A COMPLETE FIRST ORDER DYNAMIC LOGIC

by

H. Andréka, I. Németi, I. Sain

No.810930

C O N T E N T S

Math.Inst.Hung.Acad.Sci.
Budapest, Reáltanoda u. 13-15, Hungary, H-1053

INTRODUCTION

In Computer Science and related fields there have been around logical systems in which reasoning about consequences of actions is also possible. I.e. in addition to being able to say "All humans are mortal.", "Sokrates is human." etc. we are also allowed to say "After throwing the switch there will be light." or "After touching the hot stove there will be pain." . The new patterns of thought appearing in these logics are of the kind "After doing action p it will be the case that φ ." where φ is a formula of classical logic. These patterns of thought are at the very core of human reasoning and hence such logics have appeared not only in Computer Science but also in e.g. Child Psychology , Developmental Psychology , Linguistics , Philosophical Logic . See e.g. Segerberg[45].

Dynamic logic is intended to be the common backbone of all these and related logics, and accordingly its aim is to find that basic structure which is common in all these logics, that basic structure which makes all of them tick. See e.g. Pratt[42],[45],[9], Harel[31].

Propositional Dynamic Logic is flourishing in a healthy, convincing and rather attractive way, its model theory is a clear Kripke style one [40],[31] which fits beautifully into the system of model theories of well understood logics. Propositional Dynamic Logic is developed coherently along the lines of the well-established General Methodology for doing Nonclassical Propositional Logic, see e.g. Segerberg[45]. Specially the proof concept of Propositional Dynamic Logic is a decidable one and the set of valid formulas is recursively enumerable, see e.g. Pratt[39], Segerberg[44],[38]. The notion of a proof concept and the property of its being decidable or not can be found in Definition 10 of the present paper (in §4). In short, a proof concept is decidable if the set of all correct proofs is decidable. (E.g. the proof concept of classical first order logic is decidable.) All the published completeness theorems for Propositional Dynamic Logic are based on decidable proof concepts and therefore are completely satisfactory [38],[44], Berman[8],[39].

First order Dynamic Logic (see e.g. Harel[31]) is in a state which is not nearly as satisfactory. There are several different alternative model theories around, these are fairly ad-hoc and the proof concepts used are not decidable, and most of the proposed model theories are such that the set of the valid formulas is not recursively enumerable. We shall refer to the semantics (or model theory) used in Harel[31] as standard (because it refers implicitly to the standard model of arithmetic as it was pointed out in Gergely-Úry[28],[29]).
First we tried to select a very small sublanguage S of First order Dynamic Logic such that an acceptable completeness theorem could be proved for S (at least). It turned out in Andréka-Németi-Sain[5] that this is impossible with standard semantics even if we are extremally permissive about the choice of S. To alleviate this problem, the standard literature started to use undecidable proof concepts, e.g.: infinitely long proofs (§3.4.2 in [31]), the so called Effective ω-rule (which is not decidable either), the so called Arithmetical Axiomatization of Harel[31] etc. However, there is a nonstandard literature too, (e.g. Gergely-Úry[28-30], Csirmaz[16-21], Andréka-Németi[1-7], Sain[43], Bíró[10], [37a-c],[42a],[30a] etc.) which reacted differently to the negative results.
The present paper belongs to this nonstandard school. (This nonstandard school is sometimes called Explicit Time school see §6 of the present paper.) One of our theses is that since there does exist a well established methodology for doing First order Nonclassical Logic (e.g. Gallin[25], Bowen[11]) by using this methodology one could develop a less ad-hoc model theory for First order Dynamic Logic (similarly to that what is happening with Propositional Dynamic Logic). I.e. one of our aims is to make First order Dynamic Logic fit better into the already existing culture of Nonclassical Logics. Some of our considerations in this line are collected in §8.

Another aim of the nonstandard school is to find decidable proof concepts for the logics under investigation. In Definition 13 of this paper a proof concept $\vdash^N$ is introduced for First order Dynamic Logic which is decidable. In Theorem 2 this proof concept $\vdash^N$ is proved to be strongly complete, i.e. for every theory Th and formula φ of First order Dynamic Logic we have $Th \models \varphi$ iff $Th \vdash^N \varphi$. The proof concept $\vdash^N$ provides also a new proof method for program verification. In §5 the new proof method $\vdash^N$ is compared with

some old ones. It is proved e.g. that $\vdash^N$ is strictly stronger than the so called Floyd-Hoare method.

Concerning the syntax of Dynamic Logic we followed the ideas presented in Pratt[41]. Hence the syntax of Dynamic Logic in the present paper is slightly different from the one in e.g. Pratt[39] but this difference does not affect the expressive power of the language. The difference is that our "action terms" or "programs" are not structured. The present paper can be translated to the original structured syntax of Dynamic Logic without changing any of the results. This translation is done in Sain[43a]. So for the present work with "regular programs" instead of "flowchart programs" see [43a].

On the connection with Henkin's higher order model theory
Higher order logic is strongly incomplete w.r.t. its standard model theory. Henkin devised a nonstandard model theory for higher order logic, see [35]. Higher order logic is complete w.r.t. Henkin's nonstandard model theory [35]. Henkin's nonstandard model theory proved to be rather useful and satisfactory e.g. in getting a deeper understanding of the nature of higher order reasoning, in applications in computer science etc. Gallin[25] adopted Henkin's nonstandard method to intensional logic. We are applying Henkin's nonstandard method to first order dynamic logic and logics of programs. (This will be especially outstanding in Definition 16 and Theorem 7.) By postulating appropriate axioms we can keep our models to be no more nonstandard than Henkin's models are. If Henkin's nonstandard approach was useful for higher order logic we do not see why it would not be useful for first order dynamic logic too.

Inclompleteness of higher order logic w.r.t. standard models is due to the fact that there is a standard model $\underset{\sim}{A}$ and a higher order formula φ such that the standard meaning of φ in $\underset{\sim}{A}$ does not depend on $\underset{\sim}{A}$ but on the model of set theory which "we are living in". Thus in different models of set theory the same φ in the same $\underset{\sim}{A}$ might have different truthvalues. This means that when the standard semantics was defined, some subtle "misinterpretation" of the language happened since a language is supposed to speak about its models (its interpretations) and not about something else. The Henkin-semantics was an attempt to repair this subtle error and in a sense it was succesful. We claim that in first order dynamic logic the situation is exactly the same, and the solution is the same too. For more in this line see Sain[43].

Historical remark

The nonstandard point of view proposed here was also adopted in §(4.3), §(4.4) of Constable[51] pages 281-282 independently of us. We discovered this fact in September 1980, actually, it seems to be the case that a problem formulated near to the end of §(4.4) of Constable[51] is solved positively in Andréka-Németi-Sain[6] and in the present paper. It appears that this Henkin type nonstandard model theory for first order Dynamic Logic was discovered independently by Constable and ourselves in 1977 and probably by many different people at different times. The basic idea of course is due to Leon Henkin. Nonstandard time models are also investigated in works of E.M.Szabo (Montréal) [42a], F.Berman e.g.[8], in Salwicki[50], Gergely-Úry[28-30], Csirmaz[16-21], Andréka-Németi[1-7], Sain[43], Bíró[10], Constable[51], Hájek[30a]. This list is far from being exhaustive.

Added in proof

Further results in the line of the present §5 are found in [37b] and [37c]. In particular, the proofs of the lattice of dynamic logics illustrated on p.109 here are in [37b],[37c].

In the following we shall recall some standard notations from textbooks on logic (mainly from [35] ,[13]). The reader is advised to skip this list and use it only when needed.

NOTATIONS

Throughout the paper:

d denotes an arbitrary similarity type of classical one-sorted models. I.e. d correlates arities (natural numbers) to function symbols and relation symbols. See Definition 1(i) in this paper.

ω denotes the set of natural numbers such that $0 \in \omega$.

$X = \{ x_w : w \in \omega \}$ denotes a set of variables.

F_d is the set of classical first order formulas of type d with variables in X. Cf. e.g. [13] p.22.

τ denotes a term of type d in the usual sense of logic, see [13]p.22 or [35]p.166 Def.10.8.(ii).

M_d denotes the class of all classical one sorted models of type d, see e.g. [13] or [35] Def.11.1., or Definition 1 and 3 here.

A classical one sorted model is denoted by an underlined capital like $\underset{\sim}{T}$ or $\underset{\sim}{D}$ and its universe is denoted by the same capital without underlining. E.g. T is the universe of $\underset{\sim}{T}$, and D is that of $\underset{\sim}{D}$.

By a "valuation of the variables" in a model $\underset{\sim}{D}$ a function $g : \omega \longrightarrow D$ is understood, see [35] p.195.

$\tau[q]_{\underset{\sim}{D}}$ denotes the value of the term τ in the model $\underset{\sim}{D}$ under the valuation q of the variables, see [13]p.27 D.13.13 or [35] Def.11.2.. If τ contains no variable then we write τ instead of $\tau[q]_{\underset{\sim}{D}}$, if $\underset{\sim}{D}$ is understood.

$\underset{\sim}{D} \models \varphi[q]$ denotes that the valuation q satisfies the formula φ in the model $\underset{\sim}{D}$.

$L_d = \langle F_d , M_d , \models \rangle$ is the classical first order language of similarity type d, see [43].

td denotes a certain many-sorted similarity type, see Definitions 3,4 here and [35].

F_{td} is the set of all classical first order many-sorted formulas of similarity type td, see [35] and Definition 5 here.

M_{td} is the class of all classical many-sorted models of similarity type td, see [35] and Definitions 3,5 here.

$L_{td} = \langle F_{td}, M_{td}, \models \rangle$ is the classical first order many-sorted language of similarity type td, see [35] and Definition 5 here.

$\mathfrak{M}$ denotes a classical many-sorted model, usually $\mathfrak{M} \in M_{td}$. The parts of $\mathfrak{M}$ are always denoted as $\mathfrak{M} = \langle \underset{\sim}{T}, \underset{\sim}{D}, I, ext \rangle$, see Definitions 3-4 here.

A_B denotes the set of all functions from A into B, i.e. $A_B = \{ f : f \text{ maps } A \text{ into } B \}$, see [35] p.7.

A function is considered to be a class of pairs.

Dom f denotes the domain of the function f, and
Rng f denotes the range of the function f, i.e.
$\text{Dom } f = \{ a : (\exists b) \langle a,b \rangle \in f \}$,
$\text{Rng } f = \{ b : (\exists a) \langle a,b \rangle \in f \}$.

A sequence s of length n is a function with
$\text{Dom } s = n \overset{df}{=} \{ 0,1,\ldots,n-1 \}$.

$\langle U_s : s \in S \rangle$ denotes the function $\{ \langle s, U_s \rangle : s \in S \}$. Moreover for an expression $Expr(x)$ and class S we define $\langle Expr(x) : x \in S \rangle$ to be the function $f : S \longrightarrow \text{Rng } f$ such that $(\forall x \in S)\ f(x) = Expr(x)$.

Natural numbers are used in the von Neumann sense i.e. $n = \{0,1,\ldots,n-1\}$ and especially

0 is the empty set.

§1. SYNTAX (of program schemes)

The followings are basically the same as the content of p.242-244 in Manna [32].

Recall d, X, F_d from the list of notations.

Now we define the set P_d of program schemes of type d.

The set Lab of "label symbols" is defined to be an arbitrary but fixed subset of the set Tm_d^0 of all constant terms of type d, i.e. d-type terms which do not contain variable symbols.

Logical symbols: $\{ \wedge, \neg, \exists, = \}$.

Other symbols: $\{ \leftarrow, \text{IF}, \text{GOTO}, \text{HALT}, (,), : \}$.

The set U_d of commands of type d is defined as:

$(i: x \leftarrow \tau) \in U_d$ if $i \in Lab$, $x \in X$, and τ is a term of type d and with all variables in X.

$(i: \text{IF } \chi \text{ GOTO } v) \in U_d$ if $i,v \in Lab$, $\chi \in F_d$ is a formula without quantifiers.

$(i: \text{HALT}) \in U_d$ if $i \in Lab$.

These are the only elements of U_d.

By a program scheme of type d we understand a finite sequence p of commands (elements of U_d) ending with a "HALT", in which no two members have the same label, and in which the only "HALT-command" is the last one. Further, if $(i: \text{IF } \chi \text{ GOTO } v)$ occurs in p then there is u such that the command $(v: u)$ occurs in p. I.e. an element p of P_d is of the form

$$p = \langle (i_0:u_0), \ldots, (i_{n-1}:u_{n-1}), (i_n:\text{HALT}) \rangle$$

where $n \in \omega$, $(i_m:u_m) \in U_d$ for $m \leq n$ etc..

Examples for program schemes can be found e.g. in the proof of Prop.6, between Def.16 and Thm.7, and in the proof of Thm.9.

The set Lab of labels was chosen the above way for technical reasons only. There are many other possible ways for handling labels. The anomalies arising from the present choice of Lab can be avoided by expanding the type d and by fixing some simple axioms, see IA^+ in Definition 14.

§2. SEMANTICS (of program schemes)

By a language with semantics we understand a triple

$L = \langle F , M , \models \rangle$ of classes such that $\models \subseteq M \times F \times Sets$

where Sets is the class of all sets.

Here F is called the syntax of L,
M is called the class of models or possible interpretations of L,
and $\models$ is called the satisfaction relation of L. Instead of $\langle a,b,c \rangle \in \models$ we write $a \models b[c]$, and we say "c satisfies b in a".
See [37], [43], and [47] p.265.

Here we try to develop a natural semantic framework for programs and statements about programs. In trying to understand the "Programming Situation", its languages, their meanings etc., the first question is how an interpretation or model of a program or program scheme $p \in P_d$ should look like. The classical approach (Manna[32], Ianov[48]) says that an interpretation or model of a program scheme is a relational structure $\underset{\sim}{D} \in M_d$ consisting of all the possible <u>data values</u>. The program p contains variables, say "x". The classical approach says that x denotes elements of D just as variables in classical first order logic do. Now we argue that x does <u>not</u> denote elements of D but rather x denotes some kind of "locations" or "addresses" which may <u>contain</u> different data values (i.e. elements of D) at different points of time. Thus there is a set I of locations, a set T of time points, and a function ext: $I \times T \longrightarrow D$ which tells for every location $s \in I$ and time point $b \in T$ what the content of location s is at time point b. Of course, this content ext(s,b) is a data value i.e. it is an element of D. Time has a structure too ("later than" etc.) and data values have structure too, thus we have structures $\underset{\sim}{T}$ and $\underset{\sim}{D}$ over the sets T and D of time points and possible data values respectively. Therefore we shall define a model or interpretation for programs $p \in P_d$ to be a four-tuple $\mathfrak{M} = \langle \underset{\sim}{T} , \underset{\sim}{D} , I , ext \rangle$ where $\underset{\sim}{T}$ and $\underset{\sim}{D}$ are the time structure and data structure resp., I is the set of locations and ext: $I \times T \longrightarrow D$ is the "content of... at time..." function (see Definition 4). We shall call the elements of I <u>intensions</u> instead of locations. The reasons for this and for the name "ext" are explained in §8. For a detailed account of the above considerations see also §7.

Of course, when specifying semantics of a programming language P_d we may have ideas about how an interpretation $\mathfrak{M}$ of P_d may look like and how it may not look. These ideas may be expressed in the form of <u>axioms</u> about $\mathfrak{M}$. E.g. we may postulate that $\underset{\sim}{T}$ of $\mathfrak{M}$ has to satisfy the Peano Axioms of arithmetic. For such axioms see Definitions 14, 15, 16, 19. These axioms are easy to express since a closer investigation of $\mathfrak{M}$ defined above reveals that it is a model of classical 3-sorted logic (the sorts being "time", "data" and "intensions"). Thus the axioms can be formed in classical 3-sorted logic (Definition 5) in a convenient manner to express all our ideas or postulates about the semantics of the programming language P_d under consideration.

Now we turn to work out these ideas in detail.

Definition 1 (one-sorted models)

(i) By a (classical or one-sorted) similarity type d we understand a pair $d = \langle H, d_1 \rangle$ such that d_1 is a function $d_1 : \Sigma \longrightarrow \omega$ for some set Σ, $H \subseteq \Sigma$ and $(\forall\, r \in \Sigma)\ d_1(r) > 0$.

The elements of Σ are called the symbols of d and the elements of H are called the operation symbols or function symbols of d.

Let $r \in \Sigma$. Then we shall write $d(r)$ instead of $d_1(r)$.

(ii) Let $d = \langle H, d_1 \rangle$ be a similarity type, let $\Sigma = \mathrm{Dom}\ d_1$ as above.

By a model of type d we understand a pair $\underset{\sim}{D} = \langle D, R \rangle$ such that R is a function with $\mathrm{Dom}\ R = \Sigma$ and $(\forall r \in \Sigma)\ R(r) \subseteq {}^{d(r)}D$ and if $r \in H$ then $R(r) : {}^{(d(r)-1)}D \longrightarrow D$.

Notation: $\langle D, R_r \rangle_{r \in \Sigma} \stackrel{\mathrm{df}}{=} \langle D, \langle R_r : r \in \Sigma \rangle \rangle \stackrel{\mathrm{df}}{=} \langle D, R \rangle$.

I.e. $\underset{\sim}{D} = \langle D, R_r \rangle_{r \in \Sigma}$ is a model of type d iff R_r is a $d(r)$-ary relation over D and if $r \in H$ then R_r is a $(d(r)-1)$-ary function, for all $r \in \Sigma$.

If $r \in H$ and $d(r) = 1$ then there is a unique $b \in D$ such that $R_r = \{\langle b \rangle\}$ and we shall identify R_r with b. If $r \in H$, $d(r) = 1$ then r is said to be a constant symbol and $R_r \in D$ is the constant element denoted by r in $\underset{\sim}{D}$.

The set D is called the universe of $\underset{\sim}{D}$.

(iii) $M_d \stackrel{\mathrm{df}}{=} \{ \underset{\sim}{D} : \underset{\sim}{D}$ is a model of type d $\}$.

End of Definition 1

Definition 2 (the similarity type t of arithmetic and its standard model $\underset{\sim}{N}$)

t denotes the similarity type of Peano's Arithmetic.

In more detail, $t = \langle \{+, \cdot, 0, 1\}, t_1 \rangle$ where $\mathrm{Dom}\ t_1 = \{\leq, +, \cdot, 0, 1\}$, $t(\leq) = 2$, $t(+) = t(\cdot) = 3$, and $t(0) = t(1) = 1$.

The standard model $\underset{\sim}{N}$ of t will be sloppily denoted as $\langle \omega, \leq, +, \cdot, 0, 1 \rangle = \underset{\sim}{N}$ instead of the more precise notation $\underset{\sim}{N} = \langle \omega, R \rangle$ where $R(\leq) = \{ \langle n, m \rangle \in {}^2\omega : n \leq m \}, \ldots,$ $R(1) = 1$.

Note that $\underset{\sim}{N} \in M_t$.

End of Definition 2

Throughout the paper t is supposed to be disjoint from any other similarity type, moreover if d is a similarity type then $\mathrm{Dom}\ d_1 \cap \mathrm{Dom}\ t_1 = 0$ is assumed throughout the paper.

Definition 3(many-sorted models, Monk[35])

Let S be a set. Then S^* denotes the set of all finite strings of elements of S, i.e. $S^* \stackrel{df}{=} \cup\{{}^nS : n\in\omega\}$.

(i) By a many-sorted similarity type m we understand a triple $m = \langle S, H, m_2\rangle$ such that m_2 is a function $m_2 : \Sigma \longrightarrow S^*$ for some set Σ, $H \subseteq \Sigma$ and $(\forall r\in\Sigma)\ m_2(r) \notin {}^0S$.
The elements of S are called the sorts of m.
If $r\in\Sigma$ then we shall write $m(r)$ instead of $m_2(r)$.

(ii) Let m be a many-sorted similarity type and let $\Sigma = \mathrm{Dom}\ m_2$ as above.
By a (many-sorted) model of type m we understand a pair $\mathfrak{M} = \langle\langle U_s : s\in S\rangle, R\rangle$ such that R is a function with $\mathrm{Dom}\ R = \Sigma$ and
if $r\in\Sigma$ and $m(r) = \langle s_1,\dots,s_n\rangle$ then $R(r) \subseteq U_{s_1}\times\dots\times U_{s_n}$
and if in addition
$r\in H$ then $R(r)$ is a function $R(r) : U_{s_1}\times\dots\times U_{s_{n-1}} \longrightarrow U_{s_n}$.
U_s is said to be the universe of sort s of $\mathfrak{M}$.

$M_m \stackrel{df}{=} \{\mathfrak{M} : \mathfrak{M}$ is a many-sorted model of type $m\}$.

End of Definition 3

For illustrations see the figures in the proofs of Theorem 9 and Proposition 6.

Definition 4(the 3-sorted similarity type td)

(i) To any one-sorted similarity type d we associate a 3-sorted similarity type td as follows:

Let $d = \langle H, d_1\rangle$ be any one-sorted similarity type.
Recall that t is a fixed similarity type introduced in Definition 2 and by our convention $\mathrm{Dom}\ d_1 \cap \mathrm{Dom}\ t_1 = 0$.

Now we define td to be $td \stackrel{df}{=} \langle S, K, td_2\rangle$ where:

a. $S \stackrel{df}{=} \{t, d, i\}$, $|S|=3$. (S is the set of sorts of td.)
Here the elements of S are used as symbols only; we could have chosen $S = \{0, 1, 2\}$ as well.

b. $K \stackrel{df}{=} \{+,\cdot, 0, 1, \mathrm{ext}\} \cup H$. (K is the set of operation symbols of td.)

c. $td_2 : (\mathrm{Dom}\ t_1 \cup \mathrm{Dom}\ d_1 \cup \{\mathrm{ext}\}) \longrightarrow S^*$ such that
$td_2(\mathrm{ext}) = \langle i,t,d\rangle$,
$td_2(r) \in {}^{n}\{t\}$ if $t(r)=n$ and
$td_2(r) \in {}^{n}\{d\}$ if $d(r)=n$.
E.g. $td_2(\leq) = \langle t,t\rangle$, $td_2(+) = \langle t,t,t\rangle,\dots, td_2(1) = \langle t\rangle$.

By these the 3-sorted similarity type td is defined.

(ii) Let $\mathfrak{M} = \langle\langle U_t, U_d, U_i\rangle, R_r\rangle_{r\in\Sigma}$ be a td-type model.
Then (1) - (3) below hold.:

(1) $\langle U_t, R_r\rangle_{r\in \mathrm{Dom}\ t_1} \in M_t$.

(2) $\langle U_d, R_r\rangle_{r\in \mathrm{Dom}\ d_1} \in M_d$.

(3) $R_{\mathrm{ext}} : U_i\times U_t \longrightarrow U_d$.

Notation:

$$\langle \langle U_t , R_r \rangle_{r \in \mathrm{Dom}\, t_1} , \langle U_d , R_r \rangle_{r \in \mathrm{Dom}\, d_1} , U_i , R_{ext} \rangle \stackrel{df}{=}$$

$$\stackrel{df}{=} \langle \langle U_t, U_d, U_i \rangle , R_r \rangle_{r \in \Sigma} = \mathfrak{M} .$$

We define

$$\underset{\sim}{T} \stackrel{df}{=} \langle U_t , R_r \rangle_{r \in \mathrm{Dom}\, t_1} , \quad T \stackrel{df}{=} U_t ,$$

$$\underset{\sim}{D} \stackrel{df}{=} \langle U_d , R_r \rangle_{r \in \mathrm{Dom}\, d_1} , \quad D \stackrel{df}{=} U_d \quad \text{and} \quad I \stackrel{df}{=} U_i .$$

Convention:

Whenever an element of M_{td} is denoted by the letter $\mathfrak{M}$ then the parts of $\mathfrak{M}$ are denoted as follows:

$$\langle \underset{\sim}{T} , \underset{\sim}{D} , I , ext \rangle \stackrel{df}{=} \langle \langle U_t^{\mathfrak{M}} , U_d^{\mathfrak{M}} , U_i^{\mathfrak{M}} \rangle , r^{\mathfrak{M}} \rangle_{r \in \Sigma} \stackrel{df}{=} \mathfrak{M} .$$

The sorts t, d, and i are called time, data and intensions respectively.

$\underset{\sim}{T}$ is said to be the time-structure of $\mathfrak{M}$.

End of Definition 4

Note that $\mathfrak{M} \in M_{td}$ iff ($\underset{\sim}{T} \in M_t$, $\underset{\sim}{D} \in M_d$, and $ext : I \times T \longrightarrow D$).

For a more detailed introduction to many-sorted languages, like $L_{td} = \langle F_{td} , M_{td} , \models \rangle$ defined below, the reader is referred e.g. to the textbook [35]. If understanding Definitions 3-6 here is hard for the reader then consulting the textbook [35] should help since L_{td} is the most usual classical many-sorted language of similarity type td .

For examples and drawings of elements of M_{td} see the proofs of Proposition 6 and Theorem 9.

Definition 5 (the first order 3-sorted language $L_{td} = \langle F_{td} , M_{td} , \models \rangle$ of type td , Monk [35])

Let $d = \langle H , d_1 \rangle$ be any one-sorted similarity type.

Recall from Definitions 2,4 that t is a fixed similarity type, and td is a 3-sorted similarity type with sorts {t,d,i} .

(i) We define the set F_{td} of first order 3-sorted formulas of type td :

Let $X \stackrel{df}{=} \{ x_w : w \in \omega \}$, $Y \stackrel{df}{=} \{ y_w : w \in \omega \}$ and $Z \stackrel{df}{=} \{ z_w : w \in \omega \}$ be three disjoint sets (and $x_w \neq x_j$ if $w \neq j \in \omega$ etc.).

We define Z , X , and Y to be the sets of variables of sorts t , d , and i respectively.

F_t^Z denotes the set of all first order formulas of type t with variables in Z ,

F_d denotes the set of all first order formulas of type d with variables in X , and

Tm_t^Z denotes the set of all first order terms of type t with variables in Z .

The set $Tm_{td,d}$ of terms of type td and of sort d is defined to be the smallest set satisfying conditions (1) - (3) below.

(1) $X \subseteq Tm_{td,d}$.

(2) $ext(y_w, \tau) \in Tm_{td,d}$ for any $\tau \in Tm_t^Z$ and $w \in \omega$.

(3) $f(\tau_1, \ldots, \tau_n) \in Tm_{td,d}$ for any $f \in H$ if $d(f) = n+1$ and $\tau_1, \ldots, \tau_n \in Tm_{td,d}$.

The set F_{td} of first order formulas of type td is defined to be the smallest set satisfying conditions (4) - (8) below.

(4) $(\tau_1 = \tau_2) \in F_{td}$ for any $\tau_1, \tau_2 \in Tm_{td,d}$.

(5) $r(\tau_1, \ldots, \tau_n) \in F_{td}$ for any $\tau_1, \ldots, \tau_n \in Tm_{td,d}$ and for any $r \notin H$ if $d(r) = n$.

(6) $(y_w=y_j) \in F_{td}$ for any $w,j \in \omega$.

(7) $F_t^Z \subseteq F_{td}$.

(8) If $\varphi, \psi \in F_{td}$ then

$\{ \neg\varphi, (\varphi \wedge \psi), (\exists z_w \varphi), (\exists x_w \varphi), (\exists y_w \varphi) : w \in \omega \} \subseteq F_{td}$.

By this the set F_{td} has been defined.

Note that $F_d \subseteq F_{td}$.

(ii) Now we define the "meanings" of elements of F_{td} .

By a valuation (of the variables) into $\mathfrak{M}$ we understand a triple $v = \langle g,k,r \rangle$ such that $g \in {}^{\omega}T$, $k \in {}^{\omega}D$ and $r \in {}^{\omega}I$.

The statement "the valuation $v = \langle g,k,r \rangle$ satisfies φ in $\mathfrak{M}$" is denoted by $\mathfrak{M} \models \varphi[v]$ or equivalently by $\mathfrak{M} \models \varphi[g,k,r]$.

The truth of $\mathfrak{M} \models \varphi[g,k,r]$ is defined the usual way (see Monk [35]) which is completely analogous with the one-sorted case. E.g.:

$\mathfrak{M} \models (y_0=y_1)[g,k,r]$ iff $r_0=r_1$, and

$\mathfrak{M} \models (x_1=\mathrm{ext}(y_2,z_0))[g,k,r]$ iff $k_1=\mathrm{ext}^{\mathfrak{M}}(r_2,g_0)$,

$\mathfrak{M} \models \varphi[g,k,r]$ iff $\underset{\sim}{T} \models \varphi[g]$ for $\varphi \in F_t^Z$,

$\mathfrak{M} \models \varphi[g,k,r]$ iff $\underset{\sim}{D} \models \varphi[k]$ for $\varphi \in F_d$ etc..

The formula $\varphi \in F_{td}$ is valid in $\mathfrak{M}$, in symbols $\mathfrak{M} \models \varphi$, iff $(\forall g \in {}^{\omega}T)(\forall k \in {}^{\omega}D)(\forall r \in {}^{\omega}I)\ \mathfrak{M} \models \varphi[g,k,r]$.

(iii) The (3-sorted) language L_{td} of type td is defined to be the triple $L_{td} = \langle F_{td}, M_{td}, \models \rangle$ where $\models$ is the satisfaction relation defined in (ii) above.

End of Definition 5

Definition 6(the similarity type d' and the standard model $\mathfrak{N}$ of type td')

Let the similarity type d' be a disjoint copy of t , i.e. let $d' = \langle \{+', \cdot', 0', 1'\}, d_1' \rangle$ where $\mathrm{Dom}\, d_1' = \{\leq', +', \cdot', 0', 1'\}$ and $d'(+')=d'(\cdot')=3$, $d'(0')=d'(1')=1$ and $d'(\leq')=2$.

In defining $\mathfrak{N} \in M_{td}$, we shall use the Convention at the end of Def.4

The standard model $\mathfrak{N}$ of type td' is defined to be

$\mathfrak{N} \overset{df}{=} \langle \underset{\sim}{N}, \underset{\sim}{N}', {}^{\omega}\omega, \mathrm{ext} \rangle$ where

$\underset{\sim}{N}$ is the standard model of type t (see Definition 2) ,

$\underset{\sim}{N}'$ is the same standard model but of type d' , and

$(\forall s \in {}^{\omega}\omega)(\forall b \in N)\ \mathrm{ext}(s,b) = s(b)$.

End of Definition 6

In this paper we shall define several sets of axioms in the language L_{td} , see Definitions 14, 15, 16 , 19 .

Each of them will be valid in the standard model $\mathfrak{N}$.

Recall the definition of a program scheme from §1.

Now we define the meanings of program schemes $p \in P_d$ in the 3-sorted models $\mathfrak{M} \in M_{td}$.

Terminology:

What we call here a d-type model $\underset{\sim}{D} \in M_d$ was called an "interpretation" $\mathfrak{I}$ in Manna[32]4-1.2. Definition 7 is a formalized version of the one given in [32]p.244-245.

Conventions 1 :

If a program scheme is denoted by p then its parts are denoted as follows:

$$p = \langle (i_0 : u_0), \ldots, (i_{n-1} : u_{n-1}), (i_n : \mathrm{HALT}) \rangle .$$

Throughout we use the definition:

$$c \overset{d}{=} \min \{ w \in \omega : (\forall v \in \omega \sim w)[x_v \text{ does not occur in } p] \} .$$

I.e. $\{x_w : w < c\}$ contains all the variables occurring in the program scheme p, and if $c > 0$ then x_{c-1} really occurs in p. We shall use x_c as the control variable of p.

Notation:

Let $\langle \underset{\sim}{T}, \underset{\sim}{D}, I, \mathrm{ext} \rangle \in M_{td}$, see the Convention at the end of Definition 4.

Let $s_0, \ldots, s_m \in I$, $\bar{s} = \langle s_0, \ldots, s_m \rangle$. Let $b \in T$.

