} \to A$ be the valuation of the variables of $p$ into $\mathfrak{A}$ be defined by $s_a(y_i) \stackrel{d}{=} f_i(a)$ for $i\in\{0,1,2\}$ and $s_a(\lambda) \stackrel{d}{=} f_\lambda(a)$. According to our convention made earlier, we identify the sequence $\langle s_a \rangle_{a\in A}$ with $f$ and therefore we shall say that we want to prove that $f$ is a continuous trace of $p$ in $\mathfrak{A}$.

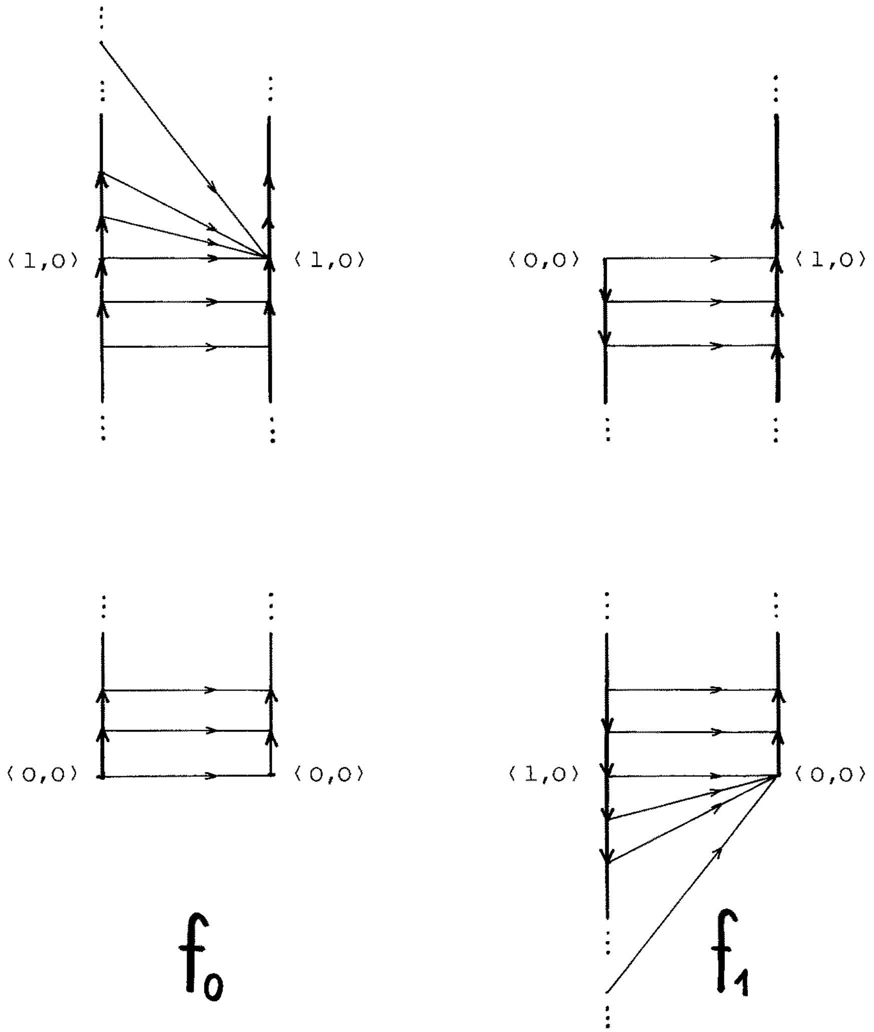


FIGURE 4

We shall use the following *notation*: Let $\varphi \in L_d$ with free variables in $V_p \cup \{\lambda\}$ and let $a \in A$. Therefore it is meaningful to write that $\mathfrak{A} \models \varphi[s_a]$ because $s_a : V_p \cup \{\lambda\} \rightarrow A$.

