COMPLETENESS PROBLEMS IN VERIFICATION OF PROGRAMS AND PROGRAM SCHEMES

H. Andréka
I. Németi
I. Sain

Mathematical Institute of the
Hungarian Academy of Sciences
Budapest, Reáltanoda u. 13-15
H-1053 Hungary

Abstract: Thm 1 states a negative result about the classical semantics $\models^{\omega}$ of program schemes. Thm 2 investigates the reason for this. We conclude that Thm 2 justifies the Henkin-type semantics $\models$ for which the opposite of the present Thm 1 was proved in Andréka-Németi[1], [2], [3] and also in a different form in part III of Gergely-Ury[8]. The strongest positive result on $\models$ is Corollary 6 in Andréka-Németi[3].

Basic concepts

First we recall some basic notions and notations from textbooks on Logic Monk[10], Chang-Keisler[5] and from Program Schemes Theory, e.g. Manna[9], Andréka-Németi[1], [2], [3], Gergely-Ury[8].

ω denotes the set of natural numbers.

d denotes an arbitrary similarity type. I.e.: d correlates arities to some fixed function symbols and relation symbols. See Sacks[12], p.11.

$Y = \{ y_z : z \in \omega \}$ denotes the set of variable symbols.

F_d is the set of all classical first order formulas of type d with variables in Y. See Chang-Keisler[5], p.22.

M_d is the class of all classical first order models of type d. See Chang-Keisler[5] or Monk[10], Def.11.1. or Sacks[12], p.11.

$\models \subseteq F_d \times M_d$ is the usual validity relation. See Chang-Keisler[5] or Sacks[12], p.21.

τ denotes a term of type d in the usual sense of first order logic, see Chang-Keisler[5], p.22 or Monk[10], p.166. Def.10.8. (ii).

$\underset{\sim}{D}$ and $\underset{\sim}{E}$ denote elements of M_d, the universes of which are D and E respectively.

P_d denotes the set of program schemes of type d. P_d is defined as in Manna[9], Andréka-Németi[1], [2], Gergely-Ury[8], p.72. E.g., let t be the similarity type of arithmetic. Then the following sequence is in P_t, i.e. it is a program scheme of type t:

$\langle$ (0: $y_0 \leftarrow 0$) ,
(1: IF $y_0 = y_1$ THEN 4) ,
(2: $y_0 \leftarrow y_0 + 1$) ,
(3: IF $y_1 = y_1$ THEN 1) ,
(4: HALT) $\rangle$.

$P_d \times F_d$ is the set of output statements about programs. An output statement $(p,\psi) \in P_d \times F_d$ means intuitively that the program scheme p is partially correct w.r.t. output condition ψ.

$\underset{\sim}{D} \overset{\omega}{\models} (p,\psi)$ is meaningful if $\underset{\sim}{D} \in M_d$ and $(p,\psi) \in P_d \times F_d$. Now $\underset{\sim}{D} \overset{\omega}{\models} (p,\psi)$ holds if the program scheme p is partially correct w.r.t. ψ in the model $\underset{\sim}{D}$. I.e.: If p is started in $\underset{\sim}{D}$ with any input $q:\omega \longrightarrow D$ then whenever p halts with some output $k:\omega \longrightarrow D$, the formula ψ will be true in $\underset{\sim}{D}$ under the valuation k of its free variables, i.e. $\underset{\sim}{D} \models \psi[k]$. See Manna [9], Chapter 4. Note that a precise definition of $\overset{\omega}{\models}$ would strongly use the structure $\langle \omega, \leq \rangle$ of natural numbers. See [14], Gergely-Ury[8], p.78, Andréka-Németi[1], p.116, [2], [3]. The letter ω above the sign $\overset{\omega}{\models}$ serves to remind us of this fact.