Then we define

$$\mathrm{ext}(\bar{s}, b) \overset{d}{=} \langle \mathrm{ext}(s_0, b), \ldots, \mathrm{ext}(s_m, b) \rangle .$$

Definition 7 (traces of programs in time-models)

Let $p \in P_d$ and $\mathfrak{M} \in M_{td}$. We shall use the Convention in Definition 4 and Conventions 1.

Let $s_0, \ldots, s_c \in I$ be arbitrary intensions in $\mathfrak{M}$. Let $\bar{s} = \langle s_0, \ldots, s_{c-1} \rangle$.

The sequence $\langle s_0, \ldots, s_c \rangle$ of intensions is defined to be a trace of p in $\mathfrak{M}$ if the following (i) and (ii) are satisfied.:

(i) $\mathrm{ext}(s_c, 0) = i_0$ and $\mathrm{ext}(s_c, b) \in \{ i_m : m \leq n \}$ for every $b \in T$.

(ii) For every $b \in T$ and for every $j \leq c$ if $\mathrm{ext}(s_c, b) = i_m$ then statements (1) - (3) below hold.

(1) If $u_m = "x_w \leftarrow \tau"$ then

$$\mathrm{ext}(s_j, b+1) = \begin{cases} i_{m+1} & \text{if } j = c \\ \tau[\mathrm{ext}(\bar{s}, b)]_{\underset{\sim}{D}} & \text{if } j = w \\ \mathrm{ext}(s_j, b) & \text{otherwise} \end{cases} .$$

(2) If $u_m = "\text{IF } \chi \text{ GOTO } v"$ then

$$\mathrm{ext}(s_j, b+1) = \begin{cases} v & \text{if } j = c \text{ and } \underset{\sim}{D} \models \chi[\mathrm{ext}(\bar{s}, b)] \\ i_{m+1} & \text{if } j = c \text{ and } \underset{\sim}{D} \not\models \chi[\mathrm{ext}(\bar{s}, b)] \\ \mathrm{ext}(s_j, b) & \text{otherwise} \end{cases} .$$

(3) If $u_m = "\mathrm{HALT}"$ then

$$\mathrm{ext}(s_j, b+1) = \mathrm{ext}(s_j, b) .$$

End of Definition 7

For an illustration of Definition 7 (traces) see the figures in the proof of Proposition 6. We note that traces are interesting only in models of the axiom systems Ax, IA^+, IA^q etc. introduced in Definitions 14, 15, 16, 19 and illustrated on the figure at the end of section 5. The trace on the figures in Proposition 6 is in a model of IA^+ but not in a model of Ax.

Definition 8

Let $s = \langle s_0,\dots,s_c\rangle$ be a trace of $p \in P_d$ in $\mathfrak{M} \in M_{td}$.

(i) Let $k \in {}^{\omega}D$. The trace s is said to be of input k iff

$$(\forall j \prec c)\ k(j) = ext(s_j,0) \ .$$

I.e. s is of input k iff

$$\langle k_0,\dots,k_{c-1}\rangle = \langle ext(s_0,0),\dots,ext(s_{c-1},0)\rangle \ .$$

(ii) Recall from Conventions 1 that i_n is the last label of the program p (i.e. i_n is the label of the HALT-command). Let $b \in T$.

We say that s terminates p at time b in $\mathfrak{M}$ iff

$$ext(s_c,b) = i_n \ .$$

We shall sometimes write "s terminates at b" instead of "s terminates p at time b in $\mathfrak{M}$ " .

(iii) Let $k,q \in {}^{\omega}D$. We define q to be a possible output of p with input k in $\mathfrak{M}$ iff (a)-(d) below hold for some s .

(a) $s = \langle s_0,\dots,s_c\rangle$ is a trace of p in $\mathfrak{M}$.

(b) s is of input k .

(c) There is $b \in T$ such that s terminates p at time b and

$$\langle q_0,\dots,q_{c-1}\rangle = \langle ext(s_0,b),\dots,ext(s_{c-1},b)\rangle \ .$$

(d) $(\forall j \in \omega)\left[\, j \geq c \Longrightarrow q_j = k_j \,\right]$.

If q is a possible output of p with input k in $\mathfrak{M}$ then we shall also say that $\langle q_0,\dots,q_{c-1}\rangle$ is a possible output of p with input $\langle k_0,\dots,k_{c-1}\rangle$.

End of Definition 8

Remark:

A trace $\langle s_0,\dots,s_c\rangle$ of a program $p \in P_d$ correlates to each variable x_w $(w \leq c)$ occurring in the program p an intension or "history" s_w such that the value $ext(s_w,b)$ can be considered as the "value contained in" or "extension of" x_w at time point $b \in T$. The intension $s_w \in I$ represents a function $ext(s_w,-) : T \longrightarrow D$ from time points T to data values D . This function is the "history" of the variable x_w during an execution of the program p in the model $\mathfrak{M}$. Definition 7 ensures that the sequence $\langle ext(s_0,-),\dots,ext(s_c,-)\rangle$ of functions can be considered as a behaviour or "run" or "trace" of the program p in $\mathfrak{M}$. Here s_c is the intension of the "control variable" .

Observe that a trace is nothing but a valuation of some variables of sort i .

For a valuation s of the variables of sort i into the universe I of $\mathfrak{M}$ we define

$$\mathfrak{M} \models p[s] \quad \text{iff} \quad \langle s_0,\dots,s_c\rangle \text{ is a trace of } p \text{ in } \mathfrak{M} \ .$$

By now we have defined a semantics of program schemes, i.e. we have a language

$$\langle\, P_d \ , \ M_{td} \ , \models \,\rangle \ .$$

For any set $Th \subseteq F_{td}$ of axioms we define $Mod(Th) \subseteq M_{td}$ to be the class of all models of Th . Now for every set $Th \subseteq F_{td}$ we have a language

$$PL_{Th} = \langle P_d , Mod(Th) , \models \rangle$$

where $\mathfrak{M} \models p[s]$ is defined in Definition 7 for every $\mathfrak{M} \in Mod(Th)$. We shall call such a language a programming language with semantics. But it is not yet a language for reasoning about programs. That comes in the next §3.

To consider axiomatizable classes Mod(Th) of interpretations (instead of a single interpretation $\mathfrak{M}$ or all possible ones M_{td}) to be the semantics of P_d , was suggested in works of Burstall and Darlington [12] , Courcelle and Guessarian [15], [27] p.49 etc.. We believe that this is a very important point which should be emphasized. We believe that neither considering a single model $\mathfrak{M}$ to be the semantics nor considering M_{td} or M_d to be the semantics could give us really relevant information about semantics of programming. I.e. neither $\langle P_d , \mathfrak{M} , \models \rangle$ nor $\langle P_d , , \mathfrak{N} , \models \rangle$ nor $\langle P_d , M_{td} , \models \rangle$ nor $\langle P_d , M_d ,$ some meaning function$\rangle$ could give us really deep insights if we would choose them as the central subject of our study. Note that e.g. in Manna [32] only these two extremes were considered. On the other hand, investigating

$\langle P_d , Mod(Th) , \models \rangle$ for all possible choices of $Th \subseteq F_{td}$

can give insights, especially when Th is required to be recursively enumerable. Arguments explaining why this is true were given in [37], [15], [27] , and in other works on classes of interpretations by Courcelle, Guessarian and their collegues.

About using Th :
It might look counter-intuitive to execute programs in arbitrary elements of M_{td} . However, we can collect all our postulates about time into a set $Ax \subseteq F_{td}$ of axioms which this way would define the class $Mod(Ax) \subseteq M_{td}$ of all intended interpretations of P_d . Then we can use the language PL_{Ax} . Such a set Ax of axioms will be proposed in Definition 14. If one wants to define semantics with unusual time structure e.g. parallelism, nondeterminism, interactions etc. then one can choose an Ax different from the one proposed in this paper.

§3. STATEMENTS ABOUT PROGRAMS

We shall introduce our language DL_d for reasoning about programs or in other words the language DL_d of our first order dynamic logic.

Definition 9(the language DL_d of first order dynamic logic)
Let d be a (one-sorted) similarity type.

(i) DF_d is defined to be the smallest set satisfying conditions (1) - (3) below.

(1) $F_{td} \subseteq DF_d$.

(2) $(\forall p \in P_d)\ (\forall \psi \in DF_d)\ \Box(p, \psi) \in DF_d$.

(3) $(\forall \varphi, \psi \in DF_d)\ (\forall x \in X \cup Y \cup Z)\ \{\neg\varphi , (\varphi \wedge \psi), \exists x \varphi\} \subseteq DF_d$.

By this we have defined the set DF_d of dynamic formulas of type d .

(ii) Now we define the meanings of the dynamic formulas in the 3-sorted models $\mathfrak{M} \in M_{td}$.
Let $\mathfrak{M} = \langle \underset{\sim}{T} , \underset{\sim}{D} , I , ext \rangle \in M_{td}$.
Let v be a valuation of the variables of F_{td} into $\mathfrak{M}$, i.e. let $v = \langle g,k,r \rangle$ where $g \in {}^{\omega}T$, $k \in {}^{\omega}D$, and $r \in {}^{\omega}I$.
We shall define $\mathfrak{M} \models \varphi[v]$ for all $\varphi \in DF_d$.

(4) If $\varphi \in F_{td}$ then $\mathfrak{M} \models \varphi[v]$ is already defined in Definition 5 .

(5) Let $p \in P_d$ and $\psi \in DF_d$ be arbitrary.
Assume that $\mathfrak{M} \models \psi[v]$ has already been defined for every valuation v of the variables of F_{td} into $\mathfrak{M}$.
Let $g \in {}^{\omega}T$, $k \in {}^{\omega}D$, and $r \in {}^{\omega}I$. Then

$\mathfrak{M} \models \Box(p,\psi)[g,k,r]$ iff

($\mathfrak{M} \models \psi[g,q,r]$ for every possible output q of p with input k in $\mathfrak{M}$.)

For "possible output" see Definition 8.

(6) Let $\varphi, \psi \in DF_d$ and let $x \in X \cup Y \cup Z$.

Then $\mathfrak{M} \models (\neg\varphi)[g,k,r]$, $\mathfrak{M} \models (\varphi \wedge \psi)[g,k,r]$ and $\mathfrak{M} \models (\exists x \varphi)[g,k,r]$ are defined the usual way.

Let e.g. $w \in \omega$. Then $\mathfrak{M} \models (\exists z_w \varphi)[g,k,r]$ iff (there is $h \in {}^{\omega}T$ such that $(\forall j \in \omega)(j \neq w \Rightarrow h_j = g_j)$ and $\mathfrak{M} \models \varphi[h,k,r]$) .

(iii) The language DL_d of first order dynamic logic of type d is defined to be the triple

$$DL_d \overset{df}{=} \langle DF_d , M_{td} , \models \rangle$$

where $\models$ is defined in (ii) above.

End of Definition 9

Notation:

Let $p \in P_d$ and $\psi \in DF_d$.

Then $\Diamond(p,\psi)$ abbreviates the formula $\neg\Box(p,\neg\psi)$.

In our language DF_d we introduced the logical connectives $\neg$, $\wedge$, $=$, $\exists$, $\Box$ only. However, we shall use the following derived logical connectives $\forall$, $\rightarrow$, $\leftrightarrow$, $\vee$, TRUE, FALSE , $\Diamond$ too in the standard sense. E.g. $(\varphi \vee \psi)$ stands for the formula $\neg(\neg\varphi \wedge \neg\psi)$.

Remark:

Standard concepts of programming theory can be expressed in DL_d as follows:

1. p is partially correct w.r.t. output condition ψ	$\Box(p,\psi)$
2. p is partially correct w.r.t. input condition φ and output condition ψ (in Hoare's notation $\varphi\{p\}\psi$)	$(\varphi \rightarrow \Box(p,\psi))$
3. p is totally correct w.r.t. output condition ψ in the weaker sense	$\Diamond(p,\psi)$
4. p is totally correct in the stronger sense	$\Diamond(p,\psi) \wedge \Box(p,\psi)$
5. p terminates	$\Diamond(p, TRUE)$
6. if the input of p_2 is the output of p_1 then p_2 is totally correct w.r.t. ψ	$\Box(p_1, \Diamond(p_2,\psi))$
7. p is deterministic	$(\forall x_0 \dots \forall x_{2c-1})$ $[\Diamond(p,(\bigwedge_{j<c} x_j = x_{c+j})) \rightarrow \Box(p,(\bigwedge_{j<c} x_j = x_{c+j}))]$.
8. for any fixed input, if some output of p satisfies ψ then every output of p satisfies ψ	$\Diamond(p,\psi) \rightarrow \Box(p,\psi)$
9. p satisfies the input-output relation $\psi(x_0,\dots,x_{c-1},x_c,\dots,x_{2c-1})$ where $x_c,\dots,x_{2c-1}$ is the input and $x_0,\dots,x_{c-1}$ is the output	$(\forall x_0 \dots \forall x_{2c-1})$ $[(\bigwedge_{j<c} x_j = x_{j+c}) \rightarrow \Box(p,\psi(x_0,\dots,x_{2c-1}))]$.
10. p_1 simulates p_2	$(\forall x_0 \dots \forall x_{2c-1})$ $[\Diamond(p_2,\bigwedge_{j<c} x_j = x_{j+c}) \rightarrow \Diamond(p_1,\bigwedge_{j<c} x_j = x_{c+j})]$.

We shall use the model theoretic consequence relation in the usual sense. I.e.

Let $Th \subseteq DF_d$, $\varphi \in DF_d$ and $K \subseteq M_{td}$. Then

$\mathfrak{M} \models \varphi$ iff $(\forall g \in {}^{\omega_T})(\forall k \in {}^{\omega_D})(\forall r \in {}^{\omega_I})\ \mathfrak{M} \models \varphi[g,k,r]$,

$\mathfrak{M} \models Th$ iff $(\forall \varphi \in Th)\ \mathfrak{M} \models \varphi$,

$K \models Th$ iff $(\forall \mathfrak{M} \in K)\ \mathfrak{M} \models Th$,

$Mod(Th) \overset{df}{=} Mod_{td}(Th) \overset{df}{=} \{\mathfrak{M} \in M_{td} : \mathfrak{M} \models Th\}$, and

$Th \models \varphi$ iff $Mod(Th) \models \varphi$.

Note that $Mod(Th)$ is a sloppy abbreviation of $Mod_{td}(Th)$, we shall use it when context helps the reader to guess which similarity type h such that $Th \subseteq F_h$ is used in $Mod(Th) = Mod_h(Th)$.

§4. PROPERTIES OF THE LANGUAGE DL_d . COMPLETENESS THEOREM

Theorem 1

Let $Th \subseteq DF_d$ be recursively enumerable.

Then $\{\varphi \in DF_d : Th \models \varphi\}$ is recursively enumerable.

Proof:

This Theorem 1 is a consequence of Theorem 2 below. **See Fact 1 below and Lemmas 3,4 in the proof of Theorem 2.**

QED

Notations:

Let X be a set. Then X^* **denotes the set of all finite sequences of** elements of X, i.e. $X^* = \bigcup\{{}^mX : m \in \omega\}$.

We shall identify X^* with $\{H : H \subseteq X \text{ and } |H| < \omega\}^*$, and also with $(X^*)^*$.

We think of X^* as the set of "words over the alphabet X".

Sb X denotes the set of all subsets of X .

Definition 10 (proof concept Monk [35])

Let $L = \langle F, M, \models \rangle$ be a language.

By a proof concept on the set F we understand a relation $\vdash \subseteq Sb(F) \times F$ together with a set $Pr \subseteq F^*$ such that $(\forall Th \subseteq F)(\forall \varphi \in F)$

$[Th \vdash \varphi$ iff $\langle H, w, \varphi \rangle \in Pr$ for some finite $H \subseteq Th$ and for some $w \in F^*]$.

The proof concept $\langle \vdash, Pr \rangle$ is decidable iff the set Pr is a decidable subset of F^* in the usual sense of the theory of algorithms and recursive functions (i.e. if Pr is recursive).

Pr is called the set of proofs, and $\vdash$ is called derivability relation

End of Definition 10

Sometimes we shall sloppily write "$\vdash$ is a decidable proof concept" instead of "$\langle \vdash, Pr \rangle$ is a decidable proof concept" .

Note that the usual proof concept of classical first order logic is a decidable one in the sense of the above definition. As a contrast we note that the so called effective ω-rule is not a decidable proof concept.

Fact 1

Let $\langle \vdash, Pr \rangle$ be a decidable proof concept on the set F .
Let $Th \subseteq F$ be recursively enumerable. Then
$\{ \varphi \in F : Th \vdash \varphi \}$ is recursively enumerable too.

Proof:

Well known from the theory of algorithms.

QED

Theorem 2 (strong completeness of DL_d)

(i) There is a decidable proof concept $\vdash^N$ for the language DL_d such that for every $Th \subseteq DF_d$ and $\varphi \in DF_d$ we have

$$Th \models \varphi \quad \text{iff} \quad Th \vdash^N \varphi .$$

(ii) The language DL_d is compact.

Proof of Theorem 2 :

The idea of the proof is to reduce (or translate) the language DL_d to the complete and compact (classical 3-sorted) language $L_{td} = \langle F_{td}, M_{td}, \models \rangle$ by a total computable function $\theta : DF_d \longrightarrow F_{td}$ such that condition (*) below holds.

(*) for every $\varphi \in DF_d$ and for every $\mathfrak{M} \in M_{td}$ we have

$$\mathfrak{M} \models \varphi \quad \text{iff} \quad \mathfrak{M} \models \theta(\varphi) .$$

Suppose we have a "translation" function $\theta : DF_d \longrightarrow F_{td}$ which satisfies (*). Then the proof of Theorem 2 will be the following:

Let $Th \subseteq DF_d$, $\varphi \in DF_d$.

By an $\vdash^N$-proof of φ from Th we shall understand a sequence $\langle H, \langle \theta[H], w, \theta(\varphi) \rangle , \varphi \rangle$ such that $H \subseteq Th$, $|H| < \omega$, and $\langle \theta[H], w, \theta(\varphi) \rangle$ is a classical first order proof of $\theta(\varphi)$ from $\theta[H]$, where $\theta[H] \stackrel{df}{=} \{ \theta(\psi) : \psi \in H \}$.

Since $\theta : DF_d \longrightarrow F_{td}$ is a total computable function, our proof concept $\vdash^N$ will be decidable. Then using Gödel's completeness theorem for L_{td} and property (*) of θ we shall prove the completeness theorem for DL_d .

First we define the function $\theta : DF_d \longrightarrow F_{td}$.

Convention 2:

Instead of z_o we shall often write z .

Notation:

Let $\chi \in F_d$ be a formula without quantifiers.

Let $\bar{x} = \langle x_o,\ldots,x_e\rangle$ be a sequence of variables from X such that the variables occurring in χ are among $\{x_o,\ldots,x_e\}$.

Then $\chi[ext(\bar{y},z)]$ denotes the formula obtained from χ such that x_j is substituted in χ everywhere by $ext(y_j,z)$ for each $j \leq e$.

If τ is a term of type d and with variables in $\{x_o,\ldots,x_e\}$ then we use the notation $\tau[ext(\bar{y},z)]$ similarly.

Note that $\chi[ext(\bar{y},z)] \in F_{td}$ and $\tau[ext(\bar{y},z)] \in Tm_{td,d}$.

Definition 11 (the function $\theta : DF_d \longrightarrow F_{td}$)

The definition of θ goes in 2 steps.

(1) First we define a function $\mu : P_d \times \omega \longrightarrow F_{td}$.

Let $p = \langle (i_o:u_o),\ldots,(i_n:HALT)\rangle \in P_d$ be fixed .

Recall from Conventions 1 that c is associated to p such that the variables occurring in p are among $x_o,\ldots,x_{c-1}$, and x_c is the control variable of p .

Let $\bar{y} = \langle y_o,\ldots,y_c\rangle$.

First we define auxiliary formulas $\nu(p)$, $\nu_m(p,z,\bar{y})$ for every $m \leq n$ by statements (i) - (iv) below.:

(i) If $u_m = "x_w \leftarrow \tau"$ then

$$\nu_m(p,z,\bar{y}) \overset{df}{=} \Big[ext(y_c,z+1)=i_{m+1} \wedge ext(y_w,z+1)=\tau[ext(\bar{y},z)] \wedge \bigwedge_{\substack{j<c \\ j\neq w}} ext(y_j,z+1)=ext(y_j,z) \Big] .$$

(ii) If $u_m = "IF\ \chi\ GOTO\ v"$ then

$$\nu_m(p,z,\bar{y}) \overset{df}{=} \Big[\big(\chi[ext(\bar{y},z)] \longrightarrow ext(y_c,z+1)=v \big) \wedge \big(\neg\chi[ext(\bar{y},z)] \longrightarrow ext(y_c,z+1)=i_{m+1} \big) \wedge \bigwedge_{j<c} ext(y_j,z+1)=ext(y_j,z) \Big] .$$

(iii) If $u_m = "HALT"$ then

$$\nu_m(p,z,\bar{y}) \overset{df}{=} \bigwedge_{j\leq c} ext(y_j,z+1)=ext(y_j,z) .$$

(iv) $$\nu(p) \overset{df}{=} \nu(p,\bar{y}) \overset{df}{=} ext(y_c,0)=i_o \wedge \wedge \forall z\Big[\Big(\bigvee_{m\leq n} ext(y_c,z)=i_m\Big) \wedge \bigwedge_{m\leq n} \big(ext(y_c,z)=i_m \rightarrow \nu_m(p,z,\bar{y})\big)\Big] .$$

Now the definition of μ is the following:

Let $w \in \omega$. Then

$$\mu(p,w) \overset{df}{=} \mu(p,w)(x_o,\ldots,x_{c-1},x_w,\ldots,x_{w+c-1}) \overset{df}{=} \exists y_o\ldots\exists y_c\Big[\bigwedge_{j<c} ext(y_j,0)=x_i \wedge \nu(p,\bar{y}) \wedge \exists z(ext(y_c,z)=i_n \wedge \bigwedge_{j<c} ext(y_j,z)=x_{w+j}) \Big] .$$

Note that $\mu(p,w) \in F_{td}$ since $(\forall m \leq n)\ \nu_m(p,z,\bar{y}) \in F_{td}$.

By this we have defined the function $\mu : P_d \times \omega \longrightarrow F_{td}$

Remark: For the "meanings" of the formulas $\nu(p,\bar{y})$ and $\mu(p,w)$ see Fact 2 in Lemma 1 .

(2) Now we define $\theta : DF_d \longrightarrow F_{td}$ by recursion.
We shall use the function $\mu : P_d \times \omega \longrightarrow F_{td}$ defined in step (1).

(i) Let $\varphi \in F_{td}$. Then
$\theta(\varphi) \stackrel{df}{=} \varphi$.

(ii) Let $\varphi,\psi \in DF_d$ and suppose that $\theta(\varphi)$ and $\theta(\psi)$ are already defined. Let $p \in P_d$.

The definition of $\theta(\Box(p,\psi))$:

Let w be the smallest element of ω such that $w \geqslant c$ and x_j does not occur in ψ for every $j \geqslant w$.

Let $\bar{x} \stackrel{df}{=} \langle x_0,\ldots,x_{c-1},x_w,\ldots,x_{w+c-1}\rangle$.

Now $\theta(\Box(p,\psi)) \stackrel{df}{=} \theta(\Box(p,\psi))(x_0,\ldots,x_{c-1}) \stackrel{df}{=}$

$\stackrel{df}{=} \forall x_w \ldots \forall x_{w+c-1}(\mu(p,w)(\bar{x}) \rightarrow \exists x_0 \ldots \exists x_{c-1}(\bigwedge_{j<c} x_j = x_{w+j} \wedge \theta(\psi)))$.

$\theta(\varphi \wedge \psi) = \theta(\varphi) \wedge \theta(\psi)$.
$\theta(\neg\varphi) = \neg\theta(\varphi)$.
$\theta(\exists v \varphi) = \exists v\, \theta(\varphi)$ for every $v \in X \cup Y \cup Z$.

By these we have defined a function θ on DF_d by induction.
It is easy to check that $(\forall \varphi \in DF_d)\ \theta(\varphi) \in F_{td}$.

End of Definition 11

Now we prove that θ satisfies condition (*) .

Lemma 1

Let $\varphi \in DF_d$ and $\mathfrak{M} \in M_{td}$. Then (i) and (ii) below hold.

(i) $\mathfrak{M} \models \varphi$ iff $\mathfrak{M} \models \theta(\varphi)$.

(ii) Let $g \in {}^{\omega}T$, $k \in {}^{\omega}D$, $r \in {}^{\omega}I$. Then
$\mathfrak{M} \models \varphi[g,k,r]$ iff $\mathfrak{M} \models \theta(\varphi)[g,k,r]$.

Proof of Lemma 1:

It is enough to prove (ii) .

Let $\mathfrak{M} \in M_{td}$ and let $\langle g,k,r\rangle$ be a valuation of the variables into $\mathfrak{M}$, i.e. $g \in {}^{\omega}T$, $k \in {}^{\omega}D$, $r \in {}^{\omega}I$.

The following Fact 2 is easy to check (by inspecting Definition 11):

Fact 2

Let $p \in P_d$.

(1) $\mathfrak{M} \models \nu(p)[g,k,r]$ iff $\mathfrak{M} \models p[r]$ iff
$\langle r_0,\ldots,r_c\rangle$ is a trace of p in $\mathfrak{M}$.

(2) Let $w \geqslant c$. Then
$\mathfrak{M} \models \mu(p,w)[g,k,r]$ iff
$\langle k_w,\ldots,k_{w+c-1}\rangle$ is a possible output of p with input $\langle k_0,\ldots,k_{c-1}\rangle$.

(3) Let $\psi \in DF_d$. Then
$\mathfrak{M} \models \theta(\Box(p,\psi))[g,k,r]$ iff $\mathfrak{M} \models \Box(p,\theta(\psi))[g,k,r]$.

Now we prove (ii) by induction on $\varphi \in DF_d$.
If $\varphi \in F_{td}$ then (ii) holds by $\varphi = \theta(\varphi)$.

Let $\varphi,\psi \in DF_d$, $v \in X \cup Y \cup Z$ and assume that (ii) holds for φ and ψ .

Then (ii) holds for $\square(p,\psi)$ by Fact 2 (3) .

Then (ii) holds for $(\varphi \wedge \psi), \neg\varphi, \exists v\varphi$ by $\theta(\varphi \wedge \psi) = \theta(\varphi) \wedge \theta(\psi)$, $\theta(\neg\varphi) = \neg\theta(\varphi)$ and $\theta(\exists v\varphi) = \exists v\, \theta(\varphi)$.

Hence by the definition of DF_d (Definition 9), (ii) holds for every $\varphi \in DF_d$.

QED(Lemma 1)

Notation: Let $Th \subseteq DF_d$. Then $\theta[Th] \overset{df}{=} \{ \theta(\varphi) : \varphi \in Th \}$.

Lemma 2

The language DL_d is compact.

I.e. Let $Th \subseteq DF_d$, $\varphi \in DF_d$ be such that $Th \models \varphi$.

Then $H \models \varphi$ for some finite $H \subseteq Th$.

Proof of Lemma 2 :

Assume $Th \models \varphi$. Then $\theta[Th] \models \theta(\varphi)$ by Lemma 1 .

Since the classical 3-sorted language L_{td} is compact and since $\theta[Th] \subseteq F_{td}$, there is a finite $G \subseteq \theta[Th]$ such that $G \models \theta(\varphi)$.

Then there is a finite $H \subseteq Th$ such that $G = \theta[H]$ and therefore $\theta[H] \models \theta(\varphi)$. By Lemma 1 then $H \models \varphi$.

QED(Lemma 2)

Definition 12(classical proof concept $(\vdash, Prc)$ on F_{td} , Monk[35])

$(\vdash, Prc)$ denotes the usual proof concept of the classical 3-sorted logic $L_{td} = \langle F_{td}, M_{td}, \models \rangle$, see 10.14-10.27 of Monk[35].

I.e.

$Prc \overset{df}{=} \{ \langle H,w,\varphi \rangle : H \subseteq F_{td}, |H| < \omega, \varphi \in F_{td}$ and $(\exists m \in \omega)[w \in {}^m F_{td}$ and w is a classical first order proof of φ from H in the sense of Monk[35] 10.24$]\}$.

For any $Th \subseteq F_{td}$ and $\varphi \in F_{td}$ we define

$Th \vdash \varphi$ iff $(\exists \langle H,w,\varphi \rangle \in Prc)\ H \subseteq Th$.

Note that $(\vdash, Prc)$ is a proof concept on F_{td} in the sense of Definition 10.

End of Definition 12

Fact 3

$\vdash$ is a decidable proof concept (i.e. Prc is decidable), by 10.27 of Monk[35].

Next we define our proof concept $\vdash^N$.

Definition 13(the proof concept $(\vdash^N, Prn)$ on DF_d)

By a $\vdash^N$ -proof of $\varphi \in DF_d$ from $Th \subseteq DF_d$ we understand a sequence $\langle H, \langle \theta[H], w, \theta(\varphi) \rangle, \varphi \rangle$ such that $H \subseteq Th$ and $\langle \theta[H], w, \theta(\varphi) \rangle$ is a classical proof of $\theta(\varphi)$ from $\theta[H]$.

In more detail:

$Prn \overset{df}{=} \{ \langle H, \langle \theta[H], w, \theta(\varphi) \rangle, \varphi \rangle : H \subseteq DF_d, |H| < \omega, \varphi \in DF_d$ and $\langle \theta[H], w, \theta(\varphi) \rangle \in Prc \}$.

Let $Th \subseteq DF_d$, $\varphi \in DF_d$. Then we define

$Th \vdash^N \varphi$ iff $(\exists H \subseteq Th)(\exists w)\langle H,w,\varphi\rangle \in Prn$.

Note that $(\vdash^N, Prn)$ is a proof-concept on DF_d in the sense of Definition 10.

End of Definition 13

Lemma 3 $\vdash^N$ is a decidable proof concept.

Proof of Lemma 3 :

We have to prove that $Prn \subseteq (DF_d)^*$ is decidable.

Let $\pi \in (DF_d)^*$ be arbitrary. The algorithm of deciding whether $\pi \in Prn$ or not goes as follows:

If π is not a triple then $\pi \notin Prn$. Assume $\pi = \langle H,w,\varphi\rangle$.

By definition of $(DF_d)^*$ (and by our convention that we identify X^* with $\{H : H \in X, |H| < \omega\}^*$) we have that H is finite.

We also have that DF_d and Prc are decidable.

If $H \notin DF_d$ or $\varphi \notin DF_d$ or $w \notin Prc$ then $\pi \notin Prn$.

Assume $H \subseteq DF_d$, $\varphi \in DF_d$ and $w \in Prc$.

Since $w \in Prc$ we have that $w = \langle K,v,\psi\rangle$ for some finite $K \subseteq F_{td}$ and for some $\psi \in F_{td}$.

Compute $\theta[H]$. Since θ is computable, computing $\theta[H]$ terminates in finite time. Compute $\theta(\varphi)$ similarly.

If $\theta[H] \neq K$ or $\theta(\varphi) \neq \psi$ then $\pi \notin Prn$, else $\pi \in Prn$.

QED(Lemma 3)

Lemma 4 (completeness of the logic $\langle DL_d , (\vdash^N, Prn)\rangle$)

Let $Th \subseteq DF_d$ and $\varphi \in DF_d$ be arbitrary. Then

$Th \models \varphi$ iff $Th \vdash^N \varphi$.

Proof of Lemma 4 :

1. Assume $Th \models \varphi$.

Then $\theta[Th] \models \theta(\varphi)$ by Lemma 1. Then by Gödel's completeness theorem (see Monk[35] Thm.11.20) $\theta[Th] \vdash \theta(\varphi)$, i.e. there is a classical first order proof $\langle K,w,\theta(\varphi)\rangle \in Prc$ of $\theta(\varphi)$ from $\theta[Th]$. Then $K \subseteq \theta[Th]$ is finite, i.e. $K = \theta[H]$ for some finite $H \subseteq Th$. Then $\langle H , \langle \theta[H],w,\theta(\varphi)\rangle , \varphi\rangle \in Prn$ is an $\vdash^N$-proof of φ from Th. Thus $Th \vdash^N \varphi$.

