NONSTANDARD DYNAMIC LOGIC

I. Németi

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

There does exist a branch of Dynamic Logic which is called Nonstandard Dynamic Logic. Works in this line are e.g. [4],[3],[22],[23],[1],[11],[9],[13],[18],[5],[14],[21]. A systematic introductory monograph with motivation, examples, overview of the field etc. is [4] which will be sent to anybody on request. A published introduction to Nonstandard DL with at least some of these features is [3]. Intuitive examples, illustrations are in [18],[20],[4]. The first results in this field were proved in [1] in 1977 under the restriction that the data structure satisfies Peano's axioms. This condition was later eliminated by the above quoted works.

In the present paper we give the basic definitions of Nonstandard DL (§1-3). We formulate some fundamental results and indicate that this logic is not so very nonstandard as one might think, see RDL in Def.13 and Prop.2. Then we show how to use this logic to compare methods of program verification. Some well known program verification methods will be characterized, see Fig.2. Some properties of the lattice of logics of programs with decidable proof concepts will be established. §5 contains the detailed proof of Thm.6. This proof uses model theoretic tools (e.g. ultraproducts) to establish properties of program verification methods. The emphasis is on basic definitions and properties of Nonstandard DL, on Fig.2, and on the proof of Thm.6. For intuitive motivation see the very end of the present paper.

Connections with other branches of nonclassical logic and computer science are discussed in §8,9 of [3]Part II and in §6-8 of [4]. Motivation for Nonstandard DL is e.g. in [22],[4],[3].

NOTATIONS

In the following we shall recall some standard notations from textbooks on logic (mainly from [17],[8]).

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

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

Natural numbers are used in the von Neumann sense, i.e.

$n = \{0,1,\ldots,n-1\}$ and in particular

0 is the empty set.

$X = \{x_w : w \in \omega\}$ 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. [8]p.22.

τ denotes a term of type d in the usual sense of logic, see [8]p.22 or [17]p.166,Def.10.8(ii).

M_d denotes the class of all classical one-sorted models of type d, see e.g. [8] or [17]Def.11.1, or Def.s 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 $g : \omega \to D$ is understood, see [17]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 [8]p.27,Def.13.13 or [17] 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 [22].

${}^{A}B$ denotes the set of all functions from A into B, i.e. ${}^{A}B = \{f : f \text{ maps } A \text{ into } B\}$, see [17]p.7.

A function is considered to be a set of pairs.

$\mathrm{Dom}\, f$ denotes the domain of the function f, $\mathrm{Dom}\, f \overset{d}{=} \{a : (\exists b)\langle a,b\rangle \in f\}$.

$\mathrm{Rng}\, f$ denotes the range of the function f, $\mathrm{Rng}\, f \overset{d}{=} \{b : (\exists a)\langle a,b\rangle \in f\}$.

A sequence s of lenght n is a function with $\mathrm{Dom}\, s = n$.

$\langle U_s : s \in S\rangle$ denotes the function $\{\langle s,U_s\rangle : s \in S\}$. Moreover for an expression $\mathrm{Expr}(x)$ and class S we define

$\langle \mathrm{Expr}(x) : x \in S\rangle$ to be the function $f : S \to \mathrm{Rng}\, f$ such that $(\forall x \in S)$ $f(x) = \mathrm{Expr}(x)$.

$\mathrm{Sb}(X) \overset{d}{=} \{Y : Y \subseteq X\}$ is the powerset of X.

X^{*} denotes the set of all finite sequences of elements of X, i.e. $X^{*} \overset{d}{=} \bigcup\{{}^{m}X : m \in \omega\}$. 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".

$A \sim B \overset{d}{=} \{a \in A : a \notin B\}$.

§1. SYNTAX of program schemes

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. (Lab is chosen this way for technical reasons only. There are many other possible ways for handling labels, see [23].) 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 follows:

$(i: x \leftarrow \tau) \in U_d$ 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 quantifier.