$\bar{f}(a)$ denotes the sequence $\langle f_0(a), f_1(a), f_2(a), f_\lambda(a) \rangle$. We define $\mathfrak{A} \models \varphi[\bar{f}(a)]$ to mean that $\mathfrak{A} \models \varphi[s_a]$.

In order to prove that $f$ is a continuous trace of $p$ in $\mathfrak{A}$, it is enough to prove for every $\varphi(y_0, y_1, y_2, \lambda) \in L_d$ that

$$\mathfrak{A} \models ((\varphi[\bar{f}(0)] \wedge \bigwedge_{a \in A} (\varphi[\bar{f}(a)] \rightarrow \varphi[\bar{f}(\mathrm{succ}(a))])) \rightarrow \bigwedge_{a \in A} \varphi[\bar{f}(a)]).$$

We shall prove this indirectly:

*Assume* that there is a $\varphi(y_0,y_1,y_2,\lambda)\in L_d$ such that

$$\mathfrak{A} \models (\varphi[\bar{f}(0)]\wedge \bigwedge_{a\in A} (\varphi[\bar{f}(a)] \to \varphi[\bar{f}(\mathrm{succ}(a))])) \tag{1}$$

but $\mathfrak{A} \not\models \bigwedge_{a\in A} \varphi[\bar{f}(a)]$ i.e. there is an $a\in A$ such that

$$\mathfrak{A} \models \neg\varphi[\bar{f}(a)]. \tag{2}$$

Recall that $\bar{f} : A \to {}^4A$ is as represented on Figure 5, i.e.: for every $0<n\in\omega$ we have

$\bar{f}(\langle 0,n\rangle) = \langle\langle 0,n\rangle,\langle 1,-n\rangle,\langle 1,0\rangle,\langle 0,1\rangle\rangle$,

$\bar{f}(\langle 1,-n\rangle) = \langle\langle 1,-n\rangle,\langle 0,n\rangle,\langle 1,0\rangle,\langle 0,1\rangle\rangle$,

$\bar{f}(\langle 0,0\rangle) = \langle\langle 0,0\rangle,\langle 1,0\rangle,\langle 1,0\rangle,\langle 0,0\rangle\rangle$,

$\bar{f}(\langle 1,0\rangle) = \langle\langle 1,0\rangle,\langle 0,0\rangle,\langle 1,0\rangle,\langle 0,2\rangle\rangle$.

Note that $f_2$ is a constant function and $f_\lambda$ is almost constant. By (1) we have $\mathfrak{A} \models \varphi[\bar{f}(a)]$ for every standard number $a\in A$, i.e. we have that $(\forall n\in\omega)\ \mathfrak{A} \models \varphi[\bar{f}(\langle 0,n\rangle)]$ i.e.

$$\mathfrak{A} \models \varphi[\langle 0,n\rangle,\langle 1,-n\rangle,\langle 1,0\rangle,\langle 0,1\rangle] \text{ for all } 0<n\in\omega. \tag{3}$$

(See Figure 5.) Then $a$ is a nonstandard number in (2) i.e. there is a $z\in Z$ such that $\mathfrak{A} \models \neg\varphi[\bar{f}(\langle 1,z\rangle)]$. (See Figure 6.)

Then, by (1), we have that $(\forall w\le z)\ \mathfrak{A} \models \neg\varphi[\bar{f}(\langle 1,w\rangle)]$. Then there is an $m\in\omega$ such that $(\forall n>m)\ \mathfrak{A} \models \neg\varphi[\bar{f}(\langle 1,-n\rangle)]$, i.e.

$$(\forall n>m)\ \mathfrak{A} \models \neg\varphi[\langle 1,-n\rangle,\langle 0,n\rangle,\langle 1,0\rangle,\langle 0,1\rangle]. \tag{4}$$

(See Figure 5.)