For any set $Th \subseteq F_d$ of formulas, " $Th \overset{\omega}{\models} (p,\psi)$ " is defined in the usual way:

$$Th \models^{\omega} (p,\psi) \quad \text{iff} \quad (\forall D \in M_d)\left[\underset{\sim}{D} \models Th \Rightarrow \underset{\sim}{D} \models^{\omega} (p,\psi)\right] \quad .$$

<u>PROPOSITION 0</u> :
Let the type d contain the similarity type of successor arithmetic $\langle \omega, s, 0 \rangle$. Let $Th \subseteq F_d$ be such that

$$Th \supseteq \{ s^z 0 \neq s^r 0 \;:\; z < r \in \omega \} \overset{d}{=} Th'$$

where $s^0 0 \overset{d}{=} 0$ and $s^{r+1} 0 \overset{d}{=} s s^r 0$ for $r \in \omega$.
Let $\underset{\sim}{E} \in M_d$ be an arbitrary but fixed model of Th' such that $\underset{\sim}{E} =$ $= \langle \omega, s, 0, \ldots \rangle$.
Suppose H is an arbitrary set such that

$$\{ (p,\psi) \;:\; Th \models^{\omega} (p,\psi) \} \supseteq H \supseteq$$

$$\{ (p,\psi) \;:\; Th \models^{\omega} (p,\psi) \text{ and } p \text{ terminates in } \underset{\sim}{E} \text{ for every input and } \psi \text{ is quantifier-free} \} .$$

Then H is <u>not</u> recursively enumerable.

<u>Proof</u>: The present proposition is a special case of Thm 1 to be formulated later.
<u>QED</u>

Now we turn to relax the conditions made on d and Th in the above proposition. I.e. we are going to generalize Proposition 0 .

<u>From now on</u> c and τ denote <u>arbitrary terms</u> of type d such that c contains no variable and τ contains one variable y_0 . To make this explicit, we write $\tau(y_0)$.

Notation: $\tau^0 \overset{d}{=} c$, and $\tau^{z+1} \overset{d}{=} \tau(\tau^z)$ for every $z \in \omega$.

Note that the terms τ^0 , τ^1 , ... , τ^z , ... contain no variable.

DEFINITION 1 :

$Th \subseteq F_d$ is said to be good if there exist terms c and $\tau(y_o)$ such that

$$Th \supseteq \{ \tau^z \neq \tau^r \ : \ z<r \in \omega \} \overset{d}{=} Th' .$$

Let $\underset{\sim}{E} \in M_d$ be an arbitrary model of Th' such that

$(\forall b \in E)(\exists z \in \omega) \left[\tau^z \text{ denotes } b \text{ in } \underset{\sim}{E} \right]$. Then we define

$$May(Th) \overset{d}{=} \{ (p,\psi) \in P_d \times F_d \ : \ Th \overset{\omega}{\models} (p,\psi) \} .$$

$Must(Th) \overset{d}{=} \{ (p,\psi) \in May(Th) :$ p terminates in $\underset{\sim}{E}$ for every input; and ψ is an atomic formula or the negation of an atomic formula such that $Th \vdash \exists y_o \psi$ $\}$.

Remark: To a fixed Th , Must(Th) is not unique since it may depend on the choice of c , $\tau(y_o)$, and $\underset{\sim}{E}$. This makes the following theorem even stronger since it will hold for any choice of c , τ , and $\underset{\sim}{E}$. Observe that Must(Th) is a reasonably small set of output statements since ψ contains no quantifiers, no "$\vee$" or "$\wedge$" and at the same time p is such that it terminates in $\underset{\sim}{E}$ for every input. Thus Must(Th) contains no tricky statement about the "halting problem" (since p has to terminate) and no "strange sentence" since ψ has to be simple; moreover, $\exists y_o \psi$ is provable from Th .

THEOREM 1 :

Let d be arbitrary and let $Th \subseteq F_d$ be good (in sense of Def.1.) and consistent. Let H be an arbitrary set such that

$$May(Th) \supseteq H \supseteq Must(Th) .$$

Then H is not recursively enumerable.

Proof: We shall treat the constant-term "c" as zero and $\tau(y_o)$ as the successor function. E.g. τ^z will be considered to be the name of the natural number $z \in \omega$. By using successor τ and zero c we can write programs "add" $\in P_d$ and "mult" $\in P_d$ for addition and multiplication. By using these programs, for an arbitrary Diophantine equation $e(y_2,\dots,y_m)$ we can write a program $\bar{e}\in P_d$ such that after having executed $\bar{e}$ we shall have $y_0=y_1$ iff $e(y_2,\dots,y_m)$ was true before starting $\bar{e}$.