2. Assume $Th \vdash^N \varphi$.

Then there is $\langle H , \langle \theta[H],w,\theta(\varphi)\rangle , \varphi\rangle \in Prn$ for some finite $H \subseteq Th$ such that $\langle \theta[H],w,\theta(\varphi)\rangle \in Prc$. By soundness of classical first order logic (Monk[35] Thm.11.8.) then $\theta[H] \models \theta(\varphi)$. By Lemma 1 then $H \models \varphi$, and therefore $Th \models \varphi$.

QED(Lemma 4)

Lemmas 3 and 4 prove Theorem 2. Moreover by Fact 1, Lemma 3, and Lemma 4 we proved Theorem 1, too.

QED(Theorem 2)

By a logic we understand a pair $\langle L , (\vdash,Pr)\rangle$ where $L = \langle F,M,\models\rangle$ is a language in the sense of §2 and $(\vdash,Pr)$ is a proof concept for L in the sense of Definition 10.

The logic $\langle L , (\vdash,Pr)\rangle$ is said to be complete iff $[(\vdash,Pr)$ is a decidable proof concept and for all $Th \subseteq F$ and $\varphi \in F$ we have $(Th \models \varphi$ iff $Th \vdash \varphi)]$.

We define First order Dynamic Logic of type d to be the logic $\langle DL_d , (\vdash^N, Prn)\rangle$ where the proof concept $(\vdash^N, Prn)$ was defined in Definition 13. Later we shall often say sloppily "the logic DL_d ". In these cases we shall always mean to say "the logic $\langle DL_d , (\vdash^N, Prn)\rangle$".

By Theorem 2 First order Dynamic Logic $\langle DL_d , (\vdash^N , Prn)\rangle$ is complete.

Methods of proving properties of programs:

The proof concept $\langle \vdash^N , Prn\rangle$ introduced in Definition 13 is also a new method of proving properties of programs. E.g. $\vdash^N$ can be used to prove partial correctness, total correctness, termination etc. of programs, see the example after Definition 9. The proof method $\vdash^N$ is complete by Theorem 2. In Andréka-Csirmaz-Németi-Sain[1] the proof method $\vdash^N$ was compared with the Floyd-Hoare method of proving partial correctness and it was found that $\vdash^N$ is strictly stronger i.e. there are correct programs provable by $\vdash^N$ but not provable by the Floyd-Hoare method. (See Theorem 10 here.) We shall return to this point later, in §5.

There is a second definition (, Hilbert style) of the proof concept $\vdash^N$. Then $\vdash^N$ is defined by a decidable set $Lax \subseteq DF_d$ of logical axioms and a decidable set $R \subseteq (DF_d)^* \times DF_d$ of proof rules. Both Lax and R are defined by finite schemes of formulas. Then an $\vdash^N$-proof is defined to be a finite string w of elements of DF_d such that if $w = \langle \varphi_i : i<n\rangle$ for some $n \in \omega$ then for all $i<n$ either $\varphi_i \in Lax$ or there is $\langle s, \varphi_i\rangle \in R$ such that $s \in \{\varphi_j : j<i\}^*$. This definition of $\vdash^N$ is available from the authors.

About choosing axioms to express properties of time

To execute programs in arbitrary elements of M_{td} might look counter-intuitive. However, we may replace M_{td} by $Mod(Ax)$ for a certain fixed set $Ax \subseteq F_{td}$ of axioms expressing all the intuitive requirements about time and about processes "happening in time". After having done so, there is nothing wrong with executing programs in models $\mathfrak{M} \in M_{td}$ of Ax since Ax does contain all our intuitive ideas about time, processes etc.. It is important, however, to keep Ax to be recursively enumerable.

To illustrate these here, we define a set $Ax \subseteq F_{td}$ of axioms of the above kind.

Roughly speaking, Ax will be nothing but the Peano Axioms for the sort t. However, in our present syntax F_{td} variables of sort t may occur in formulas which contain symbols of sort d and i as well. The induction axioms will be stated for these formulas "of mixed sort", too. The axiom system IA defined below originates from B. Biró.

Definition 14 (the theories PA, QA, IA, Ax_o, Ax_e, Ax): Let d be a similarity type. Then td, F_{td} and Z were defined in Definitions 4,6 in §2. Let $z \in Z$ be arbitrary. Let $\varphi \in F_{td}$.

We define the induction formula φ^+_z as follows:

$$\varphi^+_z \quad \stackrel{df}{=} \quad \big(\big[\varphi(0) \wedge \forall z(\varphi \rightarrow \varphi(z+1)) \big] \rightarrow \forall z \varphi \big) ,$$

where $\varphi(0)$ and $\varphi(z+1)$ denote the formulas obtained from φ by replacing every free occurrence of z in φ by 0 and z+1 respectively.

The induction axioms are:

$$IA \stackrel{df}{=} \{ \varphi^+_z : \varphi \in F_{td} \text{ and } z \in Z \} .$$

Clearly $IA \subseteq F_{td}$ since if $\varphi(z) \in F_{td}$ and $z \in Z$ then $\varphi(0), \varphi(z+1) \in F_{td}$ because 0 and z+1 are terms of sort t.

It is important to stress here that $\varphi(z)$ may contain other free variables of all sorts. All the free variables of $\varphi(z)$ are also free in φ^+_z except for z. They are the "parameters" of the induction φ^+_z.

The theory IA says that if a "property" $\varphi(z)$ changes during time T then it must change "some time", i.e. there is a time point $b \in T$ when $\varphi(z)$ is just changing.

We define

$$IA^+ \stackrel{df}{=} IA \cup \{(j \neq k) : j \text{ and } k \text{ are two different elements of Lab}\}.$$

Notations:

We define the abbreviations $<$ and $\prec$ as follows.:

$$z_0 < z_1 \iff [z_0 \leq z_1 \wedge z_0 \neq z_1] \quad \text{and}$$

$$z_0 \prec z_1 \iff [z_0 < z_1 \wedge \forall z_2 (\, z_0 < z_2 \rightarrow z_1 \leq z_2 \,)] \quad .$$

The finite set $OA \subseteq F_{td}$ of order axioms is defined as follows:

$$OA \stackrel{df}{=} \{ \forall z_0 (z_0 \prec z_0+1),\ \forall z_0 (0 \leq z_0 \wedge [0=z_0 \vee \exists z_1 (z_1+1=z_0)]\ ),$$
$$\forall z_0 \forall z_1 \forall z_2 (\, [z_0 \leq z_1 \vee z_1 \leq z_0] \wedge [z_0 \leq z_1 \leq z_2 \rightarrow z_0 \leq z_2] \wedge$$
$$\wedge [z_0 \leq z_1 \wedge z_1 \leq z_0) \rightarrow z_0 = z_1] \,) \,\} \quad .$$

Let PA denote the set of Peano Axioms for the sort t (see e.g. Example 1.4.11 in [13]p.42).

Now we define the theories Ax_o , Ax_e , Ax :

$$Ax_o \stackrel{df}{=} OA \cup IA^+ \quad .$$

Ax_e denotes Ax_o together with the axiom of extensionality, i.e.

$$Ax_e \stackrel{df}{=} Ax_o \cup \{ \forall y_0 \forall y_1 (\, \forall z_0 [ext(y_0,z_0)=ext(y_1,z_0)] \rightarrow y_0=y_1 \,) \} \quad .$$

$$Ax \stackrel{df}{=} Ax_o \cup PA \quad .$$

End of Definition 14

Note that Ax_o , Ax_e , $Ax \subseteq F_{td}$ and $OA \subseteq F_t$, $PA \subseteq F_t$.
Recall the similarity type d' and the standard model $\mathfrak{N} \in M_{td}$, from Definition 6. Let $d = d'$. Then $\mathfrak{N} \models Ax_e \cup PA$.

Remark:

The reason for introducing Ax_o is that all the results in this paper remain true if we replace the type t by a single binary relation symbol $\leq$, i.e. if we replace the structure $\underset{\sim}{T}$ by an ordering $\langle T, \leq \rangle$ and replace the relation $z_1 = z_0+1$ by $z_0 \prec z_1$ in all the definitions and theorems. The modified OA is then a complete axiomatization of $Th(\langle \omega, \leq \rangle)$.

Theorem 3(uniqueness of traces)

Let $p \in P_d$ and $\mathfrak{M} \in Mod(Ax_e)$. Let $k \in {}^{\omega}D$.
Then p has at most one trace of input k in $\mathfrak{M}$.

Proof of Theorem 3 :

Let $\bar{s} = \langle s_0, \ldots, s_c \rangle$ and $\bar{r} = \langle r_0, \ldots, r_c \rangle$ be two traces of p in $\mathfrak{M}$ such that $(\forall j < c) ext(s_j,0) = ext(r_j,0)$.
(I.e. $\bar{s}$ and $\bar{r}$ are of the same input.)
We define

$$\varphi(z_0) \stackrel{df}{=} (\, ext(s_0,z_0)=ext(r_0,z_0) \wedge \ldots \wedge ext(s_c,z_0)=ext(r_c,z_0) \,) \quad .$$

(Here $s_0, \ldots, s_c, r_0, \ldots, r_c$ are the parameters of the induction $\varphi^+_{z_0}$.)

$\mathfrak{M} \models \varphi(0)$ by our assumption.
$\mathfrak{M} \models \forall z_0 (\, \varphi(z_0) \rightarrow \varphi(z_0+1))$ because $\bar{s}$ and $\bar{r}$ are traces of the same program p and since $\mathfrak{M} \models (j \neq k)$ for every two distinct $i,j \in Lab$ (by $IA^+ \subseteq Ax_e$).

By $IA \subseteq Ax_e$ and $\mathfrak{M} \models Ax_e$ we have

$$\mathfrak{M} \models [\varphi(0) \wedge \forall z_0 (\, \varphi(z_0) \rightarrow \varphi(z_0+1))] \rightarrow \forall z_0 \, \varphi(z_0) \quad .$$

Therefore $\mathfrak{M} \models \forall z_0 \, \varphi(z_0)$, i.e. $(\forall j < c)(\forall b \in T) ext^{\mathfrak{M}}(s_j,b) = ext^{\mathfrak{M}}(r_j,b)$.
Then $\bar{s} = \bar{r}$ by the axiom of extensionality.

QED(Theorem 3)

The following theorem says that if a trace terminates sometime in $\mathrm{Mod}(Ax_o)$ then it cannot run again any later time. Moreover if the trace $\bar{s}$ stops sometime then there is an earliest time $m \in T$ such that $\bar{s}$ stops at time m and from that time on $\bar{s}$ remains unchanged.

Theorem 4(uniqueness of termination and output)
Let $p \in P_d$ and $\mathfrak{M} \in \mathrm{Mod}(Ax_o)$.
Let $\bar{s}$ be a trace of p in $\mathfrak{M}$ and assume that $\bar{s}$ terminates p at a time.
Then there is $m \in T$ such that for every $b \in T$ conditions (i) - (iii) below are equivalent.

(i) $b \geqslant m$.

(ii) $\bar{s}$ terminates p at time b in $\mathfrak{M}$.

(iii) $\mathrm{ext}(\bar{s},b) = \mathrm{ext}(\bar{s},m)$.

Proof of Theorem 4 :
Let $p \in P_d$, $\mathfrak{M} \in \mathrm{Mod}(Ax_o)$, and let $\bar{s} = \langle s_o,\dots,s_c \rangle$ be a trace of p in $\mathfrak{M}$.
Suppose $\bar{s}$ terminates p in $\mathfrak{M}$ at time $b_o \in T$.
Then $p = \langle (i_o:u_o),\dots,(i_n:\mathrm{HALT}) \rangle$ and $\mathrm{ext}^{\mathfrak{M}}(s_c,b_o)=i_n$.
Let $H = \{ b \in T : \mathrm{ext}^{\mathfrak{M}}(s_c,b)=i_n \}$.
We have to show that
$(\exists m \in T)(H = \{ b \in T : b \geqslant m \}$ and $(\forall b \in H)\mathrm{ext}(\bar{s},b)=\mathrm{ext}(\bar{s},m))$.

(1) Let $\varphi \overset{\mathrm{df}}{=} \varphi(z_o,y_o) \overset{\mathrm{df}}{=} \mathrm{ext}(y_o,z_o) \neq i_n$.
Then $\varphi(z_o,y_o) \in F_{td}$ since i_n is a term of sort d by definition.

Now the induction formula $\varphi^+_{z_o}$ is

$$[\varphi(0,y_o) \wedge \forall z_o(\varphi(z_o,y_o) \rightarrow \varphi(z_o+1,y_o))] \rightarrow \forall z_o \varphi(z_o,y_o) \quad .$$

By $\mathfrak{M} \models Ax_o$ and $IA \subseteq Ax_o$ we have $\mathfrak{M} \models \forall y_o(\varphi^+_{z_o})$.
$\mathfrak{M} \not\models \forall z_o \varphi(z_o,s_c)$ since $\mathrm{ext}^{\mathfrak{M}}(s_c,b_o)=i_n$.
Therefore $\mathfrak{M} \not\models [\varphi(0,s_c) \wedge \forall z_o(\varphi(z_o,s_c) \rightarrow \varphi(z_o+1,s_c))]$.
Hence either $\mathrm{ext}(s_c,0)=i_n$ or $\mathrm{ext}(s_c,b)\neq i_n$ and $\mathrm{ext}(s_c,b+1)=i_n$ for some $b \in T$.
Let $m \overset{\mathrm{df}}{=} 0$ or $m \overset{\mathrm{df}}{=} b+1$ for the above b .
Then $m \in H$ and either $m=0$ or $m=b+1$ for some $b \notin H$.
Let this m be fixed for the rest of the proof.

(2) Next we prove $(\forall b \in H)(\forall a \geqslant b)\mathrm{ext}(\bar{s},a)=\mathrm{ext}(\bar{s},b)$.
Let $\bar{y} = \langle y_o,\dots,y_c \rangle$.
Let $\psi(z_o,\bar{y}) \overset{\mathrm{df}}{=}$

$$\overset{\mathrm{df}}{=} \forall z_1 \Big[(z_1 \leq z_o \wedge \mathrm{ext}(y_c,z_1)=i_n) \rightarrow \bigwedge_{j \leq c} \mathrm{ext}(y_j,z_o)=\mathrm{ext}(y_j,z_1) \Big] \quad .$$

We shall show that $\mathfrak{M} \models \forall z_o \psi(z_o,\bar{s})$.
$\mathfrak{M} \models \psi[0,\bar{s}]$ is obvious.
Assume $\mathfrak{M} \models \psi[b,\bar{s}]$. We show $\mathfrak{M} \models \psi[b+1,\bar{s}]$.:
Case 1 $(\forall a \leq b)\mathrm{ext}(s_c,a)\neq i_n$.
Then for every $a \leq b+1$ either $a=b+1$ and then $\mathrm{ext}(\bar{s},a)=\mathrm{ext}(\bar{s},b+1)$ or $a < b+1$ and then $a \leq b$ and hence $\mathrm{ext}(s_c,a)\neq i_n$.
Thus $\mathfrak{M} \models \psi[b+1,\bar{s}]$.
Case 2 $(\exists a \leq b)\mathrm{ext}(s_c,a)=i_n$.
Then $\mathrm{ext}(s_c,b)=i_n$ by our assumption $\mathfrak{M} \models \psi[b,\bar{s}]$. Thus $\mathrm{ext}(\bar{s},b+1)=\mathrm{ext}(\bar{s},b)$ by the definition of a trace and hence $\mathfrak{M} \models \psi[b+1,\bar{s}]$.

Cases 1-2 prove that $\mathfrak{M} \models \forall z_0(\psi(z_0,\bar{s}) \to \psi(z_0+1,\bar{s}))$.

Then by $\mathfrak{M} \models \forall \bar{y}\ \psi(z_0,\bar{y})^+_{z_0}$ and by $\mathfrak{M} \models \psi[0,\bar{s}]$ we have

$\mathfrak{M} \models \forall z_0\ \psi(z_0,\bar{s})$.

I.e. we have

$\mathfrak{M} \models \forall z_0 \forall z_1 \left[(ext(s_c,z_1)=i_n \wedge z_1 \leq z_0) \to ext(\bar{s},z_0)=ext(\bar{s},z_1) \right]$.

(3) Now we prove $H = \{ b \in T : b \geq m \}$ and

$(\forall b \in H) ext(\bar{s},b)=ext(\bar{s},m)$.

By (2) we have that $(\forall a \geq m) ext(\bar{s},a)=ext(\bar{s},m)$ and therefore

$H \supseteq \{ b \in T : b \geq m \}$.

Therefore it is enough to prove $H \subseteq \{b \in T : b \geq m\}$.

Here we shall use that $\mathfrak{M} \models OA$ by $OA \subseteq Ax_0$.

If $m=0$ then $\{b \in T : b \geq m\}=T$ by $\mathfrak{M} \models OA$.

Suppose $m=b_1+1$ and $b_1 \notin H$.

Let $b \in H$ be arbitrary. Then $b \neq b_1$ by (2) and $b_1 \notin H$.

Then $b > b_1$ and therefore $b \geq b_1+1=m$ by $\mathfrak{M} \models OA$.

We have seen that there is an earliest time m when $\bar{s}$ stops and from that time on $\bar{s}$ remains unchanged.

QED(Theorem 4)

Corollary 5

Let $p \in P_d$. Then statements (i) - (iii) below hold.

(i) Let $\mathfrak{M} \models Ax_0$ and let $k \in {}^{\omega}D$.
Then there is at most one output of p with input k in $\mathfrak{M}$, i.e. p is deterministic.

(ii) $Ax_0 \models [\Diamond(p,\psi) \to \Box(p,\psi)]$ for every $\psi \in DF_d$.

(iii) $Ax_0 \models (\forall x_0 \ldots x_{2c-1}) \left[\Diamond(p, \bigwedge_{j<c} x_j=x_{c+j}) \to \Box(p, \bigwedge_{j<c} x_j=x_{c+j}) \right]$.

Proof of Corollary 5 :

(iii) is a special case of (ii) and (ii) follows from (i) which is an immediate corollary of Theorem 4 .

QED(Corollary 5)

Note that the meta-formula $\mathfrak{M} \not\models (\Diamond(p,\psi) \to \Box(p,\psi))$ means that in $\mathfrak{M}$ there is an input such that to this fixed input there are two different outputs of p such that one output satisfies ψ while the other does not. See row 8 in the table in the Remark below Definition 9.

Definition 15(the sets Pe , IA^q , and IA^f of axioms)

$Pe \overset{df}{=} \{ 0 \neq z_0+1 ,\ z_0 \neq 0 \to \exists z_1(z_1+1=z_0) ,\ z_0+1=z_1+1 \to z_0=z_1 ,$
$z_0 \neq z_0+1 ,\ z_0 \neq (z_0+1)+1 , \ldots , z_0 \neq (\ldots(z_0+1)\ldots+1) , \ldots \}$.

$IA^f \overset{df}{=} \{ \varphi \in IA^+ : \varphi$ contains no free variable of sort t or d $\}$.

$Lax \overset{df}{=} \{ j \neq k : j,k \in Lab$ and $j \neq k \}$.

$IA^q \overset{df}{=} \{ \varphi^+_z : z \in Z ,\ \varphi \in F_{td}$ and no variable of sort t is quantified in $\varphi \} \cup Lax$.

That is:

$IA^q = \{ \varphi^+_z : z \in Z ,\ \varphi \in F_{td}$ and for all $i \in \omega$ the symbol "$\exists z_i$" does not occur in $\varphi \} \cup Lax$.

Proposition 6 (Andréka-Csirmaz)

Statements (i) - (iv) below hold.

(i) $IA^+ \cup Pe \not\models \{ \Diamond(p,\psi) \to \Box(p,\psi) : p \in P_d,\ \psi \in F_d$ has one free variable $\}$.

(ii) $(IA^q \cap IA^f) \cup PA \not\models \{ \Diamond(p,\psi) \to \Box(p,\psi) : p \in P_d,\ \psi \in F_d$ has one free variable $\}$.

(iii) $IA^q \cup OA \models \{ \Diamond(p,\psi) \to \Box(p,\psi) : p \in P_d ,\ \psi \in DF_d \}$.

(iv) $IA^f \cup OA \models \{ \Diamond(p,\psi) \to \Box(p,\psi) : p \in P_d ,\ \psi \in DF_d \}$.

Proof of Proposition 6 :

To prove (iv) it is enough to observe that all the induction axioms used in the proof of Theorem 4 were ones without parameters, i.e. they were members of IA^f . (iii) was proved in [1]. (i) and (ii) can be proved from the results in sec 5 of [21] using the proof of Prop.12 in the present paper. A direct proof of (i) can be obtained by using ultraproducts. See §5 of [37/a].

Idea of proof of (i) of Proposition 6 (due to L.Csirmaz)

Z is the set of integers. $T \overset{df}{=} \omega \cup (2 \times Z)$, see the figure.

$\text{suc } n \overset{df}{=} n+1$ and $\text{suc } \langle i,z \rangle \overset{df}{=} \langle i,z+1 \rangle$ for all $i \in 2$ and $z \in Z$.

Then $\text{suc} : T \longrightarrow T$.

$\underset{\sim}{T} \overset{df}{=} \langle T, \leq, +, \cdot, 0, 1 \rangle$ where $0,1 \in T$ are the usual $0,1 \in \omega \subseteq T$,

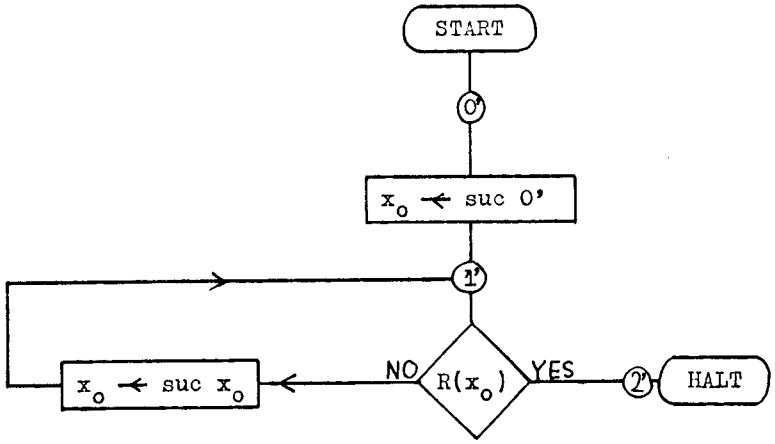
$(\forall d,b \in T)[\, d+b = \text{suc } d = d \cdot b \quad \text{and} \quad d \leq b \,]$. See the figure.

Let d contain the function symbol suc , constant symbol 0' and unary relation symbol R i.e.

$d_1 = \{ \langle \text{suc},1 \rangle , \langle 0',0 \rangle , \langle R,1 \rangle \}$ and $d_o = \{ \text{suc}, 0' \}$ and $d = \langle d_o, d_1 \rangle$.

$\underset{\sim}{D} \overset{df}{=} \langle T, \text{suc}, 0, R^D \rangle$ where $R^D \overset{df}{=} \{ \langle 0,0 \rangle, \langle 1,0 \rangle, \langle 1,1 \rangle \}$.

Let $p \in P_d$ be the program on the blockdiagram.

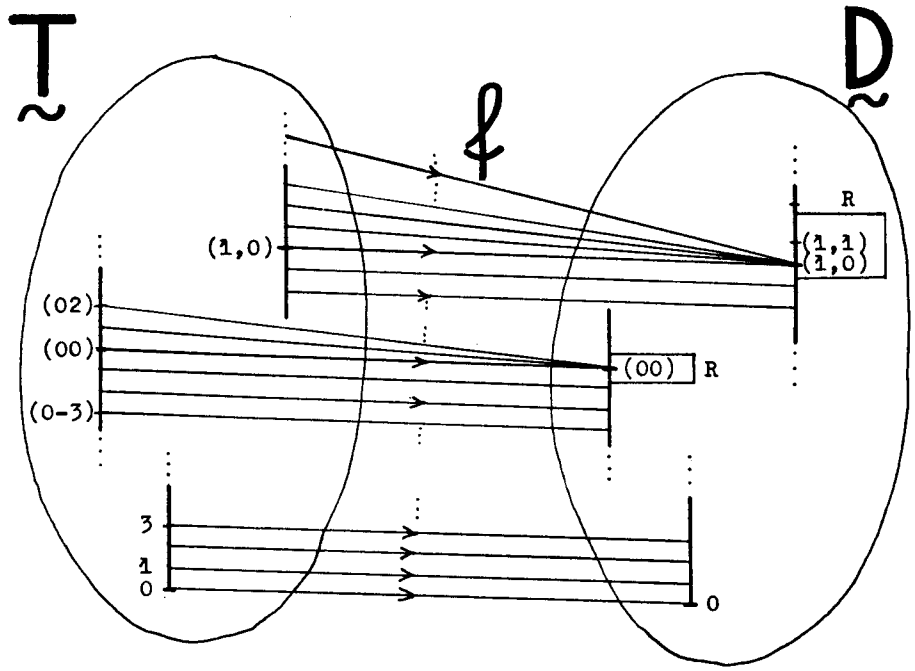


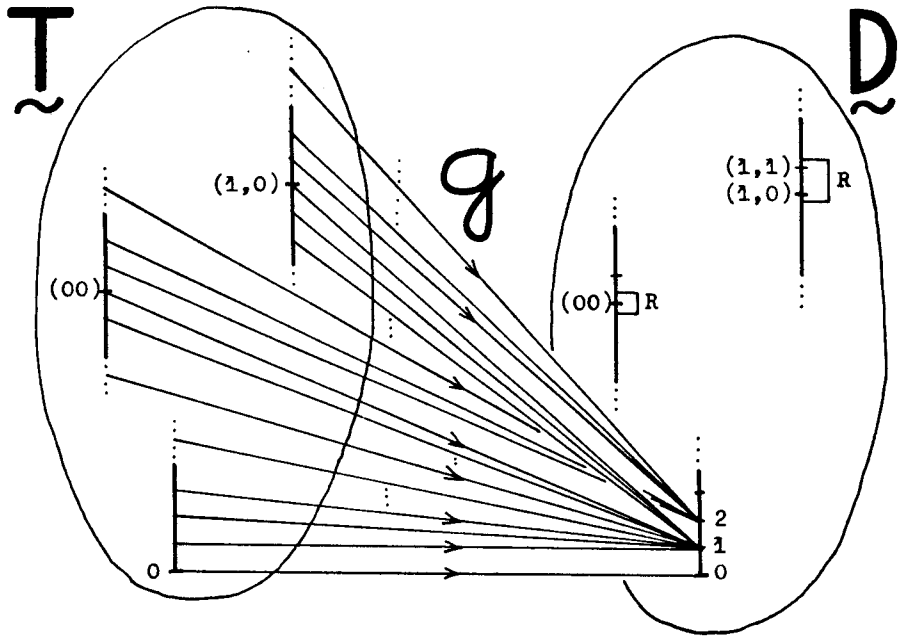
Note that to simplify matters, we omitted a label.

Let the functions f,g be as indicated on the figures below. That is:

Let $g : T \longrightarrow D$ be such that

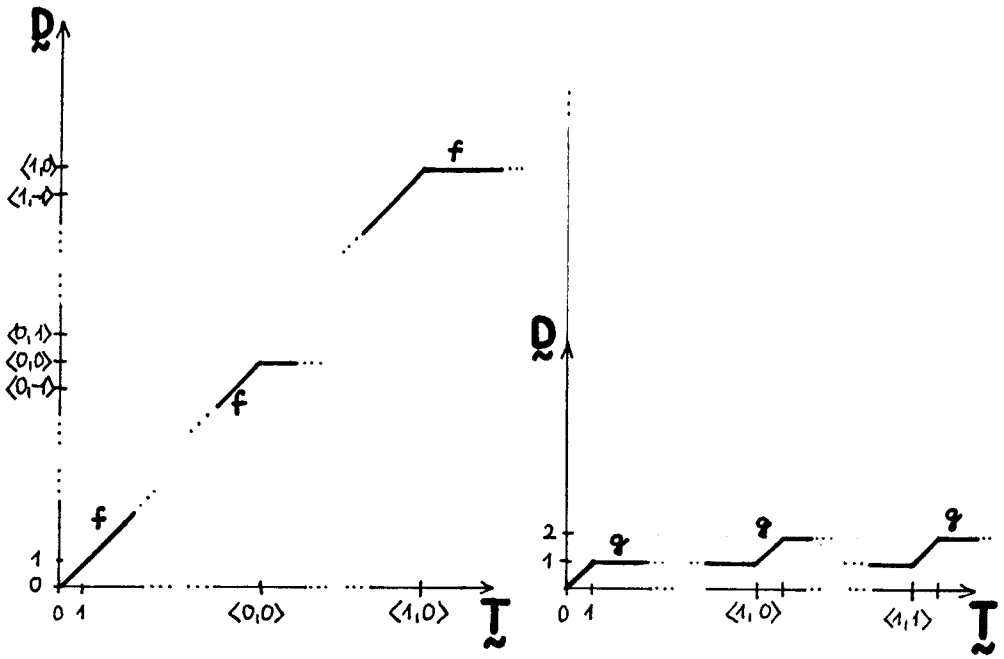
$g(0)=0$ and $(\forall n \in \omega)(\forall i \in 2)\big([n>0 \Longrightarrow (\, g(n)=1 \quad \text{and} \quad g(\langle i,n \rangle) = 2 \,)]$ and $g(\langle i,-n \rangle)=1 \,)$.





Let $f : T \longrightarrow D$ be such that

$(\forall n \in \omega)(\forall i \in 2)\big[f(n)=n$ and $f(\langle i,-n\rangle) = \langle i,-n\rangle$ and $f(\langle i,n\rangle) = \langle i,0\rangle \big]$.



Let $I \stackrel{df}{=} \{ g,f \}$ and $(\forall s \in T)\ ext(s,b) \stackrel{df}{=} s(b)$.

By using ultraproducts, one can prove

$\mathfrak{M} \stackrel{df}{=} \langle \underset{\sim}{T},\underset{\sim}{D},I,ext\rangle \models IA^+$, see Németi[37/a]. Obviously

$\mathfrak{M} \models Pe$ since $\underset{\sim}{T} \models Pe$.

Clearly $\bar{s} = \langle f,g\rangle$ is a trace of p in $\mathfrak{M}$. Then

$\mathfrak{M} \models \Diamond(p,R(\mathrm{suc}\ x_o))$ but $\mathfrak{M} \not\models \Box(p,R(\mathrm{suc}\ x_o))$ since at time $\langle 0,0\rangle$ $\bar{s}$ terminates and $\neg R(\mathrm{suc}\ s_o(\langle 0,0\rangle))$. Hence

$\mathfrak{M} \not\models \big[\Diamond(p,R(\mathrm{suc}\ x_o)) \rightarrow \Box(p,R(\mathrm{suc}\ x_o)) \big]$.

QED(Proposition 6)

L.Csirmaz proved that

$OA \cup \{ \varphi \in IA^+ : \varphi$ contains no free variable of sort t $\} \not\vdash IA$.

Thus Proposition 6(iv) is strictly stronger than $Ax_o \models \Diamond(p,\psi) \rightarrow \Box(p,\psi)$.

In many situations, the following set Ex of axioms does belong to the intuitively natural assumptions about processes happening in time.

Definition 16(the set Ex of axioms)

Notation:

"$\exists! x_o$" means that "there exists a unique x_o such that" , i.e.