Let $F$ be a nonprincipal ultrafilter over $\omega$. $\mathfrak{B}$ denotes the ultrapower ${}^\omega\mathfrak{A}/_F$. For $\mathfrak{B}$ see Figure 9. We define

$b \stackrel{d}{=} \langle\langle 0,n\rangle : n\in\omega\rangle/_F$,

$c \stackrel{d}{=} \langle\langle 1,-n\rangle : n\in\omega\rangle/_F$,

$d \stackrel{d}{=} \langle\langle 1,0\rangle : n\in\omega\rangle/_F$,

$e \stackrel{d}{=} \langle\langle 0,1\rangle : n\in\omega\rangle/_F$.

Clearly $b,c,d,e\in B$. Then, by Łoś lemma and (3) we have

$$\mathfrak{B} \models \varphi[b,c,d,e]. \tag{5}$$

By Łoś lemma and (4) we have

$$\mathfrak{B} \models \neg\varphi[c,b,d,e]. \tag{6}$$

We define $\mathrm{succ}^n$ for $n\in\omega$ as: $\mathrm{succ}^0(g) \stackrel{d}{=} g$ and $\mathrm{succ}^{n+1}(g) \stackrel{d}{=}$ $\stackrel{d}{=} \mathrm{succ}(\mathrm{succ}^n(g))$ for every $g\in B$. $\mathrm{pred}^n$ is defined similarly to $\mathrm{succ}^n$.

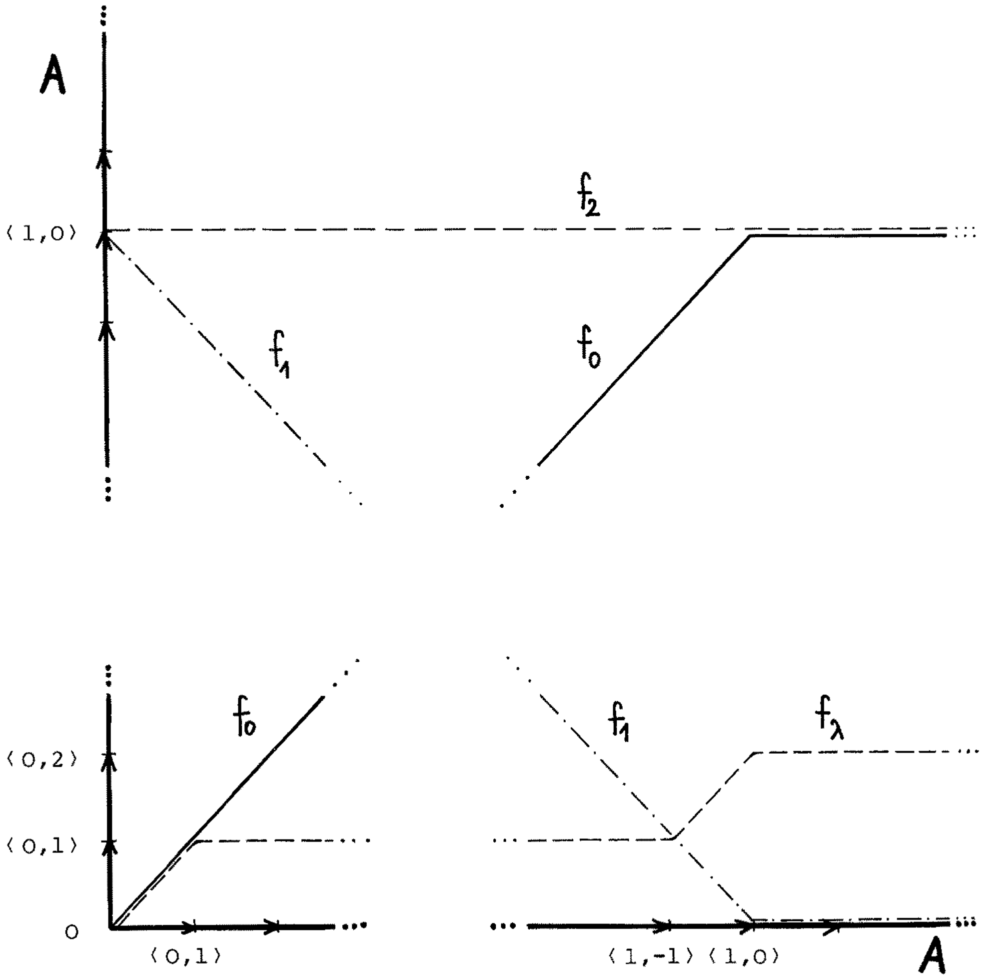


FIGURE 5

Clearly, $\mathcal{B}$ contains the following 2 "chains" $Y,W$ illustrated on Figure 7. See also Figure 8.