Let p be an m-variable version of the program scheme given as an example at the beginning of this paper. Namely, p starts with

$(0{:}\ y_{m+1}\leftarrow c)$, $(1{:}\ \text{IF}\ \ y_2=y_{m+1}\ \ \text{THEN}\ \ 4)$, $(2{:}\ y_{m+1}\leftarrow y_{m+1})$,
$(3{:}\ \text{IF}\ \ \text{TRUE}\ \ \text{THEN}\ \ 1)$, $(4{:}\ y_{m+1}\leftarrow c)$, $(5{:}\ \text{IF}\ \ y_3=y_{m+1}\ \ \text{THEN}\ \ 8)$, ...

This program p terminates iff all the initial values of $y_2,\dots,y_m$ can be reached from "c" by finitely many applications of τ . Now, by writing $\bar{e}$ after p we obtain a program $p\bar{e}\in P_d$ which first checks whether $y_2,\dots,y_m$ can be reached from "c" by applications of τ and if yes then results $y_0=y_1$ if $e(y_2,\dots,y_m)$ was true, $y_0\neq y_1$ if $e(y_2,\dots,y_m)$ was false for the initial values. Now to each Diophantine equation $e(\bar{y})$ correlate $\bar{\bar{e}} = (p\bar{e},\ y_0\neq y_1)$.

Clearly $\bar{\bar{e}}\in P_d\times F_d$. Also $Th \overset{\omega}{\models} \bar{\bar{e}}$ iff e has no solution in the standard model $\langle\omega,+,\cdot,0,1\rangle$ of arithmetic. If there were a recursively enumerable H as in the statement of the present theorem then

$$Eq \overset{d}{=} \{\ e\in\text{"Diophantine equations"}\ :\ Th \overset{\omega}{\models} \bar{\bar{e}}\ \}$$

would be recursively enumerable since the construction of $\bar{\bar{e}}$ from e was "constructive" . But, since Hilbert's tenth problem is unsolvable (Davis[6] or Monk[10]), this is impossible.
<u>QED</u>

The following theorem says that if one "avoids Logic" and proves properties of programs by using "Mathematics in general" then this will <u>not help</u> one to avoid the "shortcoming" formulated in Thm 1 .

<u>THEOREM 2</u> :

Let the real world $\langle V,\in\rangle \models ZFC$ of Set Theory (see Devlin[7], p.3, line 4 from below or Chang-Keisler[5], p.476) be fixed. I.e.: V is the class of all sets and $\in$ is the "element of" relation between them.

Then the following is true:

There exist

- a similarity type d , and
- a model $\langle W,E\rangle \models ZFC$ of Set Theory inside of $\langle V,\in\rangle$ (i.e. $\langle W,E\rangle$ is an element of V and $\langle W,E\rangle \models ZFC$ is true inside of $\langle V,\in\rangle$, see Devlin[7], p.14, line 6)

such that (i) and (ii) below hold.

(i) There are a finite set $Th \subseteq F_d$ of axioms and an output statement (p,ψ) such that

$Th \overset{\omega}{\models} (p,\psi)$ is true, but inside of $\langle W,E\rangle$ we have

$Th \overset{\omega}{\not\models} (p,\psi)$.

More precisely:

$\langle V,\in\rangle \models$ " $Th \overset{\omega}{\models} (p,\psi)$ " but

$\langle W,E\rangle \models$ " $Th \overset{\omega}{\not\models} (p,\psi)$ " .

(Observe that " $Th \overset{\omega}{\models} (p,\psi)$ " is a statement of the language of ZFC .)

(ii) There is an output statement (p,ψ) such that

$\langle V,\in\rangle \models$ " $M_d \overset{\omega}{\models} (p,\psi)$ " while

$\langle W,E\rangle \models$ " $M_d \overset{\omega}{\not\models} (p,\psi)$ " .

As a <u>contrast</u> we note that:

For all $\varphi \in F_d$ and for every model $\langle W,E\rangle \in V$ of ZFC ,

$\langle V,\in\rangle \models$ " $M_d \models \varphi$ " implies $\langle W,E\rangle \models$ " $M_d \models \varphi$ " .

<u>Proof</u>:

(i)

Let $d \overset{d}{=} \{ \langle 0,0\rangle , \langle s,1\rangle \}$.