$(i: \text{HALT}) \in U_d$ 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 $p = \langle (i_0:u_0),\dots,(i_{n-1}:u_{n-1}),(i_n:\text{HALT})\rangle$ where $n \in \omega$, $(i_m:u_m) \in U_d$ for $m \leq n$ etc.

Convention 1 If a program scheme is denoted by p then its parts are denoted as follows:

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

Throughout we shall 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.

An example for a program scheme $p \in P_d$ is found in §5 in the proof of Thm.6 on Fig.3.

§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 the class of models or possible interpretations of L, and $\models$ 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 [22].

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 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 $ext : I\times T \to D$ 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 $ext : I\times T \to D$ is the "content of ... at time ..." function (see Def.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 [3]§9,[4]§8. For a detailed account of the above considerations see also §8,9 of [3] and §7,8 of [4].

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 Def.s 13-17. 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 (Def.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 $d_1 : \Sigma \to \omega$ for some set Σ, $H \subseteq \Sigma$ and $(\forall r \in \Sigma) d_1(r) \neq 0$.

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 $R(r) : {}^{(d(r)-1)}D \to D$.

Notation: $\langle D, R_r \rangle_{r \in \Sigma} \overset{d}{=} \langle D, \langle R_r : r \in \Sigma \rangle \rangle \overset{d}{=} \langle D, R \rangle$.

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) $M_d \overset{d}{=} \{ \underset{\sim}{D} : \underset{\sim}{D}$ 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, $t = \langle \{0, sc, +, \cdot\}, t_1 \rangle$ where $\mathrm{Dom}\, t_1 = \{\leq, 0, sc, +, \cdot\}$, $t(\leq)=2$, $t(0)=1$, $t(sc)=2$ and $t(+)=t(\cdot)=3$.

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

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 $Dom(d_1) \cap Dom(t_1) = 0$ is assumed throughout the paper.

DEFINITION 3 (many-sorted models, [17])

(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 $m_2 : \Sigma \to S^*$ 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 $\Sigma = Dom\ m_2$ 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 $Dom\ R = \Sigma$ and if $r \in \Sigma$ and $m(r) = \langle s_1, \ldots, s_n \rangle$ then $R(r) \subseteq U_{s_1} \times \ldots \times U_{s_n}$ and if in addition $r \in H$ then $R(r)$ is a function $R(r) : U_{s_1} \times \ldots \times U_{s_{n-1}} \to U_{s_n}$.

U_s is said to be the universe of sort s of $\mathfrak{M}$.

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

End of Definition 3

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 Def.2 and by our convention $Dom(d_1) \cap Dom(t_1) = 0$.

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

a) $S \stackrel{d}{=} \{t, d, i\}$, $|S| = 3$. (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) $K \stackrel{d}{=} \{ext, 0, sc, +, \cdot\} \cup H$. ($K$ is the set of operation symbols of td.)

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

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

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

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

Notation: $\langle\langle U_t, R_r\rangle_{r\in Dom(t_1)}, \langle U_d, R_r\rangle_{r\in Dom(d_1)}, U_i, R_{ext}\rangle \overset{d}{=}$
$\overset{d}{=} \langle\langle U_t, U_d, U_i\rangle, R_r\rangle_{r\in\Sigma}$.

We define: $\underset{\sim}{T} \overset{d}{=} \langle U_t, R_r\rangle_{r\in Dom(t_1)}$, $T \overset{d}{=} U_t$,
$\underset{\sim}{D} \overset{d}{=} \langle U_d, R_r\rangle_{r\in Dom(d_1)}$, $D \overset{d}{=} U_d$ and $I \overset{d}{=} U_i$.