More precisely, $Y$ and $W$ are the two subalgebras of $\langle B,\mathrm{succ},\mathrm{pred}\rangle$ generated by the elements $b$ and $c$ respectively. Then $b\in Y$ and $c\in W$ and $Y$ is the smallest subset of $B$ closed under succ and pred and containing $b$. Similarly for $W$ and $c$. Then

$$\langle Y,\mathrm{succ},\mathrm{pred}\rangle \cong \langle W,\mathrm{succ},\mathrm{pred}\rangle.$$

Let $k : \langle Y,\mathrm{succ},\mathrm{pred}\rangle \to \langle W,\mathrm{succ},\mathrm{pred}\rangle$ be an isomorphism such that $k(b) = c$. For any $g\in B$ we define

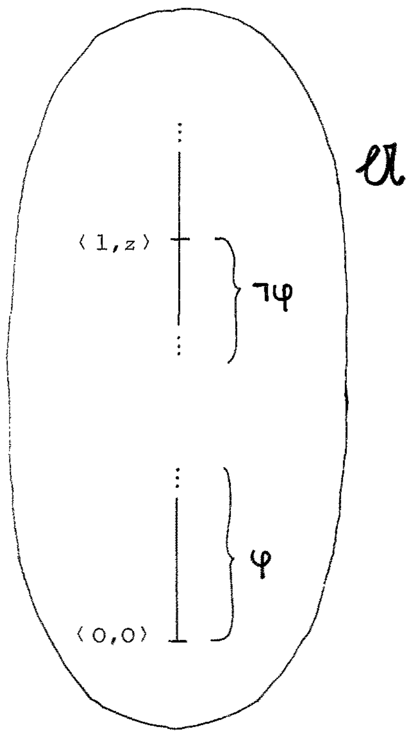


FIGURE 6

$$h(g) \overset{d}{=} \begin{cases} k(g) & \text{if } g \in Y \\ k^{-1}(g) & \text{if } g \in W \\ g & \text{otherwise.} \end{cases}$$

Then $h : B \to B$ is a function. Moreover, $h : \mathfrak{B} \to \mathfrak{B}$ is an automorphism of $\mathfrak{B}$. See Figure 9.
Therefore by (5) we have $\mathfrak{B} \models \varphi[h(b),h(c),h(d),h(e)]$, that is $\mathfrak{B} \models$ $\models \varphi[c,b,d,e]$. But this contradicts (6).

We derived a contradiction from the assumption (2). This proves $\mathfrak{A} \models \bigwedge_{a \in A} \varphi[\bar{f}(a)]$. This completes the proof. <u>QED(Prop.3)</u>

Clearly the "halting point" of the trace $f$ of $p$ in $\mathfrak{A}$ is the time point $\langle 1,0\rangle$. $f_0(\langle 1,0\rangle) = \langle 1,0\rangle$ and $f_2(\langle 1,0\rangle) = \langle 1,0\rangle$. Therefore the output condition $y_0=y_2$ of $p$ is satisfied by trace $f$ of $p$ in $\mathfrak{A}$.