Let Th consist of the following two axioms:

$\forall y\,(sy \neq 0)$

$\forall y_1\ \forall y_2\ (sy_1 = sy_2 \rightarrow y_1 = y_2)$.

We know that Hilbert's tenth problem is unsolvable. This implies the existence of a Diophantine equation $e(\bar{y})$ such that the set theoretic formula

$$\text{"} \langle \omega, s, +, \cdot, 0, 1\rangle \models \exists \bar{y}\ e(\bar{y}) \text{"}$$

is false in $\langle V,\in\rangle$ but is true in $\langle W,E\rangle$ for some model $\langle W,E\rangle \in V$ of ZFC .

Now let the output statement $\bar{\bar{e}} = (p\bar{e}\ ,\ y_0 \neq y_1)$ be the one defined in the proof of Thm 1 . There it was observed that

$$Th \models^{\omega} \bar{\bar{e}} \qquad \text{iff} \qquad \langle \omega,+,\cdot,0,1 \rangle \not\models \exists \bar{y}\ e(\bar{y}) \quad .$$

(Note that the present Th satisfies the conditions of Thm 1 .)

Thus $\langle V,\epsilon \rangle \models$ " $Th \models^{\omega} \bar{\bar{e}}$ " and $\langle W,E \rangle \models$ " $Th \not\models^{\omega} \bar{\bar{e}}$ " .

(ii)

The proof of (ii) is an easy modification of the proof of (i) above. Namely, let us choose the above e , $\langle W,E \rangle$, and $\bar{\bar{e}} = (p\bar{e},\ y_0 \neq y_1)$. Let φ be the conjunction of all elements of Th . (Note that Th is finite and therefore $\varphi \in F_d$.) Let $\psi \stackrel{d}{=} (\varphi \rightarrow y_0 \neq y_1)$. Now,

$\langle V,\epsilon \rangle \models$ " $M_d \models^{\omega} (p\bar{e},\psi)$ " while

$\langle W,E \rangle \models$ " $M_d \not\models^{\omega} (p\bar{e},\psi)$ " .

<u>QED Thm 2</u> (For a more detailed proof cf. Andréka-Németi-Sain[4].)

The above Thm 2 says that something is wrong with the classical semantics (or model theory) $\models^{\omega}$ of program schemes. Namely: There exists a good program (p,ψ) which is <u>not provable</u> by mathematics, i.e. the goodness of (p,ψ) is not "a mathematical truth" i.e. it is not implied by ZFC despite of the fact that it happens to be the case that (p,ψ) <u>is</u> good. See Németi-Sain[11]Def.2 and Andréka-Németi-Sain[4] about " $Th \models^{\omega} (p,\psi)$ " -s being a formula of Set Theory. This way Thm 2 supports the Henkin-type semantics introduced in Andréka-Németi [1]-[3], the consequence concept $(Th \models (p,\psi))$ of which does <u>not</u> have the above shortcoming.

By Thm 2 above there exists an output statement (p,ψ) which is valid, i.e. " $\models^{\omega} (p,\psi)$ ", but the validity of which is <u>not</u> a mathematical truth, i.e. ZFC $\not\models$ " $\models^{\omega} (p,\psi)$ " . A semantics with this paradoxical property was called <u>instable</u> in Andréka-Németi-Sain[4]. It was proved in [4] that any "reasonable" semantics has to be stable. Indeed, the Henkin-type semantics introduced in Andréka-Németi[1]-[3] was proved to be <u>stable</u> there.

On basis of Thm 2 above an effective inference system for program correctness was given in Andréka-Németi-Sain[4] such that if (p,ψ) cannot be proved then there exists a model of ZFC Set Theory in which the program p <u>is actually not</u> correct w.r.t. ψ . Cf. Andréka-Németi [1]-[3], too.

A HENKIN TYPE SEMANTICS FOR PROGRAM SCHEMES

Now to every classical (one-sorted) similarity type d we define an associated <u>3-sorted similarity</u> type td . About many-sorted logic and its model theory see Monk[10], p.483.

As before, d is an arbitrary type. Let t denote the similarity type of <u>Peano Arithmetic</u> and let t be disjoint from d . The type td is defined as follows:

There are <u>3 sorts of td</u> : $\bar{t}$, $\bar{d}$, $\bar{i}$ called "time , "data" , and "intensions" respectively.