$\exists! x_o \psi \iff^{df} \exists x_o(\psi \wedge \forall x_k(\exists x_o(x_k=x_o \wedge \psi) \rightarrow x_k=x_o))$, where x_k does not occur in ψ .

$$Ex \stackrel{df}{=} \{ ([\forall z_o \exists! x_o \varphi] \rightarrow \exists y_j \forall z_o \forall x_o [ext(y_j,z_o)=x_o \leftrightarrow \varphi]) : \varphi \in F_{td} \text{ and } y_j \text{ does not occur in } \varphi \} .$$

Note that φ may contain free variables and therefore the formulas in Ex written out in more detail are as follows.

Let z_o, x_o, and y_j not occur in $\bar{z}$, $\bar{x}$, and $\bar{y}$.

Let $\varphi(z_o,\bar{z},x_o,\bar{x},\bar{y})$ contain no other variables than indicated.

Then the "existence-formula" belonging to $\varphi(z_o,\bar{z},x_o,\bar{x},\bar{y})$ is

$$\forall \bar{z} \forall \bar{x} \forall \bar{y} (\forall z_o \exists! x_o\ \varphi(z_o,\bar{z},x_o,\bar{x},\bar{y}) \rightarrow \rightarrow \exists y_j \forall z_o \forall x_o (ext(y_j,z_o)=x_o \leftrightarrow \varphi(z_o,\bar{z},x_o,\bar{x},\bar{y}))) .$$

End of Definition 16

The set Ex of axioms is useful when proving formulas of kind $\Diamond(p,\psi)$. Here we illustrate this by Theorem 7 .

Let $d \overset{df}{=} \langle \{ +', -, 0', 1' \} , \{(+',3),(-,3),(0',1),(1',1)\} \rangle$.

We shall use the following abbreviations:

2' abbreviates (1'+'1') and 3' abbreviates (2'+'1') .

Let $p \in P_d$ be the following program:

$$p \overset{df}{=} \langle (0' : \text{IF } x_0=0' \text{ GOTO } 3') , (1' : x_0 \leftarrow x_0-1') , (2' : \text{IF TRUE GOTO } 0') , (3' : \text{HALT}) \rangle .$$

(Here n=3 and c=1 .)

Next we define the set DIA of induction axioms for the data.

Let $\varphi \in F_{td}$.

Then $\varphi(0')$ and $\varphi(x_0+'1')$ denote the formulas obtained from φ by replacing (every free) occurrence of x_0 in φ by 0' and $x_0+'1'$ respectively.

We define the induction formula φ^x as follows:

$$\varphi^x \overset{df}{=} ([\varphi(0') \wedge \forall x_0(\varphi \rightarrow \varphi(x_0+'1')] \rightarrow \forall x_0 \varphi) .$$

$$DIA \overset{df}{=} \{ \varphi^x : \varphi \in F_{td} \} .$$

Theorem 7

Let p and DIA be as defined above.

Let $Th \overset{df}{=} Ex \cup DIA \cup \{ \forall x_0((x_0+'1')-1'=x_0) \} \cup OA$.

Then $Th \models \Diamond(p,TRUE)$.

I.e. p terminates for every input in every model of Th .

Proof of Theorem 7 :

Recall the function θ from Definition 11.

Consider the formula $\theta(\Diamond(p,TRUE)) \in F_{td}$.

Note that the only free variable of $\theta(\Diamond(p,TRUE))$ is x_0 .

Let $\mathfrak{M} \in M_{td}$ be such that $\mathfrak{M} \models Th$.

We shall use that fact that $\mathfrak{M} \models \theta(\Diamond(p,TRUE))^x$.

First we show that $\mathfrak{M} \models \theta(\Diamond(p,TRUE))(0')$.

By Lemma 1 in the proof of Theorem 2 and by the meaning of $\Diamond(p,TRUE)$ we have to show that there is a trace $\langle s_0,s_1 \rangle$ of p in $\mathfrak{M}$ which terminates and which is of input 0' .

Let ψ_0 denote the formula $x_0=0'$.

Then $\psi_0 \in F_{td}$ and $\mathfrak{M} \models \forall z_0 \exists! x_0 \psi_0$. Then by $\mathfrak{M} \models Ex$ we have $\mathfrak{M} \models \exists y_0 \forall z_0 \forall x_0 (ext(y_0,z_0)=x_0 \leftrightarrow \psi_0)$ i.e. $\mathfrak{M} \models \exists y_0 \forall z_0 ext(y_0,z_0)=0'$.

Let $s_0 \in I$ be such that $(\forall b \in T) ext(s_0,b)=0'$.

Similarly, let ψ_1 denote the formula $(z_0=0 \rightarrow x_0=0') \wedge (z_0 \neq 0 \rightarrow x_0=3')$.

Then $\psi_1 \in F_{td}$ and $\mathfrak{M} \models \forall z_0 \exists! x_0 \psi_1$.

Hence $\mathfrak{M} \models \exists y_0 (ext(y_0,0)=0' \wedge (\forall z_0 \neq 0) ext(y_0,z_0)=3')$.

Let $s_1 \in I$ be an intension such that $ext(s_1,0)=0'$ and $ext(s_1,b)=3'$ for every $b \in T$, $b \neq 0$.

Now it is easy to check that $\langle s_0,s_1 \rangle$ is a trace of p in $\mathfrak{M}$, with input 0' and which terminates (at time 1) .

Therefore $\mathfrak{M} \models \theta(\Diamond(p,TRUE)) (0')$.

Let φ denote $\theta(\Diamond(p,TRUE))$.

Next we show that $\mathfrak{M} \models (\varphi \rightarrow \varphi(x_0+'1'))$.

Let $a \in D$ and suppose $\mathfrak{M} \models \varphi[a]$, i.e. suppose that in $\mathfrak{M}$ there is a trace $\langle s_2,s_3 \rangle$ of p which terminates and which is of input a .

We have to show that there is a trace $\langle s_4, s_5\rangle$ of p in $\mathfrak{M}$ which terminates and which is of input a+'1' .

Let ψ_4 be the formula

$(z_0 \in 1 \to x_0 = a+'1') \wedge (z_0 = 2 \to x_0 = a) \wedge \forall z_1 (z_0 = z_1 + 3 \to x_0 = ext(s_2, z_1))$.

Let ψ_5 be the formula

$(z_0 = 0 \to x_0 = 0') \wedge (z_0 = 1 \to x_0 = 1') \wedge (z_0 = 2 \to x_0 = 2') \wedge$

$\wedge \forall z_1 (z_0 = z_1 + 3 \to x_0 = ext(s_3, z_1))$.

Then by $\mathfrak{M} \models PA$ we have $\mathfrak{M} \models (\forall z_0 \exists! x_0 \psi_4 \wedge \forall z_0 \exists! x_0 \psi_5)$.

Then by $\mathfrak{M} \models Ex$ we have two intensions $s_4, s_5 \in I$ such that $\langle s_4, s_5\rangle$ is a trace of p since $\langle s_2, s_3\rangle$ is a trace of p (and by $\mathfrak{M} \models (x_0+'1')-1' = x_0$), $\langle s_4, s_5\rangle$ terminates since $\langle s_2, s_3\rangle$ terminates, and clearly $\langle s_4, s_5\rangle$ is of input a+'1' .

We have seen that $\mathfrak{M} \models \varphi(0') \wedge \forall x_0 (\varphi \to \varphi(x_0+'1'))$.
Then $\mathfrak{M} \models \forall x_0 \varphi$ by $\mathfrak{M} \models \varphi^{\times}$.
I.e. $\mathfrak{M} \models \theta(\Diamond(p, TRUE))$. Then by Lemma 1 in the proof of Theorem 2 we have $\mathfrak{M} \models \Diamond(p, TRUE)$.
QED(Theorem 7)

As a contrast we note that according to the standard semantics (see Definition 18), the set $\{ \varphi \in F_d : Th \models \varphi \}$ of axioms does not imply termination of p .

§5. COMPARING METHODS FOR PROGRAM VERIFICATION, THE STATUS OF FLOYD'S METHOD

The set HF_d of Floyd-Hoare statements of type d is an important sublanguage of DF_d .

$HF_d \overset{df}{=} \{ (\varphi \to \Box(p, \psi)) : p \in P_d \text{ and } \varphi, \psi \in F_d \}$.

Clearly $HF_d \subseteq DF_d$.

Properties of the Floyd-Hoare languages $\langle HF_d, M_{td}, \models\rangle$ and $\langle HF_d \cup F_{td}, M_{td}, \models\rangle$ were investigated in several papers, e.g. in [1], [3-7], [10], [16-21], [28-30], [43] .

In Definition 17 below we recall the Naur-Floyd-Hoare proof concept $(\vdash^F, Prf)$ for the language HF_d .
Note that $F_d \subseteq HF_d$ is practically true since φ is semantically equivalent to $(TRUE \to \Box(\langle i_0 : HALT\rangle, \varphi))$ under very mild hypotheses (namely if $i_0 \in Lab$ and $\exists y \forall z (ext(y,z) = i_0)$).

Recall the classical proof concept $(\vdash, Prc)$ from Definition 12.
We shall use Definitions 10 and 12 below.

Definition 17 (Floyd-Hoare proof concept $(\vdash^F, Prf)$)

The set Prf of all Floyd-Hoare proofs of type d is defined as follows.

$w \in Prf$ iff $w = \langle H, r, (\varphi \to \Box(p,\psi)) \rangle$ for some $(\varphi \to \Box(p,\psi)) \in HF_d$ such that conditions (i) and (ii) below hold.

(i) $H \subseteq F_d$ and $|H| < \omega$.

(ii) $r = \langle \langle \pi_0, \ldots, \pi_{n+1} \rangle, \langle \Phi_0, \ldots, \Phi_n \rangle \rangle$ such that conditions 1. - 4. below hold for every $m \leq n$.

Recall from Conventions 1 that $p = \langle (i_0:u_0), \ldots, (i_n:HALT) \rangle$.

1. $\Phi_m \in F_d$ and $\langle H, \pi_{n+1}, (\varphi \to \Phi_0) \rangle \in Prc$.

2. If $u_m =$ "$x_j \leftarrow \tau$" then $\langle H, \pi_m, (\Phi_m \to \Phi_{m+1}(x_j/\tau)) \rangle \in Prc$, where $\Phi_{m+1}(x_j/\tau)$ denotes the formula obtained from Φ_{m+1} by replacing x_j everywhere by τ .

3. If $u_m =$ "IF χ GOTO v" then $\langle H, \pi_m, (((\Phi_m \wedge \neg\chi) \to \Phi_{m+1}) \wedge ((\Phi_m \wedge \chi) \to \Phi_v)) \rangle \in Prc$.

4. If $u_m =$ "HALT" then $\langle H, \pi_m, (\Phi_m \to \psi) \rangle \in Prc$.

By these we have defined the set $Prf \subseteq (HF_d)^*$. Clearly Prf is a decidable subset of $(HF_d)^*$.

Let $Th \subseteq F_d$ and let $\varphi \in HF_d$. Then we define

$Th \vdash^F \varphi$ iff $(\exists \langle H, w, \varphi \rangle \in Prf)\ H \subseteq Th$.

By this we have defined the proof concept $(\vdash^F, Prf)$ on the language HF_d in accordance with Definition 10 .

End of Definition 17

Proposition 8

$(\vdash^F, Prf)$ is a decidable proof concept on HF_d .

Proof of Proposition 8 :

The proof is straightforward by using the fact that the set Prc of classical first order proofs is a decidable subset of $(F_d)^*$.

QED(Proposition 8)

Recall OA and IA^q from Definitions 14 and 15 respectively.

Theorem 9 (Semantic characterization of Floyd's method)

Let $Pres \subseteq PA$ be Presburger's arithmetic, i.e. $Pres$ is the theory of $\langle \omega, 0, 1, + \rangle$, and $Pres \subseteq F_t$.

Let $Th \subseteq F_d$ and $\varphi \in HF_d$ be arbitrary.

Consider statements (i) - (v) below.

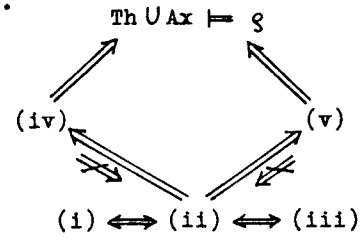
(i) $Th \vdash^F \varphi$

(ii) $Th \cup IA^q \models \varphi$

(iii) $Th \cup IA^q \cup OA \models \varphi$

(iv) $Th \cup IA^q \cup Pres \models \varphi$

(v) $Th \cup IA^+ \models \varphi$.



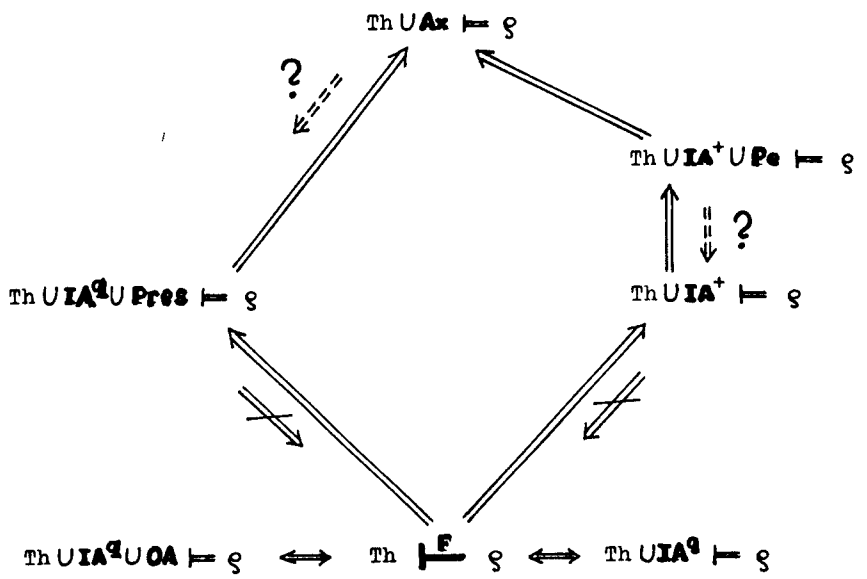
Then (i) $\Longleftrightarrow$ (ii) $\Longleftrightarrow$ (iii), (i) $\not\Longrightarrow$ (iv), (i) $\not\Longrightarrow$ (v), and (i) $\Longrightarrow$ (iv), (i) $\Longrightarrow$ (v) .

Moreover, (i) - (iii) are equivalent, but there are d , a finite $Th \subseteq F_d$ and $\varphi \in HF_d$ such that

$Th \cup IA^q \cup Pres \models \varphi$ and

$Th \cup IA \models \varphi$ but $Th \not\vdash^F \varphi$.

Illustration of Theorem 9 :



The broken arrows like ⇒ denote open problems. The arrows like $Th \cup X \models \varrho \not\Longrightarrow Th \cup Y \models \varrho$ indicate the existence of a finite d, a decidable $Th \subseteq F_d$ and $\varrho \in HF_d$ such that $Th \cup X \models \varrho$ but $Th \cup Y \not\models \varrho$.

Proof of Theorem 9 :

Let $Th \subseteq F_d$ and $\varrho \stackrel{df}{=} (\varphi \longrightarrow \Box(p,\psi)) \in HF_d$.

Proof of (i) $\Longrightarrow$ (ii) :

Assume $Th \vdash^F \varrho$.

Let $\mathfrak{M} \models Th \cup IA^q$. We have to show $\mathfrak{M} \models (\varphi \longrightarrow \Box(p,\psi))$.

Let $\langle g,k,r \rangle$ be an arbitrary evaluation of the variables.

Suppose $\mathfrak{M} \models \varphi[g,k,r]$, i.e. $\underset{\sim}{D} \models \varphi[k]$.

We have to show $\mathfrak{M} \models \Box(p,\psi)[g,k,r]$.

Let $\bar{s} = \langle s_0,\dots,s_c \rangle$ be a trace of p in $\mathfrak{M}$ with input k .

We have to show $\underset{\sim}{D} \models \psi[q]$ for every possible output q of $\bar{s}$.

$Th \vdash^F \varrho$ means that $\langle H,r,\varrho \rangle \in Prf$ for some $H \subseteq Th$ and for some r.

Let $r = \langle\langle \pi_0,\dots,\pi_{n+1} \rangle, \langle \Phi_0,\dots,\Phi_n \rangle\rangle$.

Let $\bar{y} \stackrel{df}{=} \langle y_0,\dots,y_{c-1} \rangle$ and define the formula γ as

$$\gamma \stackrel{df}{=} \gamma(z,y_0,\dots,y_c) \stackrel{df}{=} \left[\bigwedge_{m=0}^{n} (ext(y_c,z) = i_m \longrightarrow \Phi_m(ext(\bar{y},z))) \right] .$$

Clearly $\gamma \in F_{td}$ and then $\gamma^+_{z_0} \in IA^q$ since no quantifier of sort t occurs in γ .

By Definition 17 (ii)1 we have that $\langle H,r,\varrho \rangle \in Prf$ and $H \subseteq Th$ imply $Th \vdash (\varphi \longrightarrow \Phi_0)$, and then $\underset{\sim}{D} \models \Phi_0[k]$ by $\underset{\sim}{D} \models \varphi[k]$ and $\mathfrak{M} \models Th$.

Thus $\mathfrak{M} \models \gamma(0,\bar{s})$ (by $\mathfrak{M} \models \{ i_m \neq i_\ell$ for $m < \ell \leq n \}$) .

By conditions (ii)2-3 in Definition 17, by the soundness of the classical proof concept, and by the facts that $\langle H,r,\varrho \rangle \in Prf$, $\mathfrak{M} \models H$ and $\bar{s}$ is a trace of p in $\mathfrak{M}$ we obtain that $\mathfrak{M} \models \forall z_b (\gamma(z_b,\bar{s}) \rightarrow \gamma(z_{b+1},\bar{s}))$. Then by $\mathfrak{M} \models \gamma^+_{z_0}$ we conclude $\mathfrak{M} \models \forall z_0 \gamma(z_b,\bar{s})$.

Let $b \in T$ and assume that $\bar{s}$ terminates at time b with output $q \in {}^{\omega}D$. Then $\underset{\sim}{D} \models \Phi_n[q]$ since $\mathfrak{M} \models \gamma(b,\bar{s})$. By condition (ii)4 in Definition 17 we have $\langle H, \pi_n, (\Phi_n \rightarrow \psi) \rangle \in Prc$ and hence $H \models (\Phi_n \rightarrow \psi)$. Thus $\underset{\sim}{D} \models (\Phi_n \rightarrow \psi)$ and therefore $\underset{\sim}{D} \models \psi[q]$ by $\underset{\sim}{D} \models \Phi_n[q]$.

We have proved $\mathfrak{M} \models (\varphi \rightarrow \Box(p,\psi))$.

(ii) $\Longrightarrow$ (iii) is obvious.

Proof of (iii) $\Longrightarrow$ (i) :

One can prove the present implication by using [20], see Thm.4 of [19]. In doing this the methods of [16] and [21] can be useful.

We have proved (i) $\Longleftrightarrow$ (ii) $\Longleftrightarrow$ (iii) .

Proof of (v) $\not\Rightarrow$ (i) :

Let d consist of the symbols $0'$, suc, $\leq'$ with arities 0,1,2 respectively. ($0'$ and suc are function symbols and $\leq'$ is a relation symbol.) Let p_1 be the program illustrated on Figure 1 below.

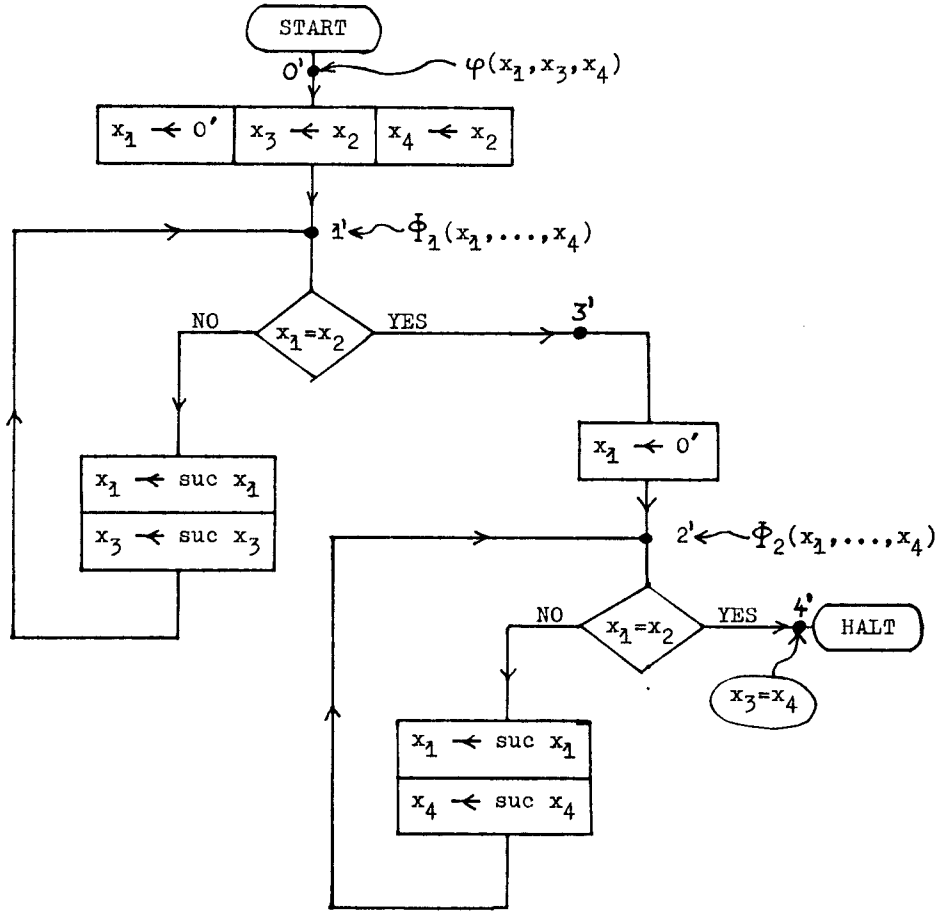


FIGURE 1

Let $\mathfrak{A} = \langle \omega, 0, \text{suc}, \leq \rangle \in M_d$ where 0 is the usual $0 \in \omega$, suc is the usual successor function on ω, $\leq$ is the natural ordering on ω. Let $\text{Thr} \stackrel{\text{df}}{=} \{ \varphi \in F_d : \mathfrak{A} \models \varphi \}$. It is known that there is a finite $\text{Th} \subseteq \text{Thr}$ such that $\text{Th} \models \text{Thr}$, see Example 3.4.4 in [13] pp.159-161. Let this Th be fixed.

In the rest of this proof we shall use

d

Th

p_1

as defined above. In the proof of Claim 9.1 we shall also use the standard model

$\mathfrak{A}$ of Th as defined above.

Claim 9.1

Let Th and p_1 be as defined above. Then

$\text{Th} \not\vdash^{F} \Box(p_1, x_3 = x_4)$.

Proof of Claim 9.1 :

Lemma 9.2

Let d , p_1 and $\text{Th} \subseteq F_d$ be as defined above. Let $\varphi \in F_d$ be any formula such that the variable x_2 does not occur in φ . Assume $\text{Th} \models \exists x_1 \exists x_3 \exists x_4 \varphi$. Then

$$\text{Th} \not\vdash^{F} (\varphi \rightarrow \Box(p_1, x_3 = x_4)) \quad .$$

Proof of Lemma 9.2 :

Let d , p_1 , $\text{Th} \subseteq F_d$ and $\varphi \in F_d$ be as in the formulation of Lemma 9.2. Let $\bar{x} = \langle x_1, x_2, x_3, x_4 \rangle$.

Suppose $\mathrm{Th} \overset{F}{\vdash} (\varphi \rightarrow \Box(p_1, x_3 = x_4))$.

Then there is a Floyd-Hoare proof $w \in \mathrm{Prf}$ of $(\varphi \rightarrow \Box(p_1, x_3 = x_4))$ such that w contains inductive assertions attached to the labels 1 and 2 of the program p_1 (see Figure 1 and Definition 17). Let $\Phi_1(\bar{x}), \Phi_2(\bar{x}) \in F_d$ be these inductive assertions attached to the labels 1 and 2 of p_1 .

We define

$$\mathrm{Fl}(\bar{x}) \overset{df}{=} \left\{ \begin{array}{l} \varphi \rightarrow \Phi_1(0', x_2, x_2, x_2) , \\ (\Phi_1(\bar{x}) \wedge x_1 \neq x_2) \rightarrow \Phi_1(\mathrm{suc}\, x_1, x_2, \mathrm{suc}\, x_3, x_4) , \\ (\Phi_1(\bar{x}) \wedge x_1 = x_2) \rightarrow \Phi_2(0', x_2, x_3, \mathrm{suc}\, x_4) , \\ (\Phi_2(\bar{x}) \wedge x_1 \neq x_2) \rightarrow \Phi_2(\mathrm{suc}\, x_1, x_2, x_3, \mathrm{suc}\, x_4) , \\ (\Phi_2(\bar{x}) \wedge x_1 = x_2) \rightarrow x_3 = x_4 \end{array} \right\} .$$

Then our hypothesis $\mathrm{Th} \overset{F}{\vdash} (\varphi \rightarrow \Box(p_1, x_3 = x_4))$ implies $\mathrm{Th} \vdash \mathrm{Fl}(\bar{x})$. Then $\mathfrak{w} \models \mathrm{Fl}(\bar{x})$.

Let $R \overset{df}{=} \{ \langle a_1, a_2, a_3, a_4 \rangle \in {}^4\omega : a_3 = 2a_2 , a_1 \leq a_2 , a_4 = a_2 + a_1 \}$.

Here $+$ is the usual addition of natural numbers and $2a_2 = a_2 + a_2$.

Claim 9.3

$(\forall \langle a_1, a_2, a_3, a_4 \rangle \in R) \; \mathfrak{w} \models \Phi_2[a_1, a_2, a_3, a_4]$.

Proof of Claim 9.3 :

Let $\langle a_1, a_2, a_3, a_4 \rangle \in R$ be arbitrary.

Since $\mathrm{Th} \models \exists x_1 \exists x_3 \exists x_4 \varphi$, there are $e_1, e_3, e_4 \in \omega$ such that $\mathfrak{w} \models \varphi[e_1, e_3, e_4]$.

Let

$\mathfrak{M} \overset{df}{=} \langle \mathfrak{w}, \mathfrak{w}, {}^{\omega}\omega, \mathrm{ext} \rangle \in M_{td}$ where $(\forall s \in {}^{\omega}\omega)(\forall b \in \omega)\; \mathrm{ext}(s,b) = s(b)$.

Consider the execution of the program p_1 in $\mathfrak{M}$ with input $\langle e_1, a_2, e_3, e_4 \rangle$. We denote the trace of p_1 of input $\langle e_1, a_2, e_3, e_4 \rangle$ by s . By the definition of a trace (Definition 7) it is easy to see that there is a time point $b \in \omega$ such that the value of the control variable x_5 at time point b is the label of Φ_2 that is

$\mathrm{ext}(s_5, b) = s_5(b) = 2$ (see Figure 1), and

the values of the program variables at time point b are:

$\langle s_1(b), s_2(b), s_3(b), s_4(b) \rangle = \langle a_1, a_2, a_3, a_4 \rangle$.

Then, by $\mathfrak{w} \models \mathrm{Fl}(\bar{x})$ and $\mathfrak{w} \models \varphi[s_1(0), s_3(0), s_4(0)]$ and by the definition of a trace we have

$$\mathfrak{w} \models \Phi_2[s_1(b), s_2(b), s_3(b), s_4(b)] .$$

So far we have seen that for every $\langle a_1, a_2, a_3, a_4 \rangle \in R$ we have

$$\mathfrak{w} \models \Phi_2[a_1, a_2, a_3, a_4] .$$

QED(Claim 9.3)

By the definition of R we have

$$\langle j, 2j, 4j, 3j\rangle \in R \quad \text{for every } j \in \omega .$$

By Claim 9.3 above we have

$$\underset{\sim}{\omega} \models \Phi_2[j, 2j, 4j, 3j] \quad \text{for every } j \in \omega .$$

Let U be a nonprincipal ultrafilter over ω .

Let $k \overset{df}{=} \langle j : j \in \omega \rangle / U$.

For every $n \in \omega$ let $nk \overset{df}{=} \langle nj : j \in \omega \rangle / U$.

Let $\underset{\sim}{D}$ be the ultrapower $\underset{\sim}{D} = {}^{\omega}\underset{\sim}{\omega}/U$ of $\underset{\sim}{\omega}$.

Then, by Łoś lemma we have

$$\underset{\sim}{D} \models \Phi_2[k, 2k, 4k, 3k] .$$

See Figure 2 !

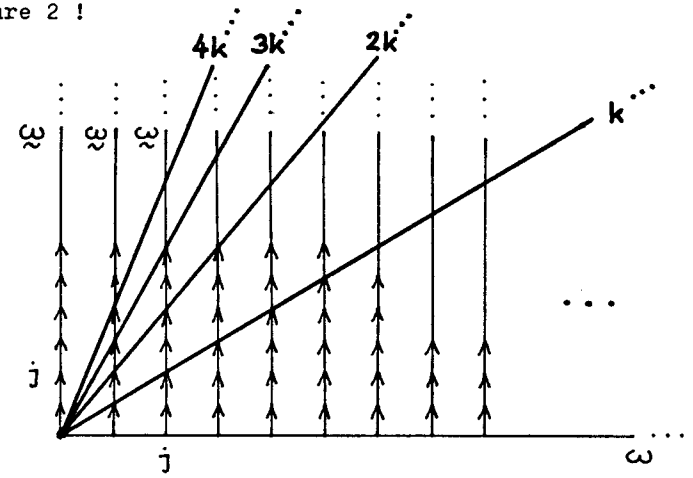


FIGURE 2

We define $\mathrm{suc}^n : D \longrightarrow D$ for every $n \in \omega$ as follows:

$$(\forall e \in D)\left[\, \mathrm{suc}^0(e) \overset{df}{=} e \quad \text{and} \quad \mathrm{suc}^{n+1}(e) \overset{df}{=} \mathrm{suc}(\mathrm{suc}\ e) \,\right] .$$

It is known that $\underset{\sim}{D} = {}^{\omega}\underset{\sim}{\omega}/U$ looks like as it is illustrated on Figure 3 and it is easy to see that the "distance" between any two of k, $2k$, $3k$, $4k$ is <u>infinite</u> i.e.

$$(\forall n \in \omega)\left[\, \mathrm{suc}^n(k) \neq 2k \quad \text{and} \quad \mathrm{suc}^n(2k) \neq 3k \quad \text{etc.} \,\right] .$$

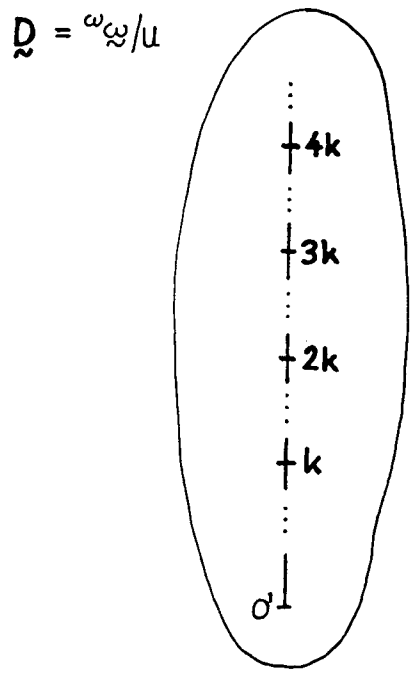


FIGURE 3

Now we define $f : D \longrightarrow D$ as follows:

Let $e \in D$ be arbitrary.

If there is an $n \in \omega$ such that $\left[\mathrm{suc}^n(e) = 4k \quad \text{or} \quad \mathrm{suc}^n(4k) = e\right]$

then $f(e) \overset{df}{=} \mathrm{suc}\ e$.

Otherwise $f(e) \overset{df}{=} e$.