<u>5.</u>
Let $f_0' : A \to A$ differ from $f_0$ only on the nonstandard numbers and such that if $a \in A$ is nonstandard then $f_0'(a) = \text{succ}(f_0(a))$. In more

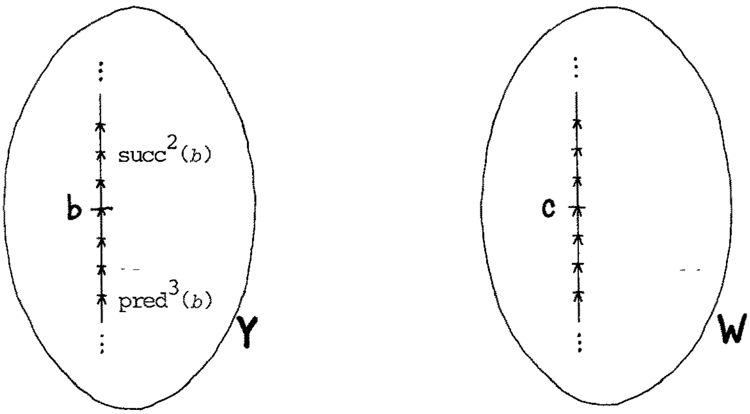


FIGURE 7

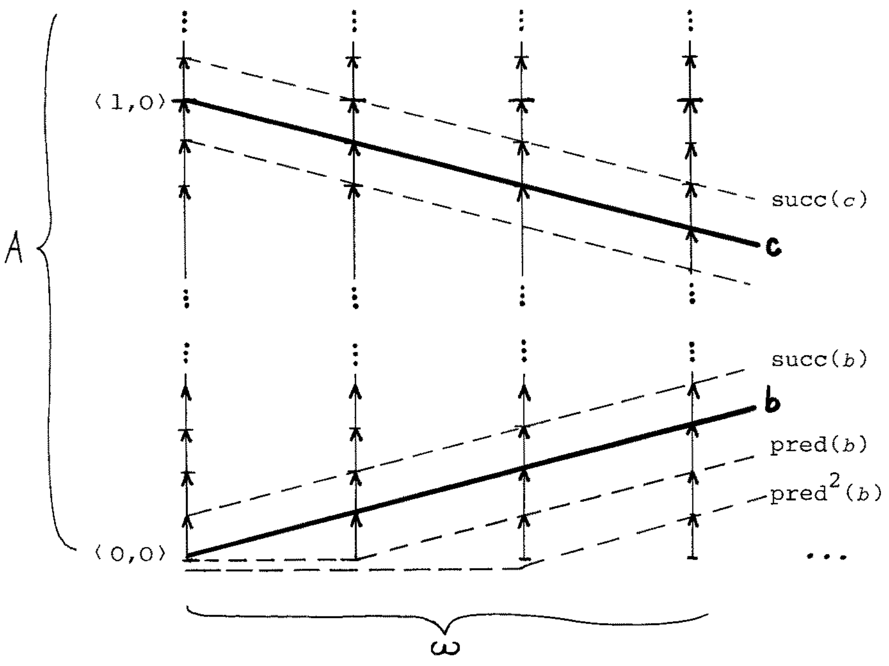


FIGURE 8

detail:

$(\forall n\in\omega)[f_0'(\langle 0,n\rangle) = \langle 0,n\rangle,\ f_0'(\langle 1,-n\rangle) = \langle 1,(-n)+1\rangle,\ f_0'(\langle 1,n\rangle) = \langle 1,n+1\rangle]$.

Let $f' \overset{d}{=} \langle f_0', f_1, f_2, f_\lambda\rangle$. By the constructions in the proof of Proposition 3 it is very easy to see that $f'$ is continuous e.g. by using