<u>The operation symbols of td</u> are the following: The operation symbols of t , the operation symbols of d , and an additional operation symbol "ext" .

<u>The sorts (or "arities") of the operation symbols of td</u> : The operation symbols of t go from sort $\bar{t}$ to sort $\bar{t}$. The operation symbols of d go from sort $\bar{d}$ to sort $\bar{d}$. The operation symbol "ext" goes from sort $(\bar{i},\bar{t})$ to sort $\bar{d}$.: I.e. "ext" has two arguments, the first is of sort $\bar{i}$, the second is of sort $\bar{t}$, and the result or value of "ext" is of sort $\bar{d}$. Now the definition of the 3-sorted type td is completed.

$TL_d = \langle TF_d , TM_d , \models \rangle$ denotes the 3-sorted language <u>of type td</u>, see Monk[10], p.483. In more detail:

(i) TM_d is the class of all <u>models of type td</u>, see Monk[10], Def.29.27. I.e. a model $\mathfrak{M} \in TM_d$ <u>has</u>

1. <u>three universes</u> throughout denoted by T , D , and I , of sort $\bar{t}$, $\bar{d}$, and $\bar{i}$ respectively.
2. Operations "$T^n \longrightarrow T$" originating from the type t , operations "$D^n \longrightarrow D$" originating from the type d , and an operation $ext: I \times T \longrightarrow D$.

Roughly speaking, we could say that $\mathfrak{M}$ consists of structures $\underset{\sim}{T} \in M_t$, $\underset{\sim}{D} \in M_d$, and an additional operation $ext: I \times T \longrightarrow D$.

Therefore we shall use the sloppy notation:

$$\mathfrak{M} \stackrel{d}{=} \langle \underset{\sim}{T}, \underset{\sim}{D}, I, ext \rangle \quad \text{for elements of } TM_d .$$

(ii) TF_d is the set of first order (3-sorted) formulas of type td. Roughly speaking, we can say that F_t and F_d are contained in TF_d, and there are additional terms of the form "$ext(y,\tau)$", where τ is a term of type t and y is a variable of sort $\bar{i}$. Further, "$ext(y,\tau)$" is defined to be a term of sort $\bar{d}$.

(iii) $\models \subseteq (TM_d \times TF_d)$ is the usual, see Monk[10], p.484.

Now we define the meanings of program schemes $p \in P_d$ in the 3-sorted models $\mathfrak{M} \in TM_d$. Let $p \in P_d$ be a fixed program scheme. Let $y_1, \dots, y_m$ be the variables occurring in p. Let $\mathfrak{M} \in TM_d$ be fixed. Recall that I is the universe of sort $\bar{i}$ of $\mathfrak{M}$.

A trace of p in $\mathfrak{M}$ is a sequence $\langle s_0, \dots, s_m \rangle \in {}^{(m+1)}I$ of elements of I satisfying (*) below. (I.e. a trace of p in $\mathfrak{M}$ is a sequence of i-sorted elements of $\mathfrak{M}$.) To formulate (*), observe that if $s \in I$ then "$ext(s,-)$" is a function $\langle ext(s,z) : z \in T \rangle$ from T into D. We shall use y_0 as "the control-variable" of p. I.e. $ext(s_0,z)$ is considered to be the "value of the control or execution" at time point z. Thus "$ext(s_0,z)$" is supposed to be a "label" in the program scheme p.

(*) The sequence $\langle ext(s_0,-), \dots, ext(s_m,-) \rangle$ of functions should be a history of an execution of p in $\underset{\sim}{D}$ along the "time axis" $\underset{\sim}{T}$.

The only difference from the classical definition (cf. Manna[9], Andréka-Németi[1]-[3], Gergely[14],[8]) of a trace of p in $\underset{\sim}{D}$ is that now the "time-axis" of execution is not necessarily $\langle \omega, s, +, \cdot, 0, 1 \rangle$ but, instead, it is $\underset{\sim}{T}$.
Condition (*) above can be made precise by replacing ω with $\underset{\sim}{T}$ in the classical definition, see Andréka-Németi[1]-[3], Gergely-Ury[8].