Clearly f is an automorphism of $\underset{\sim}{D}$. Therefore

$$\underset{\sim}{D} \models \Phi_2[k, 2k, \mathrm{suc}(4k), 3k] .$$

By Łoś lemma we have:

$$H \overset{df}{=} \{\, j \in \omega : \underset{\sim}{\omega} \models \Phi_2[j, 2j, \mathrm{suc}(4j), 3j] \,\} \in U .$$

Since U is an ultrafilter, there is a $j \in H$. Let this j be fixed.

Then

$$\underset{\sim}{\omega} \models \Phi_2[j, 2j, \mathrm{suc}(4j), 3j] .$$

Since by our hypothesis $Th \vdash Fl(\bar{x})$ and $\mathfrak{Q} \models Th$ we have

$$\mathfrak{Q} \models \left((\Phi_2(\bar{x}) \wedge x_1 \neq x_2) \rightarrow \Phi_2(\text{suc } x_1, x_2, x_3, \text{suc } x_4) \right) .$$

Thus, from $\mathfrak{Q} \models \Phi_2[j, 2j, \text{suc}(4j), 3j]$ we can derive

$$\mathfrak{Q} \models \Phi_2[\text{suc } j, 2j, \text{suc}(4j), \text{suc}(3j)] \quad \text{if } j < 2j .$$

If $\text{suc}(\text{suc } j) < 2j$ then we can repeat this step. Actually we can repeat this step exactly j times. By induction we can conclude

$$\mathfrak{Q} \models \Phi_2[2j, 2j, \text{suc}(4j), 4j] .$$

Then, using

$$\mathfrak{Q} \models \left((\Phi_2(\bar{x}) \wedge x_1 = x_2) \rightarrow x_3 = x_4 \right) \quad \text{and} \quad \mathfrak{Q} \models Th ,$$

we get from $\mathfrak{Q} \models \Phi_2[2j, 2j, \text{suc}(4j), 4j]$ that

$$\mathfrak{Q} \models (\text{suc}(4j) = 4j)$$

which is a contradiction. Therefore our hypothesis

$Th \vdash^{F} (\varphi \rightarrow \Box(p_1, x_3 = x_4))$ is false.

QED(Lemma 9.2)

Lemma 9.2 implies $Th \not\vdash^{F} \Box(p_1, x_3 = x_4)$.

QED(Claim 9.1)

Claim 9.4

Let d, $Th \subseteq F_d$ and $p_1 \in P_d$ be as defined above the formation (at) of Claim 9.1. Then

$$Th \cup IA \models \Box(p_1, x_3 = x_4) .$$

Proof of Claim 9.4 :

CONVENTIONS

1. Throughout this proof let

   d ,

   $p_1 \in P_d$ and

   $Th \subseteq F_d$ be the ones defined above the formulation of Claim 9.1.

2. Throughout this proof let

   $\mathfrak{M} = \langle \underset{\sim}{T}, \underset{\sim}{D}, I, ext \rangle \in TM_d$ be such that $\mathfrak{M} \models Th \cup IA$ and let

   $\bar{s} = \langle s_1, s_2, s_3, s_4, s_5 \rangle$ be an arbitrary trace of p_1 in $\mathfrak{M}$.

3. For the sake of brevity, throughout this proof we shall write

   $s(b)$ instead of $ext(s,b)$. That is, for any $s \in I$ and $b \in T$ we define

   $s(b) \overset{df}{=} ext(s,b)$.

END of CONVENTIONS.

Lemma 9.4.1

$IA \models \forall z_0 \exists z_1 \; (z_0 = 0 \vee z_0 = z_1 + 1)$.

Proof of Lemma 9.4.1 :

Let $\psi(z_0)$ be the formula

$\exists z_1 \; (z_0 = 0 \vee z_0 = z_1 + 1)$.

Let $\mathfrak{M}' = \langle \underset{\sim}{T}', \underset{\sim}{D}', I', ext' \rangle$ be such that $\mathfrak{M}' \models IA$.

We shall prove $\mathfrak{M}' \models \forall z_0 \psi(z_0)$ by induction on z_0 .

Step 1 : $\mathfrak{M}' \models \psi(0)$ is obvious.

Step 2 : We prove $\mathfrak{M}' \models \forall z_0[\psi(z_0) \to \psi(z_0+1)]$. :

Let $b \in T'$.

Then $\mathfrak{M}' \models \psi(b+1)$ since

$\mathfrak{M}' \models \exists z_1 \; (\; b+1=0 \vee b+1=z_1+1 \;)$ is obvious.

End of Step 2.

By Steps 1,2 we have

$$\mathfrak{M}' \models (\; \psi(0) \wedge \forall z_0[\psi(z_0) \to \psi(z_0+1)] \;) \; . \qquad (*)$$

By $\mathfrak{M}' \models$ IA and by $\psi(z_0)^+_{z_0} \in$ IA we have

$\mathfrak{M}' \models \psi(z_0)^+_{z_0}$ that is

$$\mathfrak{M}' \models (\; \psi(0) \wedge \forall z_0[\psi(z_0) \to \psi(z_0+1)] \;) \to \forall z_0 \; \psi(z_0) \; . \qquad (**)$$

By $(*)$ and by $(**)$ we have

$\mathfrak{M}' \models \forall z_0 \; \psi(z_0)$.

QED (Lemma 9.4.1)

Lemma 9.4.2

$\mathfrak{M} \models \forall z_0 [\; s_5(z_0)=0' \leftrightarrow z_0=0 \;]$.

Proof of Lemma 9.4.2 :

1. $\mathfrak{M} \models \forall z_0 [\; s_5(z_0)=0' \leftarrow z_0=0 \;]$ by the definition of a trace (see Definition 7(i)).
2. Let $b \in T$, $b \neq 0$. Then, by Lemma 9.4.1, $b=b'+1$ for some $b' \in T$.
   By the definition of a trace and that of p_1 ,
   $s_5(b') \in \{0',1',2',3',4'\}$. By applying the same definitions again,
   $s_5(b) = s_5(b'+1) \in \{1',2',3',4'\}$.
   Then, by $\mathfrak{M} \models$ Th and by Th $\models (\; 0' \notin \{1',2',3',4'\})$ we have
   $s_5(b) \neq 0'$.
   We have proved $\mathfrak{M} \models \forall z_0 \; [\; s_5(z_0) = 0' \to z_0 = 0 \;]$.

1. and 2. together complete the proof.

QED (Lemma 9.4.2)

Lemma 9.4.3

$\mathfrak{M} \models \forall z_0 [\; s_5(z_0) \neq 0' \to \exists z_1 \; (z_0=z_1+1) \;]$.

Proof of Lemma 9.4.3 :

Let $b \in T$. Assume $s_5(b) \neq 0'$. Then, by Lemma 9.4.2, $b \neq 0$.

Then, by Lemma 9.4.1, there is a $b' \in T$ such that $b = b'+1$.

QED (Lemma 9.4.3)

Claim 9.4.4

Let $i \in \{0',1',2',3'\}$. Then

$\mathfrak{M} \models \forall z_0 \forall z_1 \; (\; [s_5(z_0) = s_5(z_1) = i \wedge s_1(z_0) = s_1(z_1)] \to z_0 = z_1 \;)$.

Proof of Claim 9.4.4 :

Let $\varphi(\bar{s},i)$ be the formula

$\forall z_0 \forall z_1 \; (\; [s_5(z_0) = s_5(z_1) = i \wedge s_1(z_0) = s_1(z_1)] \to z_0 = z_1 \;)$.

We shall prove

$\mathfrak{M} \models \varphi(\bar{s},i)$ for every $i \in \{0',1',2',3'\}$.

Claim 9.4.4.1

$\mathfrak{M} \models \varphi(\bar{s},0')$.

Proof of Claim 9.4.4.1 :

Let b_0 , $b_1 \in T$. Assume $s_5(b_0) = s_5(b_1) = 0'$. Then, by Lemma 9.4.2, $b_0 = 0$ and $b_1 = 0$. Thus $b_0 = b_1$.

QED (Claim 9.4.4.1)

Claim 9.4.4.2

$\mathfrak{M} \models \varphi(\bar{s},1')$. That is

$\mathfrak{M} \models \forall z_0 \forall z_1 \, ([s_5(z_0) = s_5(z_1) = 1' \wedge s_1(z_0) = s_1(z_1)] \rightarrow z_0 = z_1)$.

Proof of Claim 9.4.4.2 :

Let $\chi(z_0,z_1,\bar{s})$ be the formula

$[s_5(z_0) = s_5(z_1) = 1' \wedge s_1(z_0) = s_1(z_1)]$.

Let $\psi(z_0,\bar{s})$ be the formula

$\forall z_1 \, (\chi(z_0,z_1,\bar{s}) \rightarrow z_0 = z_1)$.

We shall prove $\mathfrak{M} \models \forall z_0 \, \psi(z_0,\bar{s})$ by induction on z_0 .

This induction will be possible since $\mathfrak{M} \models$ IA and $\psi(z_0,\bar{s})^+_{z_0} \in$ IA .

Step 1: $\mathfrak{M} \models \psi(0,\bar{s})$ by Lemma 9.4.2 and by Th $\models (0' \neq 1')$.

Step 2: We shall prove $\mathfrak{M} \models \forall z_0 [\psi(z_0,\bar{s}) \rightarrow \psi(z_0+1,\bar{s})]$. :

Let $b_0 \in T$ be arbitrary. Assume $\psi(b_0,\bar{s})$.

We want to prove $\psi(b_0+1,\bar{s})$.

Let $b_1 \in T$ be arbitrary. Assume $\mathfrak{M} \models \chi(b_0+1,b_1,\bar{s})$

that is assume

$$\mathfrak{M} \models [s_5(b_0+1) = s_5(b_1) = 1' \wedge s_1(b_0+1) = s_1(b_1)] . \quad (4.2.1)$$

It is enough to prove $\mathfrak{M} \models (b_0+1 = b_1)$. (4.2.2)

By our assumption (4.2.1), $s_5(b_1) = 1'$. Thus, by Lemma 9.4.3 and by Th $\models (0' \neq 1')$, there is a $b_2 \in T$ such that

$$b_1 = b_2+1 . \quad (4.2.4)$$

We have

$$s_5(b_2) \in \{0',1'\} \quad (4.2.5)$$

since otherwise we would have $s_5(b_2) \in \{2',3',4'\}$ which would imply $s_5(b_1) \underset{(4.2.4)}{=} s_5(b_2+1) \in \{2',4'\}$ by the definition of a trace. But this is impossible by $s_5(b_1) = 1'$ (see (4.2.1)) and by Th $\models (1' \notin \{2',4'\})$.

Case 1 : $b_0 = 0$.

By our hypothesis (4.2.1) we have $\mathfrak{M} \models \chi(0+1,b_1,\bar{s})$ that is

$\mathfrak{M} \models [s_5(0+1) = s_5(b_1) = 1' \wedge s_1(0+1) = s_1(b_1)]$.

By (4.2.4), $b_1 = b_2+1$, and by (4.2.5), $s_5(b_2) \in \{0',1'\}$.

Assume $s_5(b_2) = 1'$.

Then $s_5(b_2+1) = s_5(b_1) = 1'$ by (4.2.1).

By $s_5(b_2) = 1' = s_5(b_2+1)$ and by the definition of a trace, we have

$s_1(b_2+1) = \text{suc } s_1(b_2)$.

Then $\text{suc } s_1(b_2) = s_1(b_2+1) = s_1(b_1) = s_1(b_0+1) = s_1(0+1) = 0'$.

(4.2.4) (4.2.1) $b_0 = 0$ definition of a trace

By $\mathfrak{M} \models$ Th , this is a contradiction ! Thus $s_5(b_2) \neq 1'$.

Thus we have proved $s_5(b_2) = 0'$ by (4.2.5). (we are in Case 1)

Then, by Lemma 9.4.2, $b_2 = 0$. Thus $b_1 = b_2+1 = 0+1 = b_0+1$.

We have proved $\mathfrak{M} \models (b_0+1 = b_1)$ in Case 1.

Case 2 : $b_0 \neq 0$.

First we shall prove $\mathfrak{M} \models \chi(b_0,b_2,\bar{s})$.

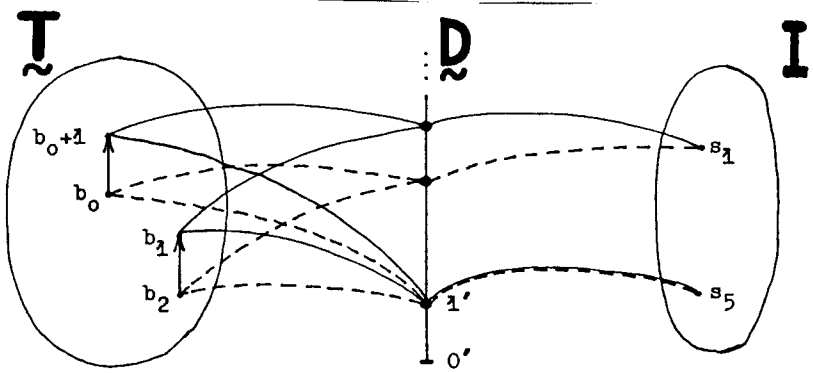


FIGURE 4

The assumption $b_0 \neq 0$ and Lemma 9.4.2 imply $s_5(b_0) \neq 0'$. Then

$$s_5(b_0) = 1' \qquad (4.2.6)$$

since otherwise we would have $s_5(b_0) \in \{2',3',4'\}$ which would imply $s_5(b_0+1) \in \{2',4'\}$ by the definition of a trace. But this is impossible by $s_5(b_0+1) = 1'$ (see (4.2.1)) and by $\mathrm{Th} \models (1' \notin \{2',4'\})$.

Now we have $s_5(b_0) = 1' = s_5(b_0+1)$. This and the definition

(4.2.6) (4.2.1)

of a trace imply

$$s_1(b_0+1) = \mathrm{suc}\, s_1(b_0) \ . \qquad (4.2.7)$$

By (4.2.5) we have $s_5(b_2) \in \{0',1'\}$.

Assume $s_5(b_2) = 0'$.

Then $s_1(b_2+1) = 0'$ by the definition of a trace. Then

$$0' = s_1(b_2+1) = s_1(b_1) = s_1(b_0+1) = \mathrm{suc}\, s_1(b_0) \ .$$

(4.2.4) (4.2.1) (4.2.7)

By $\mathfrak{M} \models \mathrm{Th}$, this is a contradiction ! Thus $s_5(b_2) \neq 0'$. Thus

$$s_5(b_2) = 1' \qquad (4.2.8)$$

by (4.2.5).

By (4.2.6) and by (4.2.8) we have seen

$$\mathfrak{M} \models (s_5(b_0) = s_5(b_2) = 1') \qquad (4.2.9)$$

By $s_5(b_2) = 1' = s_5(b_1) = s_5(b_2+1)$ and by the definition of a

(4.2.8)(4.2.1)(4.2.4)

trace we have

$$s_1(b_1) = s_1(b_2+1) = \mathrm{suc}\, s_1(b_2) \qquad (4.2.10)$$

Thus $\mathrm{suc}\, s_1(b_2) = s_1(b_1) = s_1(b_0+1) = \mathrm{suc}\, s_1(b_0)$.

(4.2.10) (4.2.1) (4.2.7)

Thus we have $\mathfrak{M} \models (\mathrm{suc}\, s_1(b_2) = \mathrm{suc}\, s_1(b_0))$.

Now, by $\mathfrak{M} \models \mathrm{Th}$ and by $\mathrm{Th} \models \forall x_0 \forall x_1 (\mathrm{suc}\, x_0 = \mathrm{suc}\, x_1 \rightarrow x_0 = x_1)$ we have

$$\mathfrak{M} \models (s_1(b_2) = s_1(b_0)) \ . \qquad (4.2.11)$$

Now, by (4.2.9) and (4.2.11) we have proved $\mathfrak{M} \models \chi(b_0, b_2, \bar{s})$.

Then, by the induction hypothesis $\psi(b_0, \bar{s})$ we conclude

$$\mathfrak{M} \models (b_0 = b_2) \ .$$

Then $\mathfrak{M} \models (b_0+1 = b_2+1)$ and by (4.2.4)

$$\mathfrak{M} \models (b_0+1 = b_1) \quad \text{in Case 2 .}$$

End of Case 2.

By Cases 1 and 2 we have proved $\mathfrak{M} \models (b_0+1 = b_1)$. Thus, by (4.2.2), we have proved $\mathfrak{M} \models \psi(b_0+1, \bar{s})$.

By the choice of b_0 we have proved

$$\mathfrak{M} \models \forall z_0 [\psi(z_0, \bar{s}) \rightarrow \psi(z_0+1, \bar{s})] \ .$$

End of Step 2.

By Steps 1 and 2 we have

$$\mathfrak{M} \models (\psi(0, \bar{s}) \wedge \forall z_0 [\psi(z_0, \bar{s}) \rightarrow \psi(z_0+1, \bar{s})]) \ . \qquad (*)$$

By $\mathfrak{M} \models \mathrm{IA}$ and by $\psi(z_0, \bar{s})^+_{z_0} \in \mathrm{IA}$ we have

$$\mathfrak{M} \models \psi(z_0, \bar{s})^+_{z_0} \ . \qquad (**)$$

By (*) and (**) we have $\mathfrak{M} \models \forall z_0 \, \psi(z_0, \bar{s})$.

Thus we have proved $\mathfrak{M} \models \psi(\bar{s}, 1')$.

QED (Claim 9.4.4.2)

Claim 9.4.4.3

$\mathfrak{M} \models \forall z_0 \, [s_2(z_0) = s_2(0)]$.

Proof of Claim 9.4.4.3 :

Let $\psi(z_0)$ be the formula $[s_2(z_0) = s_2(0)]$.

We shall prove $\forall z_0 \, \psi(z_0)$ by induction on z_0 .

Step 1 : $\mathfrak{M} \models \psi(0)$ trivially holds.

Step 2 : Now we prove $\mathfrak{M} \models \forall z_0 [\, \psi(z_0) \rightarrow \psi(z_0+1)\,]$. :

Let $b \in T$. Assume $\psi(b)$ that is assume $\mathfrak{M} \models [s_2(b) = s_2(0)]$.

Then, by the definition of a trace (see Definition 7(i)) there is a label i_m of some command $(i_m : u_m)$ of the program p_1 such that $s_5(b) = i_m$. That is $p_1 = \langle \ldots, (i_m : u_m), \ldots \rangle$ and $s_5(b) = i_m$.

By the definition of p_1 , u_m cannot be of the form " $x_2 \leftarrow \tau$ " .

Actually x_2 does not occur on the left side of any assignment in p_1 . This and the definition of a trace imply

$\mathfrak{M} \models [s_2(b+1) = s_2(b)]$.

Then, by the induction hypothesis $\mathfrak{M} \models [\, s_2(b) = s_2(0)]$ we have

$\mathfrak{M} \models [s_2(b+1) = s_2(0)]$.

We proved $\mathfrak{M} \models \forall z_0 [\, \psi(z_0) \rightarrow \psi(z_0+1)\,]$.

End of Step 2.

By Steps 1 and 2 we have proved

$$\mathfrak{M} \models (\, \psi(0) \wedge \forall z_0 [\psi(z_0) \rightarrow \psi(z_0+1)]\,) \quad . \qquad (*)$$

By $\mathfrak{M} \models$ IA and by $\psi(z_0)^+_{z_0} \in$ IA we have

$$\mathfrak{M} \models \psi(z_0)^+_{z_0} \quad . \qquad (**)$$

From (*) and (**) we conclude $\mathfrak{M} \models \forall z_0 \, \psi(z_0)$.

QED (Claim 9.4.4.3)

Claim 9.4.4.4

$\mathfrak{M} \models \forall z_0 \forall z_1 \, [\, s_5(z_0) = s_5(z_1) = 3' \rightarrow z_0 = z_1 \,]$.

Proof of Claim 9.4.4.4 :

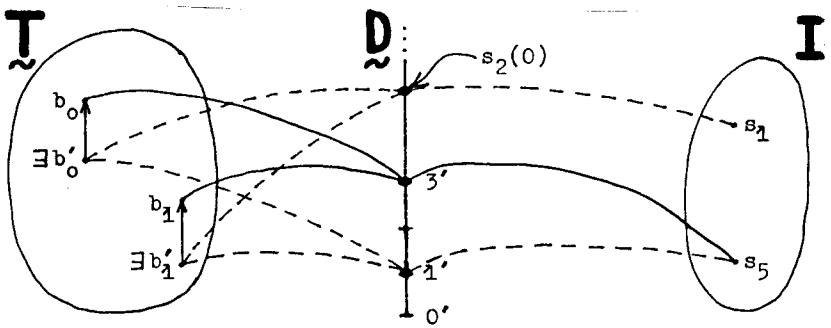


FIGURE 5

Let b_0 , $b_1 \in T$ be arbitrary. Assume $s_5(b_0) = s_5(b_1) = 3'$.

Then, by Lemma 9.4.3 and by Th $\models (0' \neq 3')$ there are $b_0' , b_1' \in T$ such that $b_0 = b_0'+1$ and $b_1 = b_1'+1$.

By the definition of a trace and that of p_1 we have

$[\, s_5(b_0') \in \{0', 2', 3', 4'\} \implies s_5(b_0) = s_5(b_0'+1) \in \{1', 2', 4'\}\,]$.

Th $\models$ ($3' \notin \{1', 2', 4'\}$) . Hence $s_5(b_0') = 1'$. The same argument yields $s_5(b_1') = 1'$.

Thus we have

$$s_5(b_0') = s_5(b_1') = 1' \quad . \qquad (4.4.1)$$

Since $s_5(b_0') = 1'$ and $s_5(b_0'+1) = s_5(b_0) = 3'$, by the definition of a trace we have

$s_1(b_0') = s_2(b_0')$.

Similarly $s_1(b_1') = s_2(b_1')$.

By Claim 9.4.4.3, $s_2(b_0') = s_2(0) = s_2(b_1')$. Therefore

$$s_1(b_0') = s_2(b_0') = s_2(b_1') = s_1(b_1') \quad . \qquad (4.4.2)$$

By (4.4.1) and by (4.4.2) we have

$\mathfrak{M} \models (\, s_5(b_0') = s_5(b_1') = 1' \wedge s_1(b_0') = s_1(b_1')\,)$.

Thus, by Claim 9.4.4.2, $b_0' = b_1'$. Hence $b_0 = b_0'+1 = b_1'+1 = b_1$.

QED (Claim 9.4.4.4)

Corollary 9.4.4.5

$\mathfrak{M} \models \varphi(\bar{s},3')$. That is

$\mathfrak{M} \models \forall z_0 \forall z_1 \; (\; [s_5(z_0)=s_5(z_1)=3' \wedge s_1(z_0)=s_1(z_1)] \rightarrow z_0=z_1 \;)$.

Proof of Corollary 9.4.4.5 :

This is an immediate consequence of Claim 9.4.4.4.

QED (Corollary 9.4.4.5)

Claim 9.4.4.6

$\mathfrak{M} \models \varphi(\bar{s},2')$. That is

$\mathfrak{M} \models \forall z_0 \forall z_1 \; (\; [s_5(z_0)=s_5(z_1)=2' \wedge s_1(z_0)=s_1(z_1)] \rightarrow z_0=z_1 \;)$.

Proof of Claim 9.4.4.6 :

Let $\chi(z_0,z_1,\bar{s})$ be the formula $[\, s_5(z_0)=s_5(z_1)=2' \wedge s_1(z_0)=s_1(z_1) \,]$.

Let $\psi(z_0,\bar{s})$ be the formula $\forall z_1 \; (\; \chi(z_0,z_1,\bar{s}) \rightarrow z_0=z_1 \;)$.

We shall prove $\mathfrak{M} \models \forall z_0 \; \psi(z_0,\bar{s})$ by induction on z_0 .

Step 1 : $\mathfrak{M} \models \psi(0,\bar{s})$ by Lemma 9.4.2 and by Th $\models (0' \neq 2')$.

Step 2 : We shall prove $\mathfrak{M} \models \forall z_0 \, [\, \psi(z_0,\bar{s}) \rightarrow \psi(z_0+1,\bar{s}) \,]$. :

Let $b_0 \in T$ be arbitrary. Assume $\mathfrak{M} \models \psi(b_0,\bar{s})$.

We shall prove $\mathfrak{M} \models \psi(b_0+1,\bar{s})$.

To see $\mathfrak{M} \models \psi(b_0+1,\bar{s})$, let $b_1 \in T$ be arbitrary and assume

$\mathfrak{M} \models \chi(b_0+1,b_1,\bar{s})$ that is we assume

$$\mathfrak{M} \models [\, s_5(b_0+1) = s_5(b_1) = 2' \wedge s_1(b_0+1) = s_1(b_1) \,] \; . \qquad (4.6.1)$$

It is enough to prove $\mathfrak{M} \models (b_0+1=b_1)$. See Figure 6 !

By Lemma 9.4.3, by $s_5(b_1) = 2'$ and by Th $\models (0' \neq 2')$, there is a $b_2 \in T$ such that $b_1 = b_2+1$. (4.6.2)

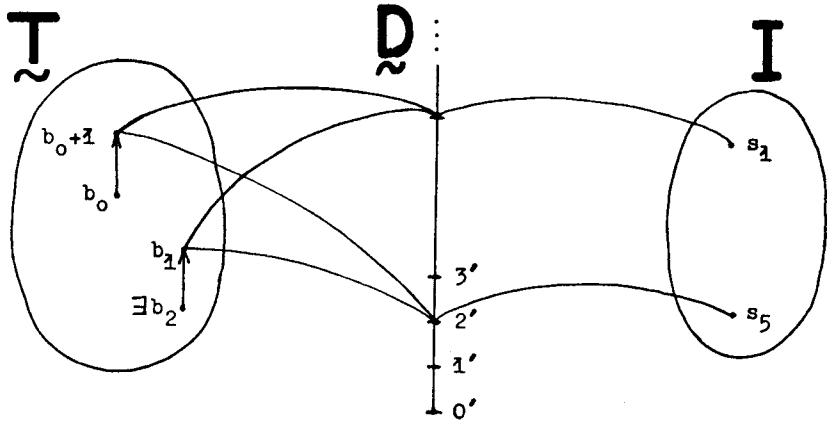


FIGURE 6

Claim 9.4.4.6.1 : $s_5(b_0) \in \{2',3'\}$ and $s_5(b_2) \in \{2',3'\}$.

Proof: By (4.6.1), we have

$$s_5(b_0+1) = 2' \quad \text{and} \quad s_5(b_2+1) = s_5(b_1) = 2' \qquad (4.6.3)$$

($s_5(b_2+1) = s_5(b_1)$ by (4.6.2); $s_5(b_1) = 2'$ by (4.6.1))

Let $k \in T$. The hypothesis $s_5(k) \in \{0', 1', 4'\}$ implies $s_5(k+1) \in \{1',3',4'\}$ by the definition of a trace.

Th $\models (2' \notin \{1',3',4'\})$. Thus

$(\forall k \in T) \left[\, s_5(k) \notin \{2',3'\} \longrightarrow s_5(k+1) \neq 2' \, \right]$. Then, by (4.6.3),

$s_5(b_0) \in \{2',3'\}$ and $s_5(b_2) \in \{2',3'\}$.

QED (Claim 9.4.4.6.1)

Claim 9.4.4.6.2 : $s_5(b_0) = s_5(b_2)$.

Proof of Claim 9.4.4.6.2 :

By Claim 9.4.4.6.1, $s_5(b_0) \in \{2',3'\}$ and $s_5(b_2) \in \{2',3'\}$.

Case 1 : Assume $s_5(b_0) = 2'$.

Then, by $s_5(b_0+1) = 2'$ (see (4.6.1)) and by the definition of a trace we have

$s_1(b_0+\bar{1}) = \text{suc}\ s_1(b_0)$.

Assume $s_5(b_2) = 3'$.

Then, by the definition of a trace, we have

$s_1(b_1) = s_1(b_2+\bar{1}) = 0'$.
(4.6.2)

Now

$0' = s_1(b_1) = s_1(b_0+\bar{1}) = \text{suc}\ s_1(b_0)$.
(4.6.1)

By $\mathfrak{M} \models \text{Th}$, this is a contradiction !

Thus $s_5(b_2) = 2'$.

Thus $s_5(b_0) = s_5(b_2)$.

Case 2 : Assume $s_5(b_0) = 3'$.

Then, by the definition of a trace, we have $s_1(b_0+\bar{1}) = 0'$.

Assume $s_5(b_2) = 2'$.

Then, by $s_5(b_2+\bar{1}) = s_5(b_1) = 2'$ and by the definition of a trace,
(4.6.2) (4.6.1)

$s_1(b_1) = s_1(b_2+\bar{1}) = \text{suc}\ s_1(b_2)$.
(4.6.2)

Now

$0' = s_1(b_0+\bar{1}) = s_1(b_1) = \text{suc}\ s_1(b_2)$.
(4.6.1)

By $\mathfrak{M} \models \text{Th}$, this is a contradiction !

Thus $s_5(b_2) = 3'$.

Thus $s_5(b_0) = s_5(b_2)$.

QED (Claim 9.4.4.6.2)

Now, by Claims 9.4.4.6.1 and 9.4.4.6.2, we have

$s_5(b_0) = s_5(b_2) \in \{2', 3'\}$. (4.6.3.1)

Case 1 : $s_5(b_0) = s_5(b_2) = 3'$.

Then, by Claim 9.4.4.4, $b_0 = b_2$. Then $b_0+\bar{1} = b_2+\bar{1} = b_1$. Thus
(4.6.2)

$\mathfrak{M} \models (b_0+\bar{1}=b_1)$.

Case 2 : $s_5(b_0) = s_5(b_2) = 2'$.

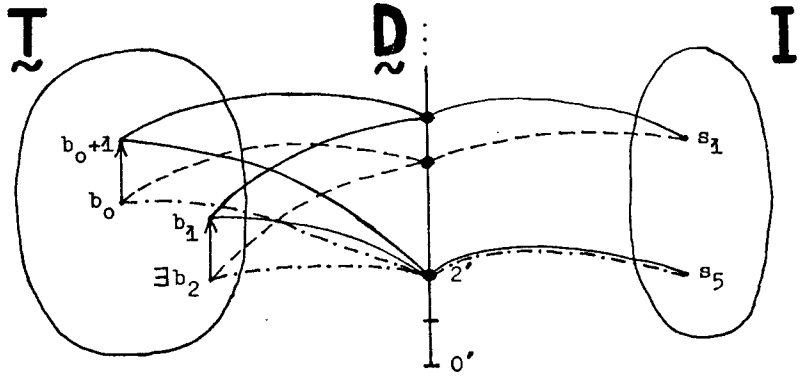


FIGURE 7

By $s_5(b_2) = s_5(b_1) = 2'$ and by the definition of a trace we have
(4.6.1)

$s_1(b_1) = s_1(b_2+\bar{1}) = \text{suc}\ s_1(b_2)$. (4.6.4)
(4.6.2)

By $s_5(b_0) = s_5(b_0+\bar{1}) = 2'$ and by the definition of a trace we have
(4.6.1)

$s_1(b_0+\bar{1}) = \text{suc}\ s_1(b_0)$. (4.6.5)

By (4.6.1),

$s_1(b_0+\bar{1}) = s_1(b_1)$. (4.6.6)

Now

$\text{suc } s_1(b_2) = s_1(b_1) = s_1(b_0+\bar{1}) = \text{suc } s_1(b_0)$, that is

(4.6.4) (4.6.6) (4.6.5)

$\mathfrak{M} \models (\text{suc } s_1(b_0) = \text{suc } s_1(b_2))$.

Now, by Th $\models \forall x_0 \forall x_1 [\text{suc } x_0 = \text{suc } x_1 \rightarrow x_0 = x_1]$ we have

$\mathfrak{M} \models (s_1(b_0) = s_1(b_2))$.