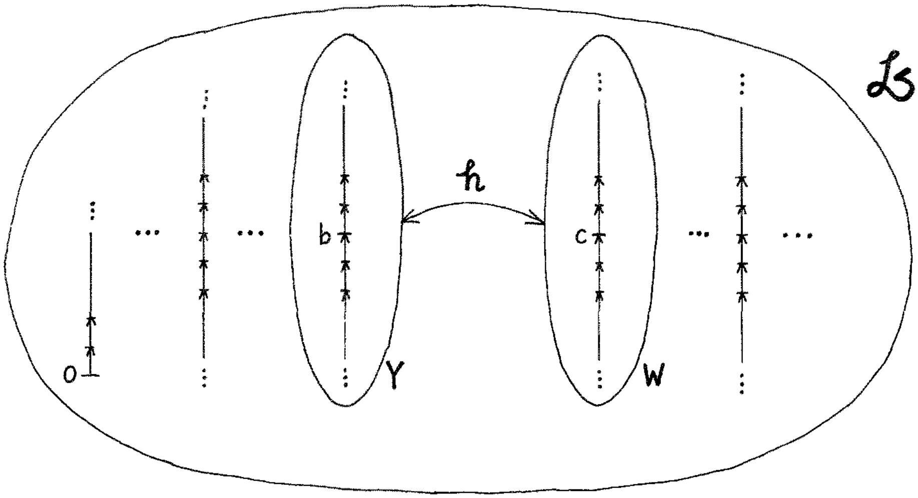


FIGURE 9

the fact that if we define

$$k(g) \overset{d}{=} \begin{cases} \mathrm{succ}(g) & \text{if } g \in W \\ g & \text{otherwise} \end{cases}$$

for all $g \in B$ then $k$ is an automorphism of $\mathcal{L}$ .
The trace $f'$ of $p$ terminates (at time point $\langle 1,1 \rangle$) with output $y_0 = \langle 1,1 \rangle$, $y_1 = \langle 0,0 \rangle$, $y_2 = \langle 1,0 \rangle$. Then for this output $y_0 \neq y_2$!
Then in the sense of Definition 3 we have

$\mathfrak{A} \not\models^{pc} (p, y_0 = y_2)$.

6.
By Proposition 2 we have

$\{\varphi \in L_d : \mathfrak{A} \models \varphi\} \not\vdash^{FH} (p,\psi)$ and also $Pa \not\vdash^{FH} (p,\psi)$

since the continuous trace $f'$ of $p$ in $\mathfrak{A}$ terminates with an output not satisfying $\psi$.

7.
Let $d'$ be the similarity type $d$ expanded with the relation symbol $<$ with arity 2. We interpret $<$ in $\mathfrak{A}$ the lexicographical way i.e.: for every $\langle a_0, a_1 \rangle, \langle b_0, b_1 \rangle \in A$,

$\langle a_0, a_1 \rangle < \langle b_0, b_1 \rangle$ iff either $a_0 < b_0$ or ($a_0 = b_0$ and $a_1 < b_1$).

Let $\mathfrak{A}'$ be the model $\mathfrak{A}$ expanded with the relation $<$ defined above. Then $\mathfrak{A}'$ is a model of similarity type $d'$. Let $f_i : A \to A$, $i \in \{0,1,2,\lambda\}$ be the functions defined in 3. above. Then $f = \langle f_0, f_1, f_2, f_\lambda \rangle$ is a trace of $p$ in $\mathfrak{A}'$. But the trace $f$ of $p$ is not continuous in $\mathfrak{A}'$!

REFERENCES

[1] Andréka,H. Németi,I.: On the completeness problem of systems for program verification. (In Hungarian.) Math.Inst.Hung.Acad.Sci.--SZKI, 1977.

[2] Andréka,H. Németi I.: Completeness of Floyd Logic. Bull.Section of Logic (Wrowlaw), Vol 7, No 3, (1978), pp.115-120.

[3] Andréka,H. Németi,I.: A characterization of Floyd provable programs. Preprint, Math.Inst.Hung.Acad.Sci. No 8/1978. In: Mathematical Foundations of Computer Science'81 (Proc. Strbské Pleso Czechoslovakia) Ed.: J.Gruska, M.Chytil, Lecture Notes in Computer Science 118, Springer, Berlin, 1981. pp.162-171.

[4] Andréka,H. Németi I.: A complete first order dynamic logic. Preprint, Math.Inst.Hung.Acad.Sci., No 810930,1980. pp.1-120.