The trace $\langle s_0, \dots, s_m \rangle$ of p in $\mathfrak{M}$ terminates if $ext(s_0,z)$ is the label of the HALT statement, for some $z \in T$. If the trace $\langle s_0, \dots, s_m \rangle$ terminates at time $z \in T$ then its output is

$\langle ext(s_1,z), \ldots, ext(s_m,z) \rangle$. Now we define for $\psi \in F_d$:

$\mathfrak{M} \models (p,\psi)$ holds iff for every <u>terminating</u> trace of p in $\mathfrak{M}$ the output satisfies ψ in $\underset{\sim}{D}$. Cf. Def.8 of Szőts-Gergely [14], Andréka-Németi[1]-[3], and def. of $\overset{\omega}{\models}$ in the present paper.

For an arbitrary theory $Th \subseteq TF_d$ the consequence relation

$Th \models (p,\psi)$

is defined in the usual way.

<u>THEOREM 3 (Completeness of Programverification)</u> :

Let $Th \subseteq TF_d$ be recursively enumerable. Then the set

$$\{ (p,\psi) \in P_d \times F_d \quad : \quad Th \models (p,\psi) \}$$

of all its consequences is also recursively enumerable.

<u>Proof</u>: The proof can be found in Andréka-Németi-Sain[4]. Moreover, a complete inference system is explicitly given there, with decidable proof concept.
<u>QED</u>

To execute programs in arbitrary elements of TM_d might look counter--intuitive. However, we may require Th to contain the Peano Axioms for the sort $\bar{t}$ and some Induction Axioms for the sort $\bar{i}$. The set of these axioms was denoted by Ax in Andréka-Németi[3]. The induction axioms for $\bar{i}$ are of the kind:

$$\forall y \left[\left(\varphi(ext(y,0)) \wedge \forall z \left[\varphi(ext(y,z)) \rightarrow \varphi(ext(y,z+1)) \right] \right) \rightarrow \forall z\ \varphi(ext(y,z)) \right],$$

for every $\varphi(x) \in F_d$. Now the models $\mathfrak{M} \in TM_d$ of Ax do satisfy all the intuitive requirements about <u>time</u> and about <u>processes</u> "happening in time" .

PROPOSITION 4 :

Let $Th \supseteq Ax$ be a subset of TF_d . Suppose that (p,ψ) is Floyd--Hoare provable from Th .
Then $Th \models (p,\psi)$.

Proof is in Andréka-Németi[3].
QED

REFERENCES

1. Andréka, H. and Németi, I., Completeness of Floyd Logic. Bulletin of Section of Logic 7(1978), 115-121, Wrocław.
2. Andréka, H. and Németi, I., A characterization of Floyd provable programs. Submitted to Proc.Coll.Logic in Programming, Salgótarján 1978. Coll.Math.Soc.J.Bolyai, North Holland.
3. Andréka, H. and Németi, I., Classical many-sorted model theory to turn negative results on program schemes to positive. Preprint 1978.
4. Andréka, H., Németi, I., and Sain, I., Abstract model theory, semantics, logics. Preprint 1979.
5. Chang, C.C. and Keisler, H.J., Model Theory. North Holland, 1973.
6. Davis, M., Hilbert's tenth problem is unsolvable. Amer.Math.Monthly 80(1973), 233-269.
7. Devlin, K.J., Aspects of Constructibility. Lecture Notes in Math. 354, Springer Verlag, 1973.
8. Gergely, T. and Ury, L., Mathematical Programming Theories.
9. Manna, Z., Mathematical Theory of Computation. McGraw Hill, 1974.
10. Monk, J.D., Mathematical Logic. Springer Verlag, 1976.
11. Németi, I. and Sain, I., Connections between Algebraic Logic and Initial Algebra Semantics of CF Languages. Submitted to Proc.Coll. Logic in Programming, cf.[2].
12. Sacks, G.E., Saturated Model Theory. W.A.Benjamin, Inc. Publ., Reading, Massachusetts, 1972.
13. Sain, I., On the General Theory of Semantics of Languages. Preprint 1979.
14. Szőts, M. and Gergely, T., On the incompleteness of proving partial correctness. Acta Cybernetica Tom 4, Fasc 1, Szeged 1978, pp.45-57.