Thus we have

$\mathfrak{M} \models (s_5(b_0) = s_5(b_2) = 2' \wedge s_1(b_0) = s_1(b_2))$, that is

$\mathfrak{M} \models \chi(b_0, b_2, \bar{s})$.

This together with the induction hypothesis $\mathfrak{M} \models \psi(b_0, \bar{s})$ imply

$\mathfrak{M} \models (b_0 = b_2)$.

Then $b_0+\bar{1} = b_2+\bar{1} = b_1$ that is $\mathfrak{M} \models (b_0+\bar{1} = b_1)$.

End of Case 2.

By Cases 1 and 2 above and by (4.6.3.1) we have proved $b_0+\bar{1} = b_1$, and by the choice of b_1 , then $\mathfrak{M} \models \psi(b_0+\bar{1}, \bar{s})$.

End of Step 2.

By Steps 1 and 2 we have proved

$\mathfrak{M} \models (\psi(0, \bar{s}) \wedge \forall z_0 [\psi(z_0, \bar{s}) \rightarrow \psi(z_0+\bar{1}, \bar{s})])$. (*)

By $\mathfrak{M} \models$ IA and by $\psi(z_0, \bar{s})^+_{z_0} \in$ IA we have

$\mathfrak{M} \models \psi(z_0, \bar{s})^+_{z_0}$. (**)

Thus $\mathfrak{M} \models \forall z_0 \, \psi(z_0, \bar{s})$ by (*) and (**). Observing that the formulas $\varphi(\bar{s}, 2')$ and $\forall z_0 \, \psi(z_0, \bar{s})$ coincide completes the proof.

QED (Claim 9.4.4.6)

By Claims 9.4.4.1, 9.4.4.2, 9.4.4.6, and by Corollary 9.4.4.5 we have proved

$\mathfrak{M} \models \varphi(\bar{s}, i)$ for every $i \in \{0', 1', 2', 3'\}$.

QED (Claim 9.4.4)

Claim 9.4.5

$\mathfrak{M} \models \forall z_0 [s_5(z_0) = 4' \rightarrow s_4(z_0) = s_3(z_0)]$.

Proof of Claim 9.4.5 :

Claim 9.4.5.1

If $\bar{s}$ terminates p_1 in $\mathfrak{M}$ then

$\mathfrak{M} \models \exists z_0 [s_5(z_0) = 3']$.

Proof of Claim 9.4.5.1 :

Assume $\bar{s}$ terminates p_1 in $\mathfrak{M}$.

Assume

$\mathfrak{M} \models \forall z_0 [s_5(z_0) \neq 3']$. (5.1.1)

Let $\varphi(z_0)$ be the formula

$[s_5(z_0) = 0' \vee s_5(z_0) = 1']$.

We shall prove $\forall z_0 \, \varphi(z_0)$ by induction on z_0 .

Step 1: $\mathfrak{M} \models \varphi(0)$ by Lemma 9.4.2.

Step 2: We prove $\mathfrak{M} \models \forall z_0 [\varphi(z_0) \rightarrow \varphi(z_0+\bar{1})]$. :

Let $b \in T$ be arbitrary. Assume $\mathfrak{M} \models \varphi(b)$ that is we assume $s_5(b) \in \{0', 1'\}$.

Then, by the definition of a trace, $s_5(b+\bar{1}) \in \{1', 3'\}$.

By our hypothesis (5.1.1), $s_5(b+\bar{1}) \neq 3'$. Then $s_5(b+\bar{1}) = 1'$.

Then $\mathfrak{M} \models \varphi(b+\bar{1})$.

We proved $\mathfrak{M} \models \forall z_0 [\varphi(z_0) \rightarrow \varphi(z_0+\bar{1})]$.

End of Step 2.

By Steps 1 and 2 we have proved

$\mathfrak{M} \models \varphi(0) \wedge \forall z_0 [\varphi(z_0) \rightarrow \varphi(z_0+\bar{1})]$. (*)

By $\mathfrak{M} \models$ IA and by $\varphi(z_0)^+_{z_0} \in$ IA we have

$\mathfrak{M} \models \varphi(z_0)^+_{z_0}$. (**)

By (*) and (**) above we have proved $\mathfrak{M} \models \forall z_0 \, \varphi(z_0)$.

Hence $\mathfrak{M} \models \forall z_0 [\, s_5(z_0) \neq 4' \,]$ by Th $\models (4' \notin \{0', 1'\})$.

Hence $\bar{s}$ does not terminate p_1 in $\mathfrak{M}$. This is a contradiction !

Thus our hypothesis $\mathfrak{M} \models \forall z_0 [\, s_5(z_0) \neq 3' \,]$ is false.

We have proved $\mathfrak{M} \models \exists z_0 [\, s_5(z_0) = 3' \,]$.

QED (Claim 9.4.5.1)

Corollary 9.4.5.2

If $\bar{s}$ terminates p_1 in $\mathfrak{M}$ then

$\mathfrak{M} \models (\exists!\, z_0)\ s_5(z_0) = 3'$.

Proof: This is a consequence of Claims 9.4.4.4 and 9.4.5.1.

QED (Corollary 9.4.5.2)

Claim 9.4.5.3

$\mathfrak{M} \models \forall z_0 [\, (\, s_5(z_0)=1' \vee s_5(z_0)=3' \,) \rightarrow s_4(z_0)=s_2(0) \,]$.

Proof of Claim 9.4.5.3 :

Let $\varphi(z_0, \bar{s})$ be the formula

$[\, (\, s_5(z_0)=1' \vee s_5(z_0)=3' \,) \rightarrow s_4(z_0)=s_2(0) \,]$.

We shall prove $\forall z_0\ \varphi(z_0, \bar{s})$ by induction on z_0 .

Step 1 : By Lemma 9.4.2 and by Th $\models (0' \notin \{1', 3'\})$ we have

$\mathfrak{M} \models \varphi(0, \bar{s})$ . .

Step 2 : We shall prove $\forall z_0 [\, \varphi(z_0, \bar{s}) \rightarrow \varphi(z_0+1, \bar{s}) \,]$.

Let $b \in T$ be arbitrary. Assume $\varphi(b, \bar{s})$. We shall prove $\varphi(b+1, \bar{s})$.

Assume $(\, s_5(b+1) = 1' \vee s_5(b+1) = 3' \,)$.

It is enough to prove $s_4(b+1) = s_2(0)$. (5.3.0)

Case 1 : Assume $s_5(b+1) = 1'$. (5.3.1)

Then $s_5(b) \in \{0', 1'\}$ since the assumption $s_5(b) \in \{2', 3', 4'\}$ would imply $s_5(b+1) \in \{2', 4'\}$ by the definition of a trace. But this is impossible by (5.3.1) and by Th $\models (1' \notin \{2', 4'\})$.

If $s_5(b)=0'$ then $s_4(b+1)=s_2(0)$ by the definition of a trace.

If $s_5(b)=1'$ then $s_4(b+1)=s_4(b)$ by the definition of a trace. By the induction hypothesis $\varphi(b, \bar{s})$ and by $s_5(b)=1'$

$s_4(b+1) = s_4(b) = s_2(0)$.

Case 2 : Assume $s_5(b+1) = 3'$. (5.3.2)

Then $s_5(b)=1'$ since the assumption $s_5(b) \in \{0', 2', 3', 4'\}$ would imply $s_5(b+1) \in \{1', 2', 4'\}$ by the definition of a trace. But this is impossible by (5.3.2) and by Th $\models (3' \notin \{1', 2', 4'\})$.

By $s_5(b)=1'$ and by the induction hypothesis $\varphi(b, \bar{s})$ we have

$s_4(b) = s_2(0)$.

By $s_5(b)=1'$ and by the definition of a trace, we have

$s_4(b+1) = s_4(b)$.

Thus $s_4(b+1) = s_4(b) = s_2(0)$.

Cases 1 and 2 together prove $s_4(b+1) = s_2(0)$. By (5.3.0), we have $\varphi(b+1, \bar{s})$. End of Step 2.

By Steps 1 and 2 we have proved

$\mathfrak{M} \models \varphi(0, \bar{s}) \wedge \forall z_0 [\, \varphi(z_0, \bar{s}) \rightarrow \varphi(z_0+1, \bar{s}) \,]$. (*)

By $\mathfrak{M} \models$ IA and by $\varphi(z_0, \bar{s})^+_{z_0} \in$ IA we have

$\mathfrak{M} \models \varphi(z_0, \bar{s})^+_{z_0}$. (**)

By (*) and (**) we have $\mathfrak{M} \models \forall z_0\ \varphi(z_0, \bar{s})$.

QED (Claim 9.4.5.3)

Claim 9.4.5.4

Let $k \in T$. Assume $s_5(k)=3'$. Then

$\mathfrak{M} \models \forall z_0 \left[(s_5(z_0)=2' \vee s_5(z_0)=4') \rightarrow s_3(z_0)=s_3(k) \right]$.

Proof of Claim 9.4.5.4 :

Let $k \in T$. Assume $s_5(k)=3'$.

Let $\varphi(z_0,\bar{s})$ be the formula

$\left[(s_5(z_0)=2' \vee s_5(z_0)=4') \rightarrow s_3(z_0)=s_3(k) \right]$.

We shall prove $\forall z_0 \, \varphi(z_0,\bar{s})$ by induction on z_0 .

Step 1 : By Lemma 9.4.2 and by Th $\models (0' \notin \{2',4'\})$ we have

$\mathfrak{M} \models \varphi(0,\bar{s})$.

Step 2 : We shall prove $\forall z_0 \left[\varphi(z_0,\bar{s}) \rightarrow \varphi(z_0+1,\bar{s}) \right]$.:

Let $b \in T$. Assume $\varphi(b,\bar{s})$. We prove $\varphi(b+1,\bar{s})$. :
Assume $(s_5(b+1) = 2' \vee s_5(b+1) = 4')$.
It is enough to prove $s_3(b+1) = s_3(k)$. (5.4.0)
Case 1 : Assume $s_5(b+1) = 2'$. (5.4.1)
Then $s_5(b) \in \{3',2'\}$ since the assumption $s_5(b) \in \{0',1',4'\}$
would imply $s_5(b+1) \in \{1',3',4'\}$ by the definition of a trace.
But this is impossible by (5.4.1) and by Th $\models (2' \notin \{1',3',4'\})$.

If $s_5(b)=3'$ then by the definition of a trace
$s_3(b)=s_3(b+1)$.
By Claim 9.4.4.4 and by $s_5(b)=3'$ we have $b=k$. Thus
$s_3(k)=s_3(b)=s_3(b+1)$.

If $s_5(b)=2'$ then $s_3(b)=s_3(b+1)$ by the definition of a trace.
By the induction hypothesis $\varphi(b,\bar{s})$ we have
$s_3(k) = s_3(b) = s_3(b+1)$.

Case 2 : Assume $s_5(b+1) = 4'$. (5.4.2)
Then $s_5(b) \in \{2',4'\}$ since the assumption $s_5(b) \in \{0',1',3'\}$
would imply $s_5(b+1) \in \{1',2',3'\}$ by the definition of a trace.
But this is impossible by (5.4.2) and by Th $\models (4' \notin \{1',2',3'\})$.

Then $s_3(b) = s_3(b+1)$ by the definition of a trace.
By $s_5(b) \in \{2',4'\}$ and by our induction hypothesis $\varphi(b,\bar{s})$ we have
$s_3(k) = s_3(b) = s_3(b+1)$.

End of 2.2.

By Cases 1, 2 and by (5.4.0) we proved $\forall z_0 \left[\varphi(z_0,\bar{s}) \rightarrow \varphi(z_0+1,\bar{s}) \right]$.

End of Step 2.

By Steps 1 and 2 we have proved

$\mathfrak{M} \models \varphi(0,\bar{s}) \wedge \forall z_0 \left[\varphi(z_0,\bar{s}) \rightarrow \varphi(z_0+1,\bar{s}) \right]$. (*)

By $\mathfrak{M} \models$ IA and by $\varphi(z_0,\bar{s})^+_{z_0} \in$ IA we have

$\mathfrak{M} \models \varphi(z_0,\bar{s})^+_{z_0}$. (**)

By (*) and by (**) we have $\mathfrak{M} \models \forall z_0 \, \varphi(z_0,\bar{s})$.

QED (Claim 9.4.5.4)

Claim 9.4.5.5

If $\bar{s}$ terminates p_1 in $\mathfrak{M}$ then

$\mathfrak{M} \models \forall z_0 (s_5(z_0)=1' \rightarrow \exists z_1 [s_5(z_1)=2' \wedge s_1(z_0)=s_1(z_1) \wedge s_3(z_0)=s_4(z_1)])$.

Proof of Claim 9.5.5.5 :

Assume $\bar{s}$ terminates p_1 in $\mathfrak{M}$.

Let $\varphi(z_0,z_1,\bar{s})$ be the formula

$\left[s_5(z_1)=2' \wedge s_1(z_0)=s_1(z_1) \wedge s_3(z_0)=s_4(z_1) \right]$.

Let $\psi(z_0,\bar{s})$ be the formula

$(s_5(z_0)=1' \rightarrow \exists z_1 \, \varphi(z_0,z_1,\bar{s}))$.

We shall prove $\forall z_0 \, \psi(z_0,\bar{s})$ by induction on z_0 .

Step 1 : $\mathfrak{M} \models \psi(0,\bar{s})$ by Lemma 9.4.2 and by Th $\models (0' \neq 1')$.

Step 2: We shall prove $\forall z_0 [\psi(z_0,\bar{s}) \rightarrow \psi(z_0+\bar{1},\bar{s})]$. :

Let $b_0 \in T$ be arbitrary. Assume $\psi(b_0,\bar{s})$.

We shall prove $\psi(b_0+\bar{1},\bar{s})$. :

Assume $s_5(b_0+\bar{1})=\bar{1}'$. Then

$$s_5(b_0) \in \{0',1'\} \qquad (5.5.0)$$

since otherwise we would have $s_5(b_0+\bar{1}) \in \{2',4'\}$ by the definition of a trace. But this is impossible by $s_5(b_0+\bar{1})=\bar{1}'$ and by Th $\models$ ($\bar{1}' \notin \{2',4'\}$) .

Case $\bar{1}$: $s_5(b_0)=0'$.

Since $\bar{s}$ terminates $p_{\bar{1}}$ in $\mathfrak{M}$ by our hypothesis, by Claim 9.4.5.$\bar{1}$ there is a $b_3 \in T$ such that $s_5(b_3)=3'$.

Let $b_2 \overset{df}{=} b_3+\bar{1}$. Then, by the definition of a trace,

$$s_5(b_2) = 2' . \qquad (5.5.\bar{1})$$

Further $s_4(b_2) = s_4(b_3+\bar{1}) = s_4(b_3) = s_2(0)$. (5.5.$\bar{1}$.1)

(def. of a trace) ($s_5(b_3)=3'$, Claim 9.4.5.3)

$s_3(b_0+\bar{1}) = s_2(0)$ by $s_5(b_0)=0'$ and by the definition of a trace.

Thus we have (5.5.$\bar{1}$.1)

$$s_3(b_0+\bar{1}) = s_2(0) = s_4(b_2) . \qquad (5.5.2)$$

Further it is easy to see that

$$s_{\bar{1}}(b_0+\bar{1}) = 0' = s_{\bar{1}}(b_3+\bar{1}) = s_{\bar{1}}(b_2) \qquad (5.5.3)$$

($s_5(b_0)=0'$) ($s_5(b_3)=3'$)

by the definition of a trace.

So far we have seen $\varphi(b_0+\bar{1},b_2,\bar{s})$ by (5.5.$\bar{1}$), (5.5.2), and by (5.5.3). Thus we have $\psi(b_0+\bar{1},\bar{s})$.

Case 2 : $s_5(b_0)=\bar{1}'$.

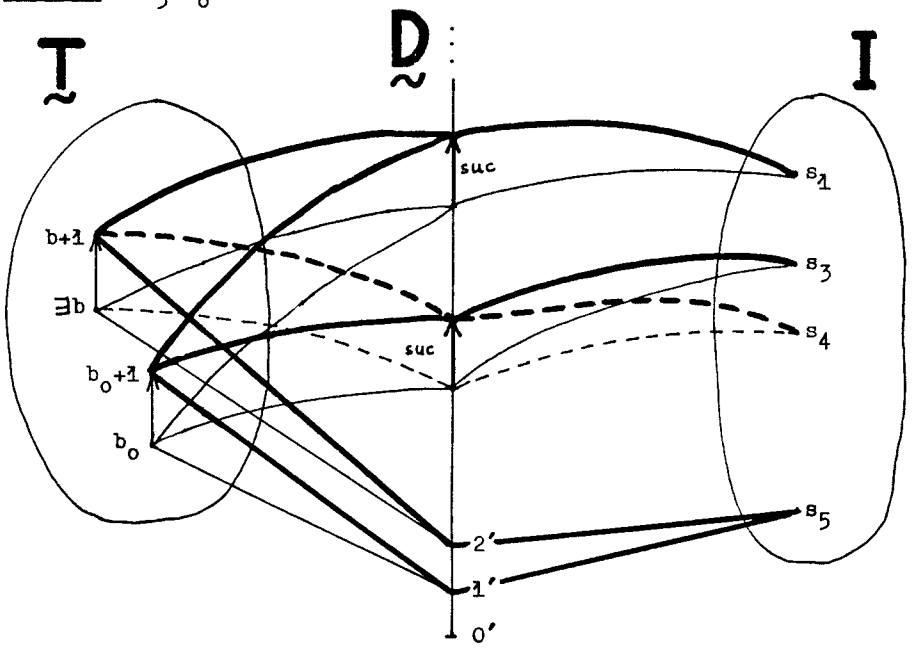


FIGURE 8

By the induction hypothesis $\psi(b_0,\bar{s})$ there is a $b \in T$ such that

$$\varphi(b_0,b,\bar{s}) . \qquad (5.5.4)$$

Then, by $s_5(b)=2'$ (see (5.5.4)) and by the definition of a trace,

$$s_5(b+\bar{1}) \in \{2',4'\} .$$

Assume $s_5(b+\bar{1})=4'$.

By $s_5(b)=2'$, by $s_5(b+\bar{1})=4'$ and by the definition of a trace and by Claim 9.4.4.3, we have:

$$s_{\bar{1}}(b_0) = s_{\bar{1}}(b) = s_2(b) = s_2(0) = s_2(b_0) .$$

(5.5.4) (Claim 9.4.4.3)

Then $s_5(b_0+\bar{1}) = 3'$ by the definition of a trace.

By $\mathrm{Th} \models (\bar{1}' \neq 3')$, this is a contradiction !

Thus $s_5(b+\bar{1}) = 2'$. (5.5.5)

Now

$$s_{\bar{1}}(b_0+\bar{1}) = \mathrm{suc}\, s_{\bar{1}}(b_0) = \mathrm{suc}\, s_{\bar{1}}(b) = s_{\bar{1}}(b+\bar{1}) \quad . \qquad (5.5.6)$$

(def. of a trace $s_5(b_0)=s_5(b_0+\bar{1})=\bar{1}'$) (5.5.4) (def. of a trace $s_5(b)=s_5(b+\bar{1})=2'$)

$$s_3(b_0+\bar{1}) = \mathrm{suc}\, s_3(b_0) = \mathrm{suc}\, s_4(b) = s_4(b+\bar{1}) \quad . \qquad (5.5.7)$$

By (5.5.5), (5.5.6), and (5.5.7) we have $\psi(b_0+\bar{1}, b+\bar{1}, \bar{s})$.

Thus we have $\psi(b_0+\bar{1}, \bar{s})$.

End of Case 2.

By Cases 1, 2 and by (5.5.0) we have seen $\psi(b_0+\bar{1}, \bar{s})$.

Thus we proved $\forall z_0 \left[\psi(z_0, \bar{s}) \rightarrow \psi(z_0+\bar{1}, \bar{s}) \right]$.

End of Step 2.

By Steps 1 and 2 we have proved

$$\mathfrak{M} \models \psi(0, \bar{s}) \wedge \forall z_0 \left[\psi(z_0, \bar{s}) \rightarrow \psi(z_0+\bar{1}, \bar{s}) \right] . \qquad (*)$$

By $\mathfrak{M} \models \mathrm{IA}$ and by $\psi(z_0, \bar{s})^+_{z_0} \in \mathrm{IA}$ we have

$$\mathfrak{M} \models \psi(z_0, \bar{s})^+_{z_0} \quad . \qquad (**)$$

By (*) and (**) we have $\mathfrak{M} \models \forall z_0\, \psi(z_0, \bar{s})$.

QED (Claim 9.4.5.5)

By passing we note that Claim 9.4.5.6 below sais

$$\mathrm{Th} \cup \mathrm{IA} \models \Diamond(p_{\bar{1}}, \mathrm{TRUE}) \rightarrow \Diamond(p_{\bar{1}}, x_3 = x_4) \quad .$$

Claim 9.4.5.6

If $\bar{s}$ terminates $p_{\bar{1}}$ in $\mathfrak{M}$ then

$$\mathfrak{M} \models \exists z_0 \left[s_5(z_0) = 4' \wedge s_4(z_0) = s_3(z_0) \right] \quad .$$

Proof of Claim 9.4.5.6 :

Assume $\bar{s}$ terminates $p_{\bar{1}}$ in $\mathfrak{M}$. Then, by Claim 9.4.5.1, there is a $b_3 \in T$ such that

$$s_5(b_3) = 3' \quad . \qquad (5.6.1)$$

Then, by Lemma 9.4.3 and by $\mathrm{Th} \models (3' \neq 0')$, there is a b_3' such that

$$b_3 = b_3' + \bar{1} \quad . \qquad (5.6.2)$$

Now

$$s_5(b_3') = \bar{1}' \qquad (5.6.3)$$

since otherwise we would have $s_5(b_3') \in \{0', 2', 3', 4'\}$ which would imply $s_5(b_3) = s_5(b_3'+\bar{1}) \in \{\bar{1}', 2', 4'\}$ by the definition of a trace.

($s_5(b_3) = s_5(b_3'+\bar{1})$: (5.6.2))

But this is impossible by (5.6.1) and by $\mathrm{Th} \models (3' \notin \{\bar{1}', 2', 4'\})$.

By Claim 9.4.5.5 and by (5.6.3) we have

$$\exists b_2 \left[s_5(b_2) = 2' \wedge s_{\bar{1}}(b_3') = s_{\bar{1}}(b_2) \wedge s_3(b_3') = s_4(b_2) \right] \quad . \qquad (5.6.4)$$

We define

$$b_4 \overset{df}{=} b_2 + \bar{1} \quad . \qquad (5.6.5)$$

Claim 9.4.5.6.1 : $s_5(b_4) = 4'$.

Proof:

$$s_{\bar{1}}(b_3') = s_2(b_3') = s_2(0) \qquad (5.6.6)$$

by $s_5(b_3') = \bar{1}'$ ((5.6.3)), by $s_5(b_3'+\bar{1}) = s_5(b_3) = 3'$ ((5.6.2), (5.6.1)), by the definition of a trace,

and by Claim 9.4.4.3. Now

$$s_{\bar{1}}(b_2) = s_{\bar{1}}(b_3') = s_2(0) = s_2(b_2) \quad . \qquad (5.6.7)$$

((5.6.4), (5.6.6), Claim 9.4.4.3)

Then $s_5(b_2) = 2'$ (5.6.4), $s_1(b_2) = s_2(b_2)$ (5.6.7), and the definition of a trace imply $s_5(b_4) = s_5(b_{2+1}) = 4'$ (5.6.5).

QED (Claim 9.4.5.6.1)

Claim 9.4.5.6.2 : $s_4(b_4) = s_3(b_4)$.

Proof:

$s_4(b_4) = s_4(b_{2+1}) = s_4(b_2) = s_3(b'_3) = s_3(b'_{3+1}) = s_3(b_3) = s_3(b_4)$

since:

$s_4(b_4)=s_4(b_{2+1})$ by the definition of b_4 (see (5.6.5)),

$s_4(b_{2+1})=s_4(b_2)$ by $s_5(b_2)=2'$ (see (5.6.4)), by $s_5(b_{2+1})=s_5(b_4)=4'$ (see (5.6.5) and Claim 9.4.5.6.1), and by the definition of a trace,

$s_4(b_2)=s_3(b'_3)$ by (5.6.4),

$s_3(b'_3)=s_3(b'_{3+1})$ by $s_5(b'_3)=1'$ (see (5.6.3)), by $s_5(b'_{3+1})=s_5(b_3)=3'$ (see (5.6.2) and (5.6.1)), and by the definition of a trace,

$s_3(b'_{3+1})=s_3(b_3)$ (see (5.6.2)),

$s_3(b_3)=s_3(b_4)$ since Claim 9.4.5.4 can be applied by $s_5(b_4)=4'$ (see Claim 9.4.5.6.1) and by $s_5(b_3)=3'$ ((5.6.1)).

Thus we have seen $s_4(b_4) = s_3(b_4)$.

QED (Claim 9.4.5.6.2)

By Claims 9.4.5.6.1 and 9.4.5.6.2 we have $\mathfrak{M} \models (s_5(b_4) = 4' \wedge s_4(b_4) = s_3(b_4))$. This completes the proof.

QED (Claim 9.4.5.6)

Claim 9.4.5.7

$$\mathfrak{M} \models \forall z_0 \forall z_1 \left[s_5(z_0)=s_5(z_1)=4' \rightarrow \bigwedge_{i=1}^{4} (s_i(z_0)=s_i(z_1)) \right] .$$

Proof of Claim 9.4.5.7 :

Let $\psi(z_0,z_1,\bar{s})$ be the formula

$$\left[s_5(z_1)=2' \wedge s_1(z_1)=s_2(z_1) \wedge \bigwedge_{i=1}^{4} (s_i(z_0)=s_i(z_1)) \right] .$$

Let $\varphi(z_0,\bar{s})$ be the formula

$$\left[s_5(z_0) = 4' \rightarrow \exists z_1 \; \psi(z_0,z_1,\bar{s}) \right] .$$

Claim 9.4.5.7.1 : $\forall z_0 \; \varphi(z_0,\bar{s})$.

Proof: We shall prove $\forall z_0 \; \varphi(z_0,\bar{s})$ by induction on z_0

Step 1: $\varphi(0,\bar{s})$ is trivial by Lemma 9.4.2 and by Th $\models (0' \neq 4')$.

Step 2: We shall prove $\forall z_0 \left[\varphi(z_0,\bar{s}) \rightarrow \varphi(z_{0+1},\bar{s}) \right]$.

Let $b \in T$ be arbitrary. Assume $\varphi(b,\bar{s})$.

We want to prove $\varphi(b+1,\bar{s})$.

Assume $s_5(b+1) = 4'$.

Case 1 : $s_5(b) = 4'$. (5.7.1)

By $s_5(b) = 4'$ and by the definition of a trace, we have

$$\bigwedge_{i=1}^{5} (s_i(b+1) = s_i(b)) . \qquad (5.7.1.1)$$

By the induction hypothesis $\varphi(b,\bar{s})$, there is a $b_1 \in T$ such that

$$\psi(b,b_1,\bar{s}) . \qquad (5.7.2)$$

Then $\psi(b+1,b_1,\bar{s})$ follows from (5.7.1.1) by the following computation.:

$$\bigwedge_{i=1}^{4} \left[s_i(b+1) = s_i(b) = s_i(b_1) \right] .$$

(5.7.1.1)(5.7.2)

By (5.7.2), $s_5(b_1) = 2' \wedge s_1(b_1) = s_2(b_1)$. Then we have seen

$$s_5(b_1) = 2' \wedge s_1(b_1) = s_2(b_1) \wedge \bigwedge_{i=1}^{4} (s_i(b+1) = s_i(b_1)) .$$

Thus we have $\psi(b+1,b_1,\bar{s})$ by the definition of ψ. Thus we have $\varphi(b+1,\bar{s})$.

Case 2 : $s_5(b) \neq 4'$.

Then

$$s_5(b) = 2' \qquad (5.7.3)$$

since otherwise we would have $s_5(b) \in \{0', 1', 3'\}$ which would imply $s_5(b+1) \in \{1', 2', 3'\}$ by the definition of a trace. But this is impossible by $s_5(b+1) = 4'$ (5.7.0) and by Th $\vdash (4' \notin \{1', 2', 3'\})$.

Now the definition of a trace, $s_5(b)=2'$ and $s_5(b+1)=4'$ imply

$$s_1(b)=s_2(b) \text{ and } \bigwedge_{i=1}^{4} (s_i(b)=s_i(b+1)) \quad . \qquad (5.7.4)$$

By (5.7.3) and (5.7.4) we have $\varphi(b+1, b, \bar{s})$.

Then we also have $\varphi(b+1, \bar{s})$.

End of Case 2.

By Cases 1, 2 we have proved $\varphi(b+1, \bar{s})$. By the choice of b , we have seen $\forall z_0 [\varphi(z_0, \bar{s}) \rightarrow \varphi(z_0+1, \bar{s})]$.

End of Step 2.

By Steps 1 and 2 we have proved

$$\mathfrak{M} \models \varphi(0, \bar{s}) \wedge \forall z_0 [\varphi(z_0, \bar{s}) \rightarrow \varphi(z_0+1, \bar{s})] \quad . \qquad (*)$$

By $\mathfrak{M} \models$ IA and by $\varphi(z_0, \bar{s})^+_{z_0} \in$ IA we have

$$\mathfrak{M} \models \varphi(z_0, \bar{s})^+_{z_0} \quad . \qquad (**)$$

By (*) and (**) we have $\mathfrak{M} \models \forall z_0 \, \varphi(z_0, \bar{s})$.

QED (Claim 9.4.5.7.1)

Let b_0 , $b_1 \in T$. Assume $s_5(b_0) = s_5(b_1) = 4'$.

Claim 9.4.5.7.1 and $s_5(b_0) = 4'$ imply

$$\exists b_0' \; \varphi(b_0, b_0', \bar{s}) \quad . \qquad (5.7.5)$$

Claim 9.4.5.7.1 and $s_5(b_1) = 4'$ imply

$$\exists b_1' \; \varphi(b_1, b_1', \bar{s}) \quad . \qquad (5.7.6)$$

Then

$$s_5(b_0') = s_5(b_1') = 2' \qquad (5.7.7)$$

by (5.7.5) and (5.7.6) and

$$s_1(b_0') = s_1(b_1') \qquad (5.7.8)$$

since $s_1(b_0') = s_2(b_0') = s_2(0) = s_2(b_1') = s_1(b_1')$.

(5.7.5) Claim 9.4.4.3 (5.7.6)

Thus we have $s_5(b_0') = s_5(b_1') = 2' \wedge s_1(b_0') = s_1(b_1')$.

(5.7.7) (5.7.8)

Thus Claim 9.4.4.6 can be applied. Hence we have

$$b_0' = b_1' \quad . \qquad (5.7.9)$$

Thus $\bigwedge_{i=1}^{4} (s_i(b_0) = s_i(b_0') = s_i(b_1') = s_i(b_1))$.

(5.7.5) (5.7.9) (5.7.6)

Since b_0 , b_1 are arbitrary elements of T , we have proved

$$\mathfrak{M} \models \forall z_0 \forall z_1 \left[s_5(z_0)=s_5(z_1)=4' \rightarrow \bigwedge_{i=1}^{4} (s_i(z_0)=s_i(z_1)) \right] \quad .$$

QED (Claim 9.4.5.7)

Now we turn to prove Claim 9.4.5.

Let $b \in T$ be arbitrary. Assume $s_5(b) = 4'$.

Then, by Claim 9.4.5.6, there is a $b' \in T$ such that

$$[s_5(b') = 4' \wedge s_4(b') = s_3(b')] \quad .$$

Then, by Claim 9.4.5.7 and by $s_5(b) = s_5(b') = 4'$, we have

$$\bigwedge_{i=1}^{4} (s_i(b) = s_i(b')) \quad .$$

Then

$$s_3(b) = s_3(b') = s_4(b') = s_4(b) \quad .$$

QED (Claim 9.4.5)

Now we prove Claim 9.4 .