[5] Andréka,H. Németi,I. Sain,I.: Completeness problems in verification of programs and program schemes. Mathematical Foundations of Computer Science MFCS'79 (Proc. Olomouc, Czechoslovakia) Lecture Notes in Computer Science 74, Springer, Berlin, 1979. pp.208-218.

[6] Andréka,H. Németi,I. Sain,I.: Henkin-type semantics for program schemes to turn negative results to positive. In: Fundamentals of Computation Theory FCT'79 (Proc. Berlin) Ed.: L.Budach, Akademie, Berlin. 1979. pp.18-24.

[7] Andréka,H. Németi,I. Sain,I.: A complete logic for reasoning about programs via nonstandard model theory. Theoretical Computer Science Vol 17 (1982) Part I in No 2, Part II in No 3. To appear.

[8] Bell,J.L. Slomson,A.B.: Models and Ultraproducts. North-Holland, 1969.

[9] Biró,B.: On the completeness of program verification methods. Bull. Section of Logic, Wroclaw, Vol 10, No 2 (1981), pp.83-90.

[10] Chang,C.C. Keiser,H.J.: Model Theory. North-Holland, 1973.

[11] Cook,S.A.: Soundness and completeness of an axiom system for program verification. SIAM J. COMPUT. Vol 7, No 1, 1978, pp.70-90.

[12] Csirmaz,L.: On definability in Peano Arithmetic. Bull.Section of Logic (Wroclaw), Vol 8, No 3, 1979. pp.148-153.

[13] Csirmaz,L.: Programs and program verifications in a general setting. Theoretical Computer Science Vol 16 (1981), pp.199-211.

[14] Csirmaz,L.: On the completeness of proving partial correctness. Acta Cybernetica, Tom 5, Fasc 2, Szeged, 1981. pp.181-190.

[15] Csirmaz,L. Paris,J.: A property of 2-sorted Peano models and

program verification. Preprint, Math.Inst.Hung.Acad.Sci., 1981.

[16] Dahn,B.I.: First order predicate logics for Kripke-Models. Dissertation(B). Humboldt Univ., Berlin, 1979.

[17] Hájek,P.: Making dynamic logic first-order. In: Mathematical Foundations of Computer Science MFCS'81 (Proc.Strbské Pleso, Czechoslovakia) Ed.: J.Gruska, M.Chytil, Lecture Notes in Computer Science 118, Springer, Berlin, 1981. pp.287-295.

[18] Henkin,L. Monk,J.D. Tarski,A. Andréka,H. Németi,I.: Cylindric Set Algebras. Lecture Notes in Mathematics 883, Springer-Verlag, Berlin, 1981. v+323p.

[19] Németi,I.: Nonstandard dynamic logic. In: Proc. Workshop on Logics of Programs (May 1981, New York) Ed.: D.Kozen, Lecture Notes in Computer Science, To appear.

[20] Richter,M.M. Szabo,M.E.: Towards a nonstandard analysis of programs. In: Proc. 2nd Victoria Symp. on Nonstandard Analysis (Victoria, British Columbia, June 1980) Lecture Notes in Mathematics. Ed.: A.Hurd, Springer Verlag, 1981.

[21] Sain,I.: There are general rules for specifying semantics: Observations on abstract model theory. CL & CL Budapest, Vol 13, 1979. pp.251-282.

[22] Sain,L.: First order dynamic logic with decidable proofs and workable model theory. In: Fundamentals of Computation Theory FCT'81, (Proc.Szeged, 1981) Ed.: F.Gécseg, Lecture Notes in Computer Science 117, Springer, Berlin, 1981. pp.334-340.

[23] Salwicki,A.: Axioms of Algorithmic Logic univocally determine semantics of programs. In: Mathematical Foundations of Computer Science MFCS'80 (Proc. Rydzyna, Poland) Lecture Notes in Computer Science 88, Springer, Berlin, 1980. pp.552-561.