Recall that $\mathfrak{M} \models \Box(p_1, x_3=x_4)$ iff (for every trace $\bar{s}$ of p_1 in $\mathfrak{M}$, $(\forall b \in T)[\, s_5(b)=4' \rightarrow s_4(b)=s_3(b)\,]$), by definition.

Then, by Claim 9.4.5 and by the choice of $\bar{s}$ at the beginning of the present proof, we have $\mathfrak{M} \models \Box(p_1, x_3=x_4)$.

By the choice of $\mathfrak{M}$, we have

$\mathrm{Th} \cup \mathrm{IA} \models \Box(p_1, x_3=x_4)$.

QED (Claim 9.4)

QED ((v) $\not\Rightarrow$ (i))

Proof of (iv) $\not\Rightarrow$ (i) :

Let d , Th and φ be as in the proof of $\big((v) \not\Rightarrow (i)\big)$. . Then $\mathrm{Th} \not\vdash^{F} \varphi$ by the same ultraproduct construction as above.

The idea of the proof of $\mathrm{Th} \cup \mathrm{IA}^q \cup \mathrm{Pres} \models \varphi$ is the following: If we have Pres postulated about time structure then we can perform addition on time. Then we can say that "if φ is true at time $z_0 < z_1$ and χ is true at time z_1 then φ is true at time z_1+z_0 too" , i.e. if e.g. we execute the same program twice and z_1 is the time of the first termination then we can say that "if φ holds at time $z_0 < z_1$ then it will hold exactly z_0 time after z_1 again" .

Another way of proving these is by using Thm.4 in [19].

QED(Theorem 9)

Part $\big((i) \Longleftrightarrow (iii)\big)$ of Theorem 9 implies that the language $\langle \mathrm{DF}_d \,,\, \mathrm{Mod}(\mathrm{Ax}_o) \,,\, \models \rangle$ is reasonable enough, it contains no impossible models. I.e. the models of Ax_o do not contradict the Floyd-Hoare proof rules for programs.

Part $\big((i) \Longleftrightarrow (ii)\big)$ of Theorem 9 is a kind of semantic characterization of the information implicitly contained in the Floyd-method. It appears that this information content of Floyd-method is IA^q. Theorem 9 also says that if we can reason about time as being ordered, i.e. use $\mathrm{OA} \subseteq F_t$, then our reasoning ability is not beyond the power of Floyd's method. But if we can perform addition on time (note that $\mathrm{Pres} \models \mathrm{OA}$) or if we can quantify over time points (IA) then our reasoning ability is definitely beyond the power of Floyd's method. Note that quantifying over time is roughly the same as using time-modalities.

It is interesting to compare the powers of different proof methods. The following Theorem 10 says that the proof method $(\vdash^{N}, \mathrm{Prn})$ is strictly stronger than the Floyd-Hoare method $(\vdash^{F}, \mathrm{Prf})$.

Corollary 9/a

There are a finite similarity type d and $\varphi \in \mathrm{HF}_d$ such that

$$\mathrm{IA} \vdash^{N} \varphi \qquad \text{but} \qquad \not\vdash^{F} \varphi \quad .$$

Proof :

By the above Theorem 9 there are a finite similarity type d and a finite $\mathrm{Th} \subseteq F_d$ and $\big(\varphi \rightarrow \Box(p,\psi)\big) \in \mathrm{HF}_d$ such that

$$\mathrm{Th} \cup \mathrm{IA} \vdash^{N} \big(\varphi \rightarrow \Box(p,\psi)\big) \qquad \text{but} \qquad \mathrm{Th} \not\vdash^{F} \big(\varphi \rightarrow \Box(p,\psi)\big) \quad .$$

Let $\{ x_0, \ldots, x_n \}$ contain all the variables occurring in Th . Let φ' be the formula $\varphi \wedge \forall x_0 \ldots \forall x_n (\wedge \mathrm{Th})$. Since $|\mathrm{Th}| < \omega$ we have $\varphi' \in F_d$. Let φ be the formula $\big(\varphi' \rightarrow \Box(p,\psi)\big)$.

QED(Corollary 9/a)

Definition 18(the standard dynamic language $\langle DF_d , STM_d , \overset{\omega}{\models} \rangle$)

Let $\mathfrak{M} = \langle \underset{\sim}{T} , \underset{\sim}{D} , I , ext \rangle \in M_{td}$.

$\mathfrak{M}$ is said to be standard iff conditions (i) - (iii) below hold.

(i) $\underset{\sim}{T} = \langle \omega, \leq, +, \cdot, 0, 1 \rangle$.

(ii) $I = {}^{T}D$.

(iii) $(\forall s \in {}^{T}D)(\forall b \in T)\ ext(s,b) = s(b)$.

The class of all standard elements of M_{td} is denoted by STM_d .

Let $Th \subseteq DF_d$ and $\varphi \in DF_d$. Then we define

$Th \overset{\omega}{\models} \varphi \iff (\forall \mathfrak{M} \in Mod(Th) \cap STM_d)\ \mathfrak{M} \models \varphi$.

End of Definition 18.

Note that $STM_d \models Ax$ and $\mathfrak{N} \in STM_{d'}$, where $\mathfrak{N}$ and d' were defined in Definition 6.

Theorem 10 (B. Biró - L. Csirmaz)

There are a similarity type d and a finite theory $Th \subseteq F_d$ such that for some Floyd-Hoare statement $(\varphi \to \Box(p,\psi)) \in HF_d$ conditions (i) - (iii) below hold.

(i) $Th \cup Ax_o \overset{N}{\vdash} (\varphi \to \Box(p,\psi))$.

(ii) $Th \overset{F}{\not\vdash} (\varphi \to \Box(p,\psi))$.

(iii) $Th \overset{\omega}{\models} (\varphi \to \Box(p,\psi))$.

Proof of Theorem 10 :

In the proof of part $\big((v) \not\Longrightarrow (i)\big)$ of Theorem 9 a finite $Th \subseteq F_d$ and $\varrho \in HF_d$ were constructed such that $Th \overset{F}{\not\vdash} \varrho$ but $Th \cup IA \overset{N}{\vdash} \varrho$.
By $IA \subseteq Ax_o$ we proved (i) and (ii) . By $STM_d \models Ax$ obviously (i) always implies (iii) but validity of (iii) can be checked directly by looking at our concrete Th and ϱ .
Note that we do not need the full power of the proof of Theorem 9 here since the ultraproduct construction proving (ii) is clear, and to prove (i) we have the full power of $Ax_o = IA^{+} \cup OA$ at our disposal.
QED(Theorem 10)

Problem

Do there exist d , $Th \subseteq F_d$ and $\varrho \in HF_d$ such that

$Ax \cup Th \models \varrho$ and

$Ax_o \cup Th \not\models \varrho$?

Definition 19(the sets PA' , PA_d of axioms about data)

Recall from Definition 6 that the similarity type d' consists of the binary relation symbol $\leq'$ and the operation symbols $+'$, $\cdot'$, $0'$, $1'$ with arities 2,2,0,0 respectively. Note that d' is disjoint from t , actually d' is a disjoint copy of t .

PA' denotes the set of Peano axioms formulated in $F_{d'}$.

Note that $\mathfrak{N} \models PA' \cup PA$ where $\mathfrak{N}$ was introduced in Definition 6.
Also note that $PA \subseteq F_t^Z$ while $PA' \subseteq F_{d'}$.

Let d be an arbitrary similarity type containing d' .
Then $F_d \supseteq F_{d'}$, but possibly $F_d \neq F_{d'}$.
We define PA_d as

$$PA_d \overset{df}{=} PA' \cup \{ ([\varphi(0') \wedge \forall x(\varphi(x) \to \varphi(x +' 1'))] \to \forall x \varphi(x)) : \varphi(x) \in F_d \} .$$

Clearly $d = d'$ iff $PA_d = PA'$.

End of Definition 19

Theorem 11 (Andréka-Csirmaz-Németi-Paris)

Let the similarity type d contain d' . Let $Th \subseteq F_d$ and $\varphi \in HF_d$. Assume $PA' \subseteq Th$. Then (i)-(iii) below hold.

(i) $Th \vdash^F \varphi \iff Th \cup Ax_o \models \varphi$.

(ii) $Th \vdash^F \varphi \iff Th \cup Ax \models \varphi$.

(iii) $PA' \vdash^F \varphi \iff PA' \cup Ax \models \varphi$.

Proof of Theorem 11 :

Proofs of (iii) can be found in [3], [10], [37/a].

The proof of (i) can be found in [1] as Thm.6 there.

(ii) was proved in [1] as Thm.6 there under the additional assumption that $Th \supseteq PA_d$. The condition $Th \supseteq PA_d$ was eliminated from the proof of (ii) by Jeff B. Paris (Manchester) and L. Csirmaz recently.

QED(Theorem 11)

About (ii) of Theorem 11 above we would like to emphasize that if $PA' \subseteq Th$ then d may contain symbols (for) which the induction axioms are not postulated, moreover, it is allowed that for some $\varphi(x) \in F_d$ we have $\left(\varphi(0) \wedge \forall x \left[\varphi(x) \longrightarrow \varphi(x+'1')\right] \wedge \exists x \neg \varphi(x)\right) \in Th$.

Of course in this case $\varphi(x) \in F_d$ but $\varphi(x) \notin F_{d'}$. To be able to appreciate the difference between the conditions $PA_d \subseteq Th$ and $PA' \subseteq Th$ see the concrete example constructed in the proof of Proposition 13.

Note that by Theorem 10 the condition $PA' \subseteq Th$ is necessary in Theorem 11(i) and (ii) .

Recall $\models^{\omega}$ from Definition 18.

Theorem 11/a

Let d , d' and PA' be as in Theorem 11 and Definition 19. Let $Th \subseteq F_d$ be recursively enumerable. Assume ZFC $\models$ " Th is consistent" and that $Th \supseteq PA'$.

Then there is $\varphi \in HF_d$ such that

ZFC $\vdash$ " $Th \models^{\omega} \varphi$ " but

$Th \not\vdash^F \varphi$.

The proof is available from the author, the same applies to Thm.11/b.

Let $Th_o \subseteq F_{d_o} \subseteq F_{d_1}$. We call Th_o nontrivial if Th_o is good in the sense of Definition 1 in [5].

Let $Th_o \subseteq Th_1 \subseteq F_{d_1}$. Th_1 is said to be a semantically conservative extension of Th_o iff every (one-sorted) model of Th_o is a reduct of a model of Th_1 .

Theorem 11/b

Let $Th_o \subseteq F_{d_o}$ be nontrivial and recursively enumerable. Then there are $\varphi \in HF_{d_o}$ and a recursively enumerable semantically conservative extension Th_1 of Th_o such that $Th_o \not\vdash^F \varphi$ and $Th_1 \vdash^F \varphi$.

If Th_o is decidable or finite then so is Th_1 .

QED(Thm.11/b).

Proposition 11/c

There is a decidable set $Na \subseteq F_t$ of axioms such that $\underset{\sim}{N} \models Na$ and for some $\varphi \in HF_d$, we have

$Na \cup IA^+ \cup PA' \overset{N}{\vdash} \varphi$ and

$PA' \overset{F}{\nvdash} \varphi$.

Let $Nax \subseteq F_t$ be defined as

$Nax \overset{df}{=} \{ \varphi \in F_t : ZFC \vdash " \underset{\sim}{N} \models \varphi " \}$.

Let $Axz \overset{df}{=} Nax \cup IA^+$.

Then $Ax \subseteq Axz$ and Axz is recursively enumerable.

We can use Axz as our axioms about time; still for any recursively enumerable $Th \subseteq F_d$ we have that the set of $Axz \cup Th \overset{N}{\vdash}$ -proofs is decidable.

Proposition 11/d

There is $\varphi \in HF_d$, such that (i) - (iii) below hold.:

(i) $Axz \cup PA' \overset{N}{\vdash} \varphi$.

(ii) $PA' \overset{F}{\nvdash} \varphi$.

(iii) $PA' \overset{\omega}{\models} \varphi$.

Before leaving this section, we briefly return to the theme of

comparing powers of different proof methods

as it was done in Theorems 9, 10 and Corollary 9/a.

IA^1 will be defined to be the set of induction axioms containing at most one time-variable.

Definition 19/a

Recall the definition of $Lax \subseteq F_d \subseteq F_{td}$ from Definition 15 (page 46).

$IA^1 \overset{df}{=} \{ \varphi^+_{z_o} : \varphi \in F_{td}$ and $(\forall i>0)($ z_i does not occur in φ neither free nor bound $) \} \cup Lax$.

Theorem 11/e

Let $Th \subseteq F_d$ and $\varphi \in HF_d$. Consider statements (i),(ii) below.

(i) $Th \overset{F}{\vdash} \varphi$.

(ii) $Th \cup IA^1 \models \varphi$.

$Th \cup IA^1 \overset{N}{\vdash} \varphi$

$\Downarrow \Uparrow$

$Th \overset{F}{\vdash} \varphi$

Then (i) $\not\Longleftrightarrow$ (ii) .

Moreover, there is a finite d and $\varphi \in HF_d$ such that

$IA^1 \overset{N}{\vdash} \varphi$ but $\overset{F}{\nvdash} \varphi$.

Proof of Theorem 11/e :

1) (i) $\Longrightarrow$ (ii) :

By carefully inspecting part ((i) $\Longrightarrow$ (ii)) of the proof of Theorem 9, one can see that Statement 11/e.1 below was proved there.

Statement 11/e.1 : $Th \overset{F}{\vdash} \varphi \Longrightarrow Th \cup (IA^1 \cap IA^q) \models \varphi$.

To see this, observe that the formula γ constructed there is such that $(\forall i>0)[z_i$ does not occur in $\gamma]$. Hence $\gamma^+_{z_o} \in IA^1$.

The only induction axiom used in part ((i) $\Longrightarrow$ (ii)) of the proof of Theorem 9 was $\gamma^+_{z_o}$.

QED((i) $\Longrightarrow$ (ii))

2) Outline of proof of (ii) $\not\Longrightarrow$ (i) :

The proof goes by modifying part ((v) $\not\Longrightarrow$ (i)) of Proof of Thm.9. We shall briefly refer to part ((v) $\not\Longrightarrow$ (i)) of Proof of Thm.9 as Proof 9.

Let d, $Th \subseteq F_d$ and p_1 be as in Proof 9. Then
$Th \overset{F}{\not\vdash} \Box(p_1, x_3=x_4)$ was proved in Claim 9.1. The proof of
$Th \cup IA^1 \models \Box(p_1, x_3=x_4)$ goes by modifying the proof of Claim 9.4.

Let $\mathfrak{M} \models Th \cup IA^1$ and let $\bar{s}$ be a trace of p_1 in $\mathfrak{M}$.
Recall that $s_i(z)$ denotes $ext(s_i, z)$.
Let $\bar{x} \overset{df}{=} \langle x_1, \ldots, x_5 \rangle$, and $\bar{s}(z) \overset{df}{=} \langle s_1(z), \ldots, s_5(z) \rangle$. Then
$\exists \bar{x} \varphi$ abbreviates $\exists x_1 \ldots \exists x_5 \varphi$.
The modified version of Lemma 9.4.1 is:

Lemma 11.4.1 : $\mathfrak{M} \models \forall z_0 \exists \bar{x} (\bar{x}=\bar{s}(z_0) \wedge (\bar{x}=\bar{s}(0) \vee \exists z_0 [\bar{x}=\bar{s}(z_0+1)]))$.
The proof goes by practically the same induction as that of the original Lemma 9.4.1.

The modifications of Lemmas 9.4.2, 9.4.3 are similar.

The modified version of Claim 9.4.4 is:

Claim 11.4.4 :
$\mathfrak{M} \models \forall z_0 \exists \bar{x} (\bar{x}=\bar{s}(z_0) \wedge \forall z_0 [(x_5=s_5(z_0) \wedge x_1=s_1(z_0)) \rightarrow \bar{x}=\bar{s}(z_0)])$.
The proof goes by systematically modifying the original proof of Claim 9.4.4 in the above spirit. We omit this, the only hint we give is the following:
Let $\varphi(z_0)$ be the formula
$\exists \bar{x} (\bar{x}=\bar{s}(z_0) \wedge \forall z_0 [(x_5=s_5(z_0) \wedge x_1=s_1(z_0)) \rightarrow \bar{x}=\bar{s}(z_0)])$.
Then Claim 11.4.4 above states $\mathfrak{M} \models \forall z_0 \varphi(z_0)$. Clearly $\varphi(z_0)^+_{z_0} \in IA^1$ hence $\mathfrak{M} \models \varphi(z_0)^+_{z_0}$. Thus we can prove $\forall z_0 \varphi(z_0)$ by induction on z_0.

Let us consider e.g. Claim 9.4.5.5. Let $\psi(z_0)$ be the formula
$[s_5(z_0)=1' \rightarrow \exists \bar{x} (\bar{x}=\bar{s}(z_0) \wedge \exists z_0 [s_5(z_0)=2' \wedge x_1=s_1(z_0) \wedge x_3=s_4(z_0)])$.
The modified version of Claim 9.4.5.5 is

Claim 11.4.5.5 : $\mathfrak{M} \models \forall z_0 \psi(z_0)$.
Clearly $\psi(z_0)^+_{z_0} \in IA^1$ and hence we can prove $\mathfrak{M} \models \forall z_0 \psi(z_0)$ by induction on z_0.

The rest of the proof goes by systematically modifying the proof of Claim 9.4 in the above manner.

Equivalence of $\models$ and $\overset{N}{\vdash}$ is stated in our completeness theorem(Theorem 2). Elimination of Th i.e. proof of
$IA \overset{N}{\vdash} \varphi$ but $\overset{F}{\not\vdash} \varphi$
goes exactly as in Corollary 9/a.

QED(Theorem 11/e)

Definition 19/b (Modal dynamic logic i.e. dynamic logic based on time-modalities only)

1) Syntax DF_d^{mod} :

Let d be any similarity type.

T_d^{mod} is defined to be the smallest set such that
$(\forall n \in \omega)(\forall \tau_1, \ldots, \tau_n \in T_d^{mod})$
$[\{ x_n, ext(y_n) \} \subseteq T_d^{mod}$ and $f(\tau_1, \ldots, \tau_n) \in T_d^{mod}$ for every function symbol f of d such that $d(f)=n]$.

DF_d^{mod} is defined to be the smallest set satisfying (i)-(iii) below.

(i) $\{ (\tau=\sigma), (y_i=y_j) \} \subseteq DF_d^{mod}$ for all $i,j \in \omega$ and $\tau, \sigma \in T_d^{mod}$.

(ii) $R(\tau_1, \ldots, \tau_n) \in DF_d^{mod}$ for every relation symbol R of d such that $n=d(R)$, and $\tau_1, \ldots, \tau_n \in T_d^{mod}$.

(iii) $\{\neg\varphi, (\varphi \wedge \psi), \exists x_i \varphi, \exists y_i \varphi, \text{First}\,\varphi, \text{Next}\,\varphi, \text{Alw}\,\varphi, \Box(p,\varphi)\} \subseteq DF_d^{mod}$
for all $i \in \omega$ and for all $\varphi, \psi \in DF_d^{mod}$ and all $p \in P_d$.

2) Translation function $\text{mod} : DF_d^{mod} \longrightarrow DF_d$.:

The definition goes by recursion on the structure of DF_d^{mod} .

$(\forall i \in \omega)\left[\text{mod}\, x_i \stackrel{df}{=} x_i\ ,\ \text{mod}\, y_i \stackrel{df}{=} y_i\ ,\ \text{mod}(\text{ext}(y_i)) \stackrel{df}{=} (\text{ext}(y_i, z_o))\right]$.

Let $d(f)=n$. Then

$\text{mod}(f(\tau_1, \ldots, \tau_n)) \stackrel{df}{=} f(\text{mod}\,\tau_1, \ldots, \text{mod}\,\tau_n)$ and

$\text{mod}(\tau_1 = \tau_2) \stackrel{df}{=} (\text{mod}\,\tau_1 = \text{mod}\,\tau_2)$

for all $\tau_1, \tau_2, \ldots, \tau_n \in T_d^{mod}$.

$\text{mod}(y_i = y_j) \stackrel{df}{=} (y_i = y_j)$ for all $i,j \in \omega$.

Let $\varphi, \psi \in DF_d^{mod}$. Let $p \in P_d$. Let $i \in \omega$. Then

$\text{mod}(\neg\varphi) \stackrel{df}{=} \neg\,\text{mod}\,\varphi$,

$\text{mod}(\varphi \wedge \psi) \stackrel{df}{=} (\text{mod}\,\varphi \wedge \text{mod}\,\psi)$,

$\text{mod}(\exists x_i \varphi) \stackrel{df}{=} \exists x_i\,(\text{mod}\,\varphi)$,

$\text{mod}(\exists y_i \varphi) \stackrel{df}{=} \exists y_i\,(\text{mod}\,\varphi)$,

$\text{mod}(\text{Alw}\,\varphi) \stackrel{df}{=} \forall z_o(\text{mod}\,\varphi)$,

$\text{mod}(\text{First}\,\varphi) \stackrel{df}{=} \exists z_o\,(z_o = 0 \wedge \text{mod}\,\varphi)$,

$\text{mod}(\text{Next}\,\varphi) \stackrel{df}{=} (\text{mod}\,\varphi)(z_o/z_o+1)$

where for any $\psi \in DF_d$ we define $\psi(z_o/z_o+1)$ to be the formula obtained from ψ by replacing every free occurrence of z_o by (z_o+1) in ψ . Note that $\psi(z_o/z_o+1)$ is equivalent with the formula $\exists z_i\left[z_i = z_o \wedge \exists z_o\,(z_o = z_i+1 \wedge \psi)\right]$ if z_i does not occur freely in ψ .

$\text{mod}(\Box(p,\varphi)) \stackrel{df}{=} \Box(p, \text{mod}\,\varphi)$.

By the above the function

$$\text{mod} : DF_d^{mod} \longrightarrow DF_d$$

is fully defined.

3) Validity relation $\models^{mod} \subseteq M_{td} \times DF_d^{mod}$.

Let $\varphi \in DF_d^{mod}$ and $\mathfrak{M} \in M_{td}$. Then we define

$$\mathfrak{M} \models^{mod} \varphi \iff^{df} \mathfrak{M} \models \text{mod}\,\varphi \quad .$$

4) The language DL_d^{mod} of modal dynamic logic (i.e. of dynamic logic based on time-modalities only)

$$DL_d^{mod} \stackrel{df}{=} \langle DF_d^{mod}, M_{td}, \models^{mod} \rangle .$$

5) Axioms IA^{mod} of modal dynamic logic :

$$IA^{mod} \stackrel{df}{=} \left\{\left(\left[\text{First}\,\varphi \wedge \text{Alw}(\varphi \to \text{Next}\,\varphi)\right] \to \text{Alw}\,\varphi\right) : \varphi \in DF_d^{mod}\right\} \cup \text{Lax} .$$

End of Definition 19/b.

Notation : $IA^{mo} \stackrel{df}{=} \{\text{mod}\,\varphi : \varphi \in IA^{mod}\}$.
Note that $IA^{mo} \subseteq IA^1$ and that $HF_d \subseteq DF_d^{mod}$.

Proposition (Completeness of DL_d^{mod})
Let $\varphi \in DF_d^{mod}$. Then

(i) $IA^{mod} \models^{mod} \varphi \iff IA^{mo} \vdash^{N} \text{mod}\,\varphi$.

(ii) $\text{Th} \models^{mod} \varphi \iff \{\text{mod}\,\psi : \psi \in \text{Th}\} \vdash^{N} \text{mod}\,\varphi$,
for all $\text{Th} \subseteq DF_d^{mod}$.

The proof is immediate by the definitions and by the completeness theorem of DL_d i.e. by Theorem 2.

Theorem 11/f

Let $Th \subseteq F_d$ and $\varphi \in HF_d$. Consider statements (i),(ii) below.

(i) $Th \vdash^{F} \varphi$.

(ii) $Th \cup IA^{mod} \models^{mod} \varphi$.

$Th \cup IA^{mod} \models^{mod} \varphi$ $\Downarrow \Uparrow$ $Th \vdash^{F} \varphi$

Then (i) $\Longrightarrow$ (ii) and (i) $\not\Longleftarrow$ (ii) .

In more detail, statements (1) and (2) below hold.

(1) For all d , $Th \subseteq F_d$ and $\varphi \in HF_d$ such that $Th \vdash^{F} \varphi$ we have $Th \cup IA^{mod} \models^{mod} \varphi$ that is $Th \cup IA^{mo} \vdash^{N} \varphi$.

(2) There is a finite similarity type d and $\varphi \in HF_d$ such that $IA^{mod} \models^{mod} \varphi$ but $\not\vdash^{F} \varphi$ (and then obviously $IA^{mo} \vdash^{N} \varphi$).

Proof of Theorem 11/f :

Proof of (i) $\Longrightarrow$ (ii) :

Part ((i) $\Longrightarrow$ (ii)) of Proof of Theorem 9 proves the present implication too, because the formula $\gamma^{+}_{z_o}$ constructed there is in IA^{mo} , that is for some $\varphi \in IA^{mod}$ we have $\gamma^{+}_{z_o} = \mathrm{mod}\, \varphi$.

Outline of proof of (ii) $\not\Longrightarrow$ (i) :

The proof goes by modifying part ((v) $\not\Longrightarrow$ (i)) of the proof of Theorem 9 exactly as we did it in the proof of Theorem 11/e above. Again, we let d , Th , p_1 to be the ones defined in the proof of Theorem 9. Then Claim 9.1 proves $Th \vdash^{F} \Box(p_1, x_3 = x_4)$. The proof of $Th \cup IA^{mod} \models^{mod} \Box(p_1, x_3 = x_4)$ goes by modifying the proof of Claim 9.4 such that the induction formulas will be members of IA^{mo} . Actually all the induction formulas in the proof of Theorem 11/e are in IA^{mo} , as it is illustrated below.

Let $\mathfrak{M} \models^{mod} IA^{mod}$ and let $\bar{s}$ be a trace of p_1 in $\mathfrak{M}$.

Let $\bar{x} = \langle x_1, \ldots, x_5 \rangle$ and $ext(\bar{s}) = \langle ext(s_1), \ldots, ext(s_5) \rangle$.

Notation: $Smt\, \varphi$ denotes the formula $\neg Alw \neg \varphi$.

Let $\varphi(\bar{s})$ be the formula

$\exists \bar{x} \, (\, \bar{x} = ext(\bar{s}) \wedge [First(\bar{x} = ext(\bar{s})) \vee Smt\, Next(\bar{x} = ext(\bar{s}))] \,)$.

Note that $\varphi \in DF_d^{mod}$ or more precisely $\varphi(\bar{s}/\bar{y}) \in DF_d^{mod}$.

Then the present version of Lemmas 9.4.1 and 11.4.1 is:

Lemma 11/f.4.1 : $\mathfrak{M} \models Alw\, \varphi(\bar{s})$.

This follows from $([First\, \varphi \wedge Alw(\varphi \rightarrow Next\, \varphi)] \rightarrow Alw\, \varphi) \in IA^{mod}$.

Let $\psi(\bar{s})$ be the formula

$\exists \bar{x} \, (\, \bar{x} = ext(\bar{s}) \wedge Alw \, [(\, x_5 = ext(s_5) \wedge x_1 = ext(s_1) \,) \rightarrow \bar{x} = ext(\bar{s})] \,)$.

Then if $\psi(y)$ is the formula obtained by writing $y_1, \ldots, y_5$ into the place of $s_1, \ldots, s_5$ in $\psi(\bar{s})$ then $\psi(\bar{y}) \in DF_d^{mod}$.

The present version of Claims 9.4.4 and 11.4.4 is:

Claim 11/f.4.4 : $\mathfrak{M} \models Alw\, \psi(\bar{s})$.

Since $([First\, \psi \wedge Alw(\psi \rightarrow Next\, \psi)] \rightarrow Alw\, \psi) \in IA^{mod}$, the proof proceeds as in Claim 11.4.4 by rewritting the original proof of Claim 9.4.4.

QED(Theorem 11/f)

Note that the above proof heavily uses the possibility of nesting modalities in DF_d^{mod} . Namely we used formulas of the kind $Alw\, \exists x \, (\varphi \wedge Smt\, \psi)$ etc.. We believe that this is the feature of modal logic that makes $(IA^{mod} \models^{mod})$ stronger than Floyd's logic $\vdash^{F}$.

Below we prove that IA^{mod} contains no information about time structure $\underset{\sim}{T}$.

Proposition 11/g

Let d be an arbitrary symilarity type. Let $\underset{\sim}{A} \in M_t$ and $\underset{\sim}{B} \in M_d$, $\underset{\sim}{B} \models Lax$.

Then there is $\mathfrak{M} \in M_{td}$ such that $\mathfrak{M} \models IA^{mod}$ and the time structure and data structure of $\mathfrak{M}$ are $\underset{\sim}{A}$ and $\underset{\sim}{B}$ respectively, that is $\mathfrak{M} = \langle \underset{\sim}{A}, \underset{\sim}{B}, I, ext \rangle$ for some I and ext.

Proof of Proposition 11/g :

Assume the hypothesis. $I \overset{df}{=} B$ and $(\forall b \in B)(\forall a \in A)\ ext(b,a) \overset{df}{=} b$.

Then $\langle \underset{\sim}{A}, \underset{\sim}{B}, I, ext \rangle \models IA^{mo}$ is easy to prove.

QED(Proposition 11/g)

THE LATTICE OF PROOF METHODS FOR HF_d :

We define a pre-ordering $\geq$ on the proof methods as follows. :

$(X \vdash^{Y}) \leq (Y \vdash^{Z})$ is defined to hold iff

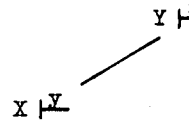
$\left[(Th \cup X \vdash^{Y} \varphi) \Longrightarrow (Th \cup Y \vdash^{Z} \varphi)\right]$ for every similarity type d , $Th \subseteq F_d$ and $\varphi \in HF_d$.

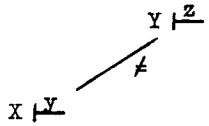
E.g. $(IA^q \vdash^{N}) \leq (Ax \vdash^{N})$ since $IA^q \subseteq Ax$, and

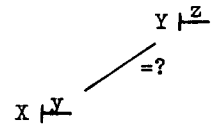
$(\vdash^{F}) \leq (IA^{+} \vdash^{N})$ by Theorem 9.

The relation $\leq$ induces an equivalence relation $\equiv$ defined as:

$(X \vdash^{Y}) \equiv (Y \vdash^{Z})$ iff $\left[(X \vdash^{Y}) \leq (Y \vdash^{Z})\right.$ and $\left.(X \vdash^{Y}) \geq (Y \vdash^{Z})\right]$.

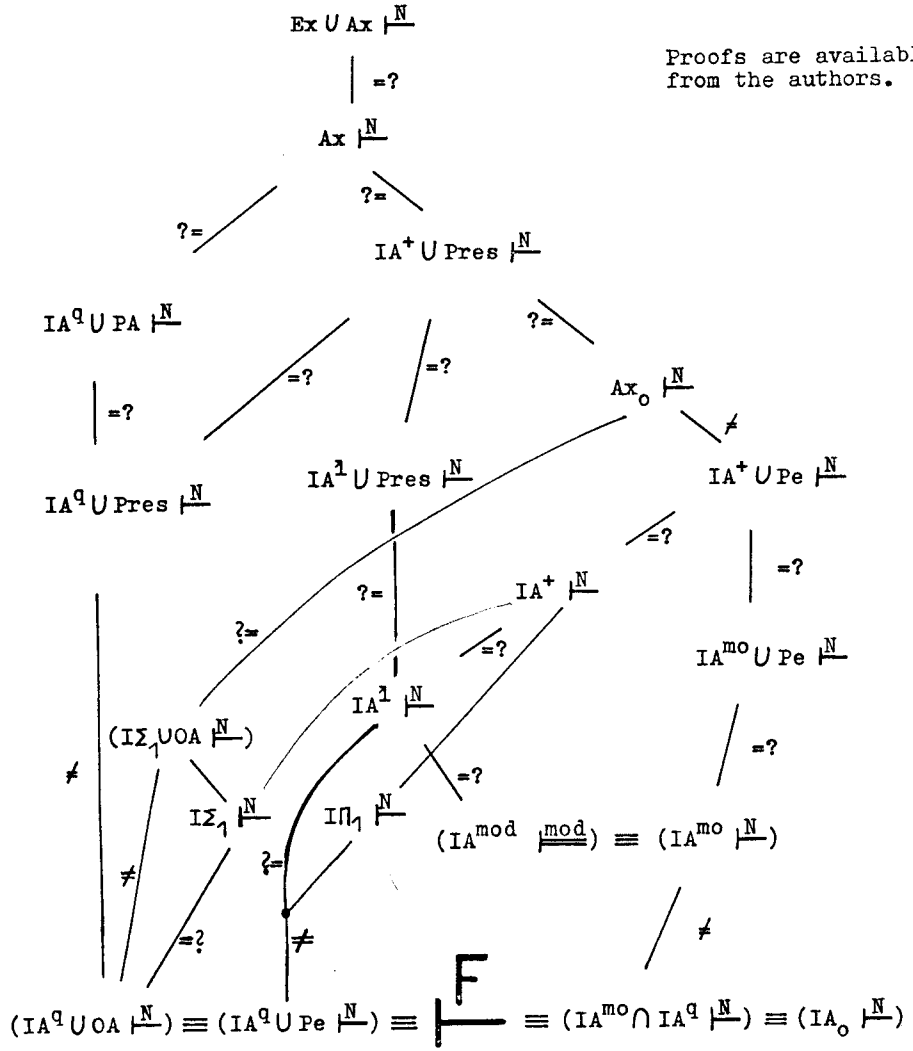
A straight line 

on the figure below indicates the relation $(X \vdash^{Y}) \leq (Y \vdash^{Z})$. A line with $\neq$ added like 

indicates the strict relation $<$ that is $\left[(X \vdash^{Y}) \not\geq (Y \vdash^{Z})\right.$ and $\left.(X \vdash^{Y}) \leq (Y \vdash^{Z})\right]$. A line with =? added like 

indicates that $(X \vdash^{Y}) \leq (Y \vdash^{Z})$ but we do not know whether $(X \vdash^{Y}) \geq (Y \vdash^{Z})$ holds or not. If two nodes are not connected then we do not know whether they are related in any direction or not that is we do not know whether they are comparable. For example we do not know whether $(IA^q \cup Pres \vdash^{N}) \leq (IA^{+} \cup Ax_o \vdash^{N})$ holds or not. Note that the fact $IA^q \not\models IA^q \cup OA$ does not imply $(IA^q \vdash^{N}) \neq (IA^q \cup OA \vdash^{N})$ since proof methods here are compared only w.r.t. $Th \subseteq F_d$ and $\varphi \in HF_d$.

The set IA_o will be defined in Definition 20 in the next section. Note that $IA_o \subseteq IA^{mo} \cap IA^q$.



Proofs are available from the authors.

We know $(IA^q \cup Pres \vdash^N) \not\equiv (IA^+ \cup Pe \vdash^N)$.

Recall that $Ax_o = IA^+ \cup OA \subseteq Ax = IA^+ \cup PA$ and

$Pe \subseteq OA \subseteq Pres \subseteq PA \subseteq F_t$ and

$IA_o \subseteq IA^q \subseteq IA^+ \supseteq IA^1 \supseteq IA^{mo} \supseteq IA_o$.

$I\Sigma_1 \stackrel{d}{=} \{(\exists z_1 \dots z_m \varphi)^+_{z_o} : m \in \omega$ and $\varphi \in F_{td}$ contains no quantifier of sort t$\}$.

$I\Pi_1 \stackrel{d}{=} \{(\forall z_1 \dots z_m \varphi)^+_{z_o} : m \in \omega$ and $\varphi \in F_{td}$ contains no quantifier of sort t$\}$.

§6. CONNECTIONS WITH SOME OTHER BRANCHES OF EXPLICIT TIME SEMANTICS OF PROGRAMMING

We use the names "Explicit Time Semantics" , "Nonstandard Semantics", "Nonstandard time semantics" as synonyms. For works in this field see [1] , [3] , [5] , [6] , [10] , [16-21] , [28-30] .

In Definition 20 below we recall the Continuous Traces Language CL_d of type d from [3] , [4] , [16 - 21] .

Definition 20 (the Continuous Traces Language $CL_d = \langle HF_d , M_d , \models^c \rangle$)

Let $\varphi \in F_d$.

Let $m \in \omega$ and let $\bar{y} = \langle y_o, \dots, y_m \rangle$ and $\bar{x} = \langle x_o, \dots, x_m \rangle$.

Let $\bar{x} = ext(\bar{y}, z_i)$ denote the formula $\bigwedge_{i \leq m} x_i = ext(y_i, z_i)$.

We define the formula $\varphi_m \in F_{td}$ to be $\forall \bar{x}(\bar{x} = ext(\bar{y}, z_i) \to \varphi)$.

Remark: $\varphi_m = \varphi_m(z_i, \bar{y}) \in F_{td}$ and φ_m is equivalent with

$\varphi(ext(y_o, z_i), \dots, ext(y_m, z_i), x_{m+1}, \dots, x_w)$ if $\varphi = \varphi(x_o, \dots, x_w)$.

End of Remark

We define

$IA_o \stackrel{df}{=} \{ (\varphi_m)^+_{z_o} : \varphi \in F_d , m \in \omega \} \cup \{ (i \neq j) : i, j \in Lab$ and $i \neq j \}$

Recall the set $Pe \subseteq F_{td}$ of 'successor axioms' from Definition 15.

The Continuous Traces Axioms are $Ctax \stackrel{df}{=} IA_o \cup Pe$.

Let $\underset{\sim}{E} \in M_d$ and $\varphi \in DF_d$. Then we define

$\underset{\sim}{E} \models^c \varphi \iff (\forall \mathcal{M} \in Mod(Ctax)) \left[\underset{\sim}{D} = \underset{\sim}{E} \to \mathcal{M} \models \varphi \right]$.

The Continuous Traces Language is

$CL_d \stackrel{df}{=} \langle HF_d , M_d , \models^c \rangle$.

End of Definition 20

Let $Th \subseteq HF_d$ and $\varphi \in HF_d$.

We shall use $Th \models^c \varphi$ in the usual sense, i.e.

$Th \models^c \varphi$ iff $(\forall \underset{\sim}{E} \in M_d)\left[\underset{\sim}{E} \models^c Th \Rightarrow \underset{\sim}{E} \models^c \varphi \right]$.

It is easy to check that $Th \models^c \varphi$ iff $Th \cup Ctax \models \varphi$.

The Continuous Traces Semantics (or Language) $CL_d = \langle HF_d , M_d , \models^c \rangle$ was introduced in [4] and further refined and investigated in [3], [16-21]. In [3], [4] $\models^c$ was denoted by $\models^{pc}$.

Recall IA^q from Definition 15.

Proposition 12(semantic characterization of Continuous Traces Semantics)

Let $Th \subseteq F_d$ and $\varsigma \in HF_d$.

Then statements (i) - (iii) below are equivalent.

(i) $Th \models^c \varsigma$.

(ii) $Th \cup IA^q \models \varsigma$.

(iii) $Th \cup IA_o \models \varsigma$.

Proof of Proposition 12:

Assume (i) . Then by [16] $Th \vdash^F \varsigma$. Then by the proof of part $\big((i) \Rightarrow (ii)\big)$ of Thm.9 we have $Th \cup IA_o \models \varsigma$. To see this observe that in the proof of Theorem 9$\big((i) \Rightarrow (ii)\big)$ the only elements of IA we used were of the form $\varphi(ext(\bar{y}, z_i))^+_{z_o}$ for some $\varphi(\bar{x}) \in F_d$, φ containing no other free variables than $\bar{x}$. This proves (i) $\Rightarrow$ (iii) .

(iii) $\Rightarrow$ (ii) is obvious since $IA_o \subseteq IA^q$.

(iii) $\Rightarrow$ (i) holds by $IA_o \subseteq Ctax$.

Assume (ii). Then by Theorem 4 in [19] $Th \vdash^F \varsigma$. Then by the proof of Theorem 9((i) $\Rightarrow$ (ii)) we have $Th \cup IA_o \models \varsigma$. I.e. (iii) holds.

QED(Proposition 12)

Remark:

Let $d \geq d'$, PA' and PA_d be as in Definition 19. Then for every $Th \subseteq F_d$ such that $PA_d \subseteq Th$ the Continuous Traces Semantics $\langle HF_d , Mod(Th) , \models^c \rangle$ is equivalent with the Definable Traces Semantics of Gergely-Úry [28] w.r.t. Th .

However, there are $d \geq d'$ and a decidable $Th \subseteq F_d$ such that $PA' \subseteq Th$ and $\langle HF_d , Mod(Th) , \models^c \rangle$ is not equivalent with the Definable Traces Semantics of [28], [30] w.r.t. Th.

Proposition 13

Let d' and PA' be as in Definition 19. Then there are $d \geq d'$ and a decidable theory $Th \subseteq F_d$ and $\varsigma \in HF_d$ such that $PA' \subseteq Th$ and $Th \models \varsigma$ and $Th \models^c \varsigma$ and $Th \vdash^F \varsigma$ but Th does not imply ς w.r.t. Definable Traces Semantics of [28], [30] .

Proof

Let d consist of d' together with a new unary relation symbol R and a new constant symbol c . Let $Th \overset{df}{=} PA' \cup \{R(0'), [R(x) \rightarrow R(x+'1')]\}$.

Let the program p be $\langle (0': x_o \leftarrow 0'), (1': \text{IF } x_o = c \text{ GOTO } 4'), (2': x_o \leftarrow x_o +'1'), (3': \text{IF TRUE GOTO } 1'), (4': \text{HALT}) \rangle$.

Then $Th \vdash^{F} \Box(p,R(x_0))$ and clearly $Th \not\models^{c} \Box(p,R(x_0))$.

Let $\underset{\sim}{D}$ be a model of Th such that $\underset{\sim}{D} \models \neg R(c)$. Such a $\underset{\sim}{D}$ exists since if $\langle D,+,\cdot,0,1\rangle$ is any nonstandard model of PA' and $R \subseteq D$ is the set of all standard numbers and $c \in D$ is any nonstandard number then $\underset{\sim}{D} \models Th$ and $\underset{\sim}{D} \models \neg R(c)$.
Then $\Box(p,R(x_0))$ is not true in $\underset{\sim}{D}$ w.r.t. definable traces since there is a definable trace of p in $\underset{\sim}{D}$ which terminates with output c (of course this definable trace is not continuous). At the same time $\underset{\sim}{D} \not\models^{c} \Box(p,R(x_0))$ and $\underset{\sim}{D} \models \Box(p,R(x_0))$ since no continuous trace of p terminates in $\underset{\sim}{D}$.
QED (Proposition 13)

In this context see §9 at the end of this paper.

§7. CONNECTIONS WITH RELATED APPROACHES IN RPOGRAMMING THEORY

Let y be a variable in a program scheme. Sometimes such a y is called an "identifier" instead of "variable". We use the word "variable". What we call an "intension" of y is called an "L-value" of y in "Scott-Strachey semantics" of programming languages (see [34] p.202), while an extension ext(y,timepoint) is called an "R-value" of y there . Intensions for y are often called "addresses or locations in a computer corresponding to the identifier y " and ext(y,timepoint) is often called the "content of the address" (mentioned above) at the point "timepoint" of time. See the "Temporal Notions" of the Milne-Strachey book [34] on programming semantics.

In "operational semantics", abstract machines with registers are often used. Then to a variable (or identifier) y which occurs in a program, a register of the machine is associated. During the "computation" or "execution of the program" the register associated to y remains the same, it does not change. However, the content of the register may change many times. The register associated to y is the intension of y while the content of this register at time z is the extension of y at time z and the present paper denotes it by ext(y,z) .

In other approaches to programming semantics, e.g. in VDL, the concepts "environment" and "state" (or "store" or "memory") do correspond to our intensions and extensions. See e.g. [34]p.203. Namely an environment maps the variables $\{y_w : w \in \omega\}$ to "locations" and a state maps the locations to datavalues i.e. to elements of D . Thus environments are like our traces, locations correspond to our intensions and a state corresponds to the function $ext(-,z): I \longrightarrow D$ where $z \in T$ is a parameter. I.e. states are like elements of T , they correlate extensions to the intensions.

In short: our intension - extension duality corresponds to the usual locations - values duality as described e.g. in p.202-203 of [34] . During the execution of a program, the intension associated to a variable y does not change. Analogously, the location associated

to y does not change. I.e. the command "y=y+1" is a meaningful statement about the location or intension associated to y but it is not so meaningful if we try to interpret it as a statement about the value or extension associated to y .

§8. CONNECTIONS WITH RELATED APPROACHES IN NONCLASSICAL MODEL THEORY, PHILOSOPHICAL LOGIC AND SEMANTICS OF NATURAL LANGUAGES

The above mentioned problem was raised and studied not only in "program semantics theory" but also in a broader theory of Semantics of Languages in general. A frequently used and well developed branch of the latter is "Intensional Model Theory" , see e.g. [36] , [25] . Intensional Logic and Model Theory was elaborated by R.Montague (a student of Tarski) and his followers, see [36], [25], Our way of using these notions in programming theory is explained in [33] , in §2 of [23]p.3-4, and in [22] with several nice drawings on p.33-34 .

In Intensional Model Theory frequently used examples are the sentences "The temperature rises.", "The price of sugar rises." etc., see [36] p.268, [26] . They correspond to " y rises" where y is a variable standing for intensions. At a valuation k of the variables into $\mathfrak{M} = \langle \mathfrak{T} , \mathfrak{D} , I , ext \rangle$ such that $k(y)=s \in I$, the statement " y rises" is true in $\mathfrak{M}$ iff $ext(s,n) < ext(s,n+1)$ holds in $\mathfrak{M}$. I.e. "y rises" is meaningful only if y denotes an "intension" i.e. a function from some set T of time points to some set D of possible values. This "motivating example" sentence: "y rises" of intensional model theory is quite similar to the "programming language sentence" y=y+1. The latter is true if $ext(s,n+1)=ext(s,n)+1$ holds in $\mathfrak{M}$ at the valuation $k(y)=s \in I$.
Another intensional logic example: "The president changes in every four years.". I.e. " y changes in every four years". This is true at valuation $k(y)=s \in I$ in $\mathfrak{M}$ iff $\mathfrak{M} \models (ext(s,n) \neq ext(s,n+4))$.
Compare with "y changes during the execution of a program ".

The model theoretic treatments of time in [24]p.188 Def.3.1 and [36] p.258 are similar to ours in the respect that a "generalized model or interpretation" in both cases contains a structure $\mathfrak{T} = \langle T, \leq, \ldots \rangle$ called time structure, see [36] p.37-38, 98-101, 258. In [36] p.258 an "interpretation" is a tuple $\langle A, I, J, \leq, F \rangle$ where $\langle J, \leq \rangle$ is the same as our $\mathfrak{T}$; A is our "D" and ${}^{J}A$ which is denoted there by S_{eAIJ} is our "I". ${}^{J}A$ is called there the "set of intensions" exactly as we do here. Our operation symbol "ext" is denoted there by " $\vee$ " .

In [24]p.188 our $\mathfrak{M}$ is "$\mathcal{A}$" and our $\mathfrak{T}$ is "$\langle T, < \rangle$" . However, there are no intensions there thus the analogy stops here. It is true that the meaning of a constant symbol "c" in $\mathcal{A}$ is an element of ${}^{T}A$ where A is $\cup\{ A_t : t \in T \}$ but there are no variables ranging over ${}^{T}A$ in [24], while in [36] the variables of type $\langle s,e \rangle$ are just doing that.

An essential difference between intensional models (M_{td} of this paper or in [36]) and "Kripke-style" models ([2] or [24]) is that in a Kripke model $\langle \mathfrak{T} , \langle \mathfrak{D}_t : t \in T \rangle \rangle$ the elements of ${}^{T}D$ do not form a separate universe or "sort" to speak about while in an intensional model $\mathfrak{M} = \langle \mathfrak{T} , \mathfrak{D} , {}^{T}D , ext \rangle$ they definitely do so. See p.3 of [2] .

For further considerations on the subject of paragraphs 8-9 see [49] .

§9. RECENT DEVELOPMENTS AND PROBLEMS

To motivate the problems below, first we formulate some results. The notations d° and $PA' \subseteq F_d$, were introduced in Definition 19. $\underset{\sim}{N}$, $\models^{\omega}$, OA, STM_d and IA^q were introduced in Definitions 6, 18, 14, 15.

Theorem 14 (Gordon D. Plotkin - Németi)

Let $Th \subseteq F_d$ be recursively enumerable. Assume $Th \supseteq PA'$ and that $ZFC \models$ " Th is consistent" . Then there is $\varphi \in HF_d$ such that

$$ZFC \vdash \text{" } Th \models^{\omega} \varphi \text{ "} \quad \text{and} \quad Th \not\vdash^{F} \varphi \ .$$

Theorem 15

There is a decidable set $Tax \subseteq F_t^Z$ of time axioms such that $\underset{\sim}{N} \models Tax$ and for some $\varphi \in HF_d$, we have

$$PA' \cup IA^+ \cup Tax \vdash^{N} \varphi \quad \text{and} \quad PA' \not\vdash^{F} \varphi \ .$$

We define

$$IA^1 \overset{df}{=} \{ \varphi^+_{z_o} : \varphi \in F_{td} \text{ and } (\forall i>0)(\ z_i \text{ does not occur in } \varphi \text{ neither free nor bound}) \} \cup Lax \ .$$

Theorem 16

There is a finite $Th \subseteq F_d$ and $\varphi \in HF_d$ such that

$$Th \cup IA^1 \cup Pe \vdash^{N} \varphi \quad \text{and} \quad Th \not\vdash^{F} \varphi \ .$$

Cf. Theorems 9, 10.

Theorem 19 above says that any fragment of DL_d in which time modalities (always, sometime) are still available has a greater reasoning power than that of Floyd-Hoare method.

Theorems 17, 18 below are not recent developments, they are included here to motivate Problem 1 below and to show that the dynamic logic $\langle DL_d , \vdash^{N} \rangle$ developed in the present paper is complete in the sense of [14] and Harel w.r.t. standard time models. About these notions see the introduction of Part I. Note that in the axiom systems Numb and Sth(K) below no dynamic formulas occur, they contain only classical formulas from F_{td}.

Theorem 17 (Arithmetical completeness in the sense of Harel)

Let $K \subseteq M_d$.

$$Sth(K) \overset{df}{=} \{ \varphi \in F_{td} : \langle \underset{\sim}{N}, \underset{\sim}{D}, {}^{\omega}D, \text{valueof} \rangle \models \varphi \text{ for all } \underset{\sim}{D} \in K \}.$$

I.e. $$Sth(K) = \{ \varphi \in F_{td} : (\forall \mathfrak{M} \in STM_d)[\ \underset{\sim}{D} \in K \Longrightarrow \mathfrak{M} \models \varphi\ ] \}.$$

Let $\varphi \in DF_d$. Then

$$Sth(K) \vdash^{N} \varphi \quad \text{iff} \quad K \models^{\omega} \varphi \ .$$

Corollary 18 (Completeness w.r.t. oracles in the sense of [14])

Let $Numb \overset{df}{=} \{ \varphi \in F_{td} : \mathfrak{N} \models \varphi \}$. Let $\varphi \in DF_d$, . Then

$$Numb \vdash^{N} \varphi \quad \text{iff} \quad \mathfrak{N} \models \varphi \ .$$

Problem 1

Does there exist $St \subseteq F_{td}$ such that for every $Th \subseteq F_d$ and $\varphi \in HF_d$ we have

$$Th \models^{\omega} \varphi \quad \Longleftrightarrow \quad St \cup Th \models \varphi \quad ?$$

Problem 2

Do there exist d , $Th \subseteq F_d$ and $\varphi \in HF_d$ such that

$$Ax \cup Th \models \varphi \quad \text{and} \quad Ax_o \cup Th \not\models \varphi \quad ?$$

Cf. Definitions 14, 18, Theorem 10, and Proposition 6.

Problem 3

What is the answer to Problem 2 above under the additional restriction that Th be recursively enumerable ?

Problem 4

Let $Pres \subseteq F_t$ be as in Theorem 9. By Theorem 9 there are a finite $Th \subseteq F_d$ and $\varphi \in HF_d$ such that

$$Th \cup IA^q \cup Pres \models \varphi \quad \text{but} \quad Th \cup IA^q \cup OA \not\models \varphi \ .$$

Note that roughly speaking $OA \subseteq Pres \subseteq PA$ and they are the theories of $\langle suc, \leq \rangle$, $\langle suc, \leq, + \rangle$ and $\langle suc, \leq, +, \cdot \rangle$ respectively.

Are there a decidable $Th \subseteq F_d$ and $\varphi \in HF_d$ for some d such that

$$Th \cup IA^q \cup PA \models \varphi \quad \text{but} \quad Th \cup IA^q \cup Pres \not\models \varphi \quad ?$$

In other words: Theorem 9 says that the ability of performing addition on time increases the reasoning power of dynamic logic. Does the ability of performing multiplication on time affect the reasoning power in any similar way ?

Problem 5

Let IA^q and IA^f be as in Definition 15 and Proposition 6. Let $Th \subseteq F_d$ and $\varphi \in HF_d$ be arbitrary. Is it true that

$$Th \vdash^F \varphi \quad \text{iff} \quad Th \cup (IA^q \cap IA^f) \cup PA \models \varphi \quad ?$$

Cf. Theorems 9, 10.

Problem 6

Let $Th \subseteq F_d$ and $\varphi \in HF_d$. Assume $Th \supseteq PA'$ and $Th \cup Ax \cup Ex \models \varphi$. Is then $Th \vdash^F \varphi$ true ?

Problem 7

Let d and DIA be as in Theorem 7. Let $\varphi \in HF_d$. Assume $F_d \supseteq Th \supseteq PA'$ and $Th \cup Ax \cup Ex \cup DIA \models \varphi$. Is then $Th \vdash^F \varphi$ true ?

Problem 8

Continue the investigation started in Definition 16 and Proposition 7 !

Problem 9

See the figure above section 6 !

R E F E R E N C E S

1 Andréka,H. Csirmaz,L. Németi,I. Sain,I.: More complete logics for reasoning about programs. Preprint, Math.Inst.Hung.Acad.Sci.(1980).

2 Andréka,H. Dahn,B.I. Németi,I.: On a proof of Shelah. Bull.Acad. Polon.Sci.(ser. Math.) 26 (1976), pp.1-7.

3 Andréka,H. Németi,I. Sain,I.: A characterization of Floyd provable programs. In: Mathematical Foundation of Computer Science'81 (Proc. Strbské Pleso, Czechoslovakia) Lecture Notes in Computer Science 118, Springer, Berlin, 1981. pp.162-171.

4 Andréka,H. Németi,I.: Completeness of Floyd logic. Bulletin of Section of Logic, Wroclaw, Vol 7, No 3 (1978) pp.115-120. This is an abstract of "Completeness of Floyd method w.r.t. nonstandard time models", Seminar Notes, Math.Inst.Hung.Acad.Sci. - SZKI 1977 (in Hungarian).

5 Andréka,H. Németi,I. Sain,I.: Completeness problems in verification of programs and program schemes. In: Mathematical Foundations of Computer Science'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'79 (Proc. Berlin 1979), Akademie Verlag, Berlin , 1979. pp.18-24.

7 Andréka,H. Németi,I. Sain,I.: Program verification within and without logic. Bulletin of Section of Logic, Wroclaw, Vol 8, No 3, (1979) pp.124-130.

8 Berman,F.: A completeness technique for D-axiomatizable semantics. In: Proc. 11th Ann.ACM Symp. on Theory of Computing (Atlanta, Georgia, May 1979) pp.160-166.

9 Berman,F.: Syntactic and semantic structure in Propositional Dynamic Logic. Technical Report No.79-07-05, Dept. of Comp.Sci., University of Washington, Seattle 98195 (1979).

10 Bíró,B.: On the completeness of program verification methods. Bulletin of Section of Logic, Wroclaw, Vol 10, No 2, (1981) pp.83-90.

11 Bowen,K.A.: Model theory for modal logic (Kripke models for modal predicate calculi). Synthese Library Vol 127, Reidel Publ.Co., 1979.

12 Burstall,R.M. Darlington,J.: A system which automatically improves programs. In: Proc. 3rd IJCAI (SRI, 1973) pp.537-542.

13 Chang,C.C. Keisler,H.J.: Model Theory. North-Holland, 1973.

14 Cook,S.A.: Soundness and completeness of an axiom system for program verification. Siam J. Comput. 7 (1978) pp.70-90.

15 Courcelle,B. Guessarian,I.: On some classes of interpretations. J. Comput. System. Sci. 17 (1978) pp.388-413.

16 Csirmaz,L.: Completeness of Floyd-Hoare program verification. Preprint, Math.Inst.Hung.Acad.Sci. No 11/1980.

17 Csirmaz,L.: Structure of program runs of nonstandard time. Acta Cybernetica 4 (1980) pp.325-331.

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

19 Csirmaz,L. A survey of semantics of Floyd-Hoare derivability. Comput.Linguist.Comput.Lang. (Budapest) 14 (1980) pp.21-42.

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

21 Csirmaz,L.: Programs and program verifications in a general setting. Theoretical Computer Science, to appear in Vol 16 (1981). Abstracted in Bull.Sec.Logic Vol 9, No 3 (1980) pp.131-136.

22 Ecsedi-Tóth,P.: Intensional Logic of Actions. Comput. Linguist. Comput. Lang. (Budapest) Vol 12 (1978) pp.31-43.

23 van Emde-Boas,P., Janssen,T.M.V.: Intensional Logic of Programming. Preprint, Amsterdam, No ZW 98/78 (1978).

24 Gabbay,D.M.: Model Theory for Tense Logics. Annals of Math.Log. 8 (1975) pp.185-236.

25 Gallin,D.: Intensional and Higher Order Modal Logic. North-Holland - Americal Elsevier, New York, 1978.

26 Garson,J.W.: Completeness of some quantified modal logics. Logique et Analyse 22 (1979) pp.153-164.

27 Gergely,T. Szöts,M.: Model theoretic investigations in programming theory. Acta Cybernetica, Tom 4, Fasc 1 , Szeged 1979 pp.45-57.

28 Gergely,T. Úry,L.: Mathematical theory of programming. Manuscript, Budapest, 1978.

29 Gergely,T. Úry,L.: Time models for programming logic. In: Mathematical Logic in Computer Science (Proc.Coll.Salgótarján 1978) Colloq.Math.Soc.J.Bolyai 26, North-Holland, Amsterdam, 1981. pp.359-427.

30 Gergely,T. Úry,L.: Specification of program behaviour through explicit time considerations. In: Information Processing 80, North-Holland, IFIP 1980. pp.107-111.

30a Hájek,P.: Making dynamic logic first-order. In: Mathematical Foundations of Computer Science'81 (Proc. Strbské Pleso, Czechoslovakia) Lecture Notes in Computer Science 118, Springer, Berlin, 1981. pp.287-295.

31 Harel,D.: First order dynamic logic. Lecture Notes in Computer Science 68, Springer Verlag, Berlin, 1979.

32 Manna,Z.: Mathematical theory of computation. McGraw Hill, 1974.

33 Márkusz,Zs. Szöts,M.: Montague's intensional model theory applied to programming. In: Mathematical Logic in Computer Science (Proc. Coll. Salgótarján 1978) Colloq.Math.Soc.J.Bolyai 26, North-Holland, Amsterdam, 1981. pp.491.507.

34 Milne,R. Strachey,C.: A theory of programming language semantics. Chapman and Hall, 1976.

35 Monk,J.D.: Mathematical Logic. Springer Verlag, 1976.

36 Montague,R.: Formal Philosophy: Selected papers of R.Montague. E.H.Thomason, ed., Yale University Press, New Haven and London, 1974.

37 Németi,I.: Connections between algebraic logic and initial algebra semantics of CF languages. In: Mathematical Logic in Computer Science (Proc.Coll.Salgótarján 1978) Colloq.Math.Soc.J.Bolyai 26, North-Holland, Amsterdam, 1981. pp.25-83, 561-605.

37a Németi,I.: Nonstandard runs of Floyd provable programs. Math. Inst.Hung.Acad.Sci., Budapest, Preprint, 1980.

37b Németi,I.: Nonstandard dynamic logic. In: Proc. Workshop on Logics of Programs (May 1981, New York) Ed.: D.Kozen, Lecture Notes in Computer Sciences. to appear.

37c Németi,I.: Results on the lattice of dynamic logics. Preprint, Math.Inst.Hung.Acad.Sci., Budapest, 1981.

38 Parikh,R.: The completeness of propositional dynamic logic. In: Mathematical Foundations of Computer Science'78 (Proc. Zakopane, Poland) Lecture Notes in Computer Science 64, Springer Verlag, Berlin, 1978. pp.403-415.

39 Pratt,V.R.: A practical decision method for propositional dynamic logic. In: Proc. 10th Ann. ACM Symp. on Theory of Computing (San Diego CA, 1978) pp.326-337.

40 Pratt,V.R.: Models of program logics. In: 20th IEEE Conference on Foundations of Computer Science (San Juan PR, 1979).

41 Pratt,V.R.: Flowgraph logic and the elimination of Kleene elimination. MIT Preprint No 4/17/80, 1980.

42 Pratt,V.R.: Dynamic algebras and the nature of induction. In: 12th Ann. ACM Symp. on Theory of Computing (Los Angeles CA, 1980).

42a Richter,M.M. Szabo,M.E.: Towards a nonstandard analysis of programs. In: Proc. 2nd Victoria Symp. on Nonstandard Analysis (Victoria, Britisch Columbia, June 1980) Lecture Notes in Mathematics. Ed.: A. Hurd, Springer Verlag, Berlin, 1981.

43 Sain,I.: There are general rules for specifying semantics: Observations on Abstract Model Theory. Comput. Linguist. Comput. Lang. (Budapest) Vol 13 (1979) pp.251-282.

43a Sain,I.: First order dynamic logic with decidable proofs and workeable model theory. In: Fundamentals of Computation Theory'81 (Proc. Szeged, Hungary) Lecture Notes in Computer Science 117, Springer Verlag, Berlin, 1981. pp.334-341.

44 Segerberg,K.: A completeness theorem in the modal logic of programs. Abstract. Notices of the Americal Mathematical Society 24, 6 (Oct.1977) A-552.

45 Segerberg,K.: Applying modal logic. Studia Logica 39 (1980) pp. 275-296.

46 Wand,M.: A new incompleteness result for Hoare's system. JACM Vol 25, No 1 (1978).

47 Stavi,J.: Compactness properties of infinitary and abstract languages. In: A.Macintyre, L.Pacholski, J.Paris, eds., Logic Colloquium'77, North-Holland, 1978. pp.263-275.

48 Ianov,Y.I.: The logical schemes of algorithms. In: Problems of Cybernetics Vol 1 , Pergamon Press, New York 1960, English translation. pp.82-140.

49 Segerberg,K. (ed.): Trends in modal logic. A monothematic issue of Studia Logica, Vol 39, No 2/3 , 1980.

50 Salwicki,A.: Axioms of algorithmic logic univocally determine semantics of programs. In: P.Dembinski, ed., Mathematical Foundations of Computer Science'80, Lecture Notes in Computer Science 88, Springer Verlag, Berlin, 1980. pp.552-561.

51 Constable,R.L.: On the theory of programming logic. In: Proc. ACM STOC 9 (1977) pp.269-285.

List of definitions