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

Convention 2 Whenever an element of M_{td} is denoted by the letter $\mathfrak{M}$ then the parts of $\mathfrak{M}$ are denoted as follows:
$\langle \underset{\sim}{T}, \underset{\sim}{D}, I, ext\rangle \overset{d}{=} \langle\langle U_t^{\mathfrak{M}}, U_d^{\mathfrak{M}}, U_i^{\mathfrak{M}}\rangle, r^{\mathfrak{M}}\rangle_{r\in\Sigma} \overset{d}{=} \mathfrak{M}$.
Note that $\mathfrak{M}\in M_{td}$ iff $[\underset{\sim}{T}\in M_t, \underset{\sim}{D}\in M_d$, and $ext : I\times T \to D]$.

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 [17]. If understanding Def.s 3-6 here is hard for the reader then consulting [17] should help since L_{td} is the most usual classical many-sorted language of similarity type td.

DEFINITION 5 (the first order 3-sorted language $L_{td} = \langle F_{td}, M_{td}, \models\rangle$ of type td, [17])

Let $d = \langle H, d_1\rangle$ be any one-sorted similarity type. Recall from Def.s 3 and 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 $X \overset{d}{=} \{x_w : w\in\omega\}$, $Y \overset{d}{=} \{y_w : w\in\omega\}$ and $Z \overset{d}{=} \{z_w : w\in\omega\}$ 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) $f(\tau_1,\dots,\tau_n) \in Tm_{td,d}$ for any $f \in H$ if $d(f)=n+1$ and $\tau_1,\dots,\tau_n \in Tm_{td,d}$.

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) $r(\tau_1,\dots,\tau_n) \in F_{td}$ for any $\tau_1,\dots,\tau_n \in Tm_{td,d}$ and for any $r \in H$ 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) $\{\neg\varphi, (\varphi\wedge\psi), (\exists z_w\varphi), (\exists x_w\varphi), (\exists y_w\varphi) : w \in \omega\} \subseteq F_{td}$ for any $\varphi, \psi \in F_{td}$.

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 [17]) which is completely analogous with the one-sorted case. E.g.

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

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

$\mathfrak{M} \models \varphi[g,k,r]$ 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 class STM_d of standard models)

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) $\underset{\sim}{T} = \underset{\sim}{N}$. (For $\underset{\sim}{N}$ see Def.2.)

(ii) $I = {}^{\omega}D$.

(iii) $(\forall s\in I)(\forall b\in T)\ ext(s,b) = s(b)$.

The class of all standard elements of M_{td} is denoted by STM_d.

End of Definition 6

In this paper we shall define several sets of axioms in the language L_{td}, see Def.s 13-17. Each of them will be valid in the class STM_d of standard models.

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

Notation: Let $\langle \underset{\sim}{T},\underset{\sim}{D},I,ext\rangle \in M_{td}$, see Convention 2. Let $s_0,\dots,s_m \in I$, $\bar{s} \overset{d}{=} \langle s_0,\dots,s_m\rangle$. Let $b\in T$. Then we define
$ext(\bar{s},b) \overset{d}{=} \langle ext(s_0,b),\dots,ext(s_m,b)\rangle$.

DEFINITION 7 (traces of programs in time-models)

Let $p\in P_d$ and $\mathfrak{M}\in M_{td}$. We shall use Conventions 1 and 2. Let $s_0,\dots,s_c \in I$ be arbitrary intensions in $\mathfrak{M}$. Let $\bar{s} = \langle s_0,\dots,s_{c-1}\rangle$. The sequence $\langle s_0,\dots,s_c\rangle$ of intensions is defined to be a trace of p in $\mathfrak{M}$ if the following (i) and (ii) are satisfied.

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

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

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

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

(3) If $u_m =$ "HALT" then $ext(s_j,b+1) = ext(s_j,b)$.

End of Definition 7

DEFINITION 8 (possible output)

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 <u>of input k</u> iff $(\forall j < c)\ k(j) = ext(s_j, 0)$.

(ii) Recall from Convention 1 that i_n is the label of the HALT-command of p. Let $b \in T$. We say that <u>s terminates at time b in $\mathfrak{M}$</u> iff $ext(s_c, b) = i_n$.

(iii) Let $k, q \in {}^{\omega}D$. We define q to be a <u>possible output</u> of p <u>with input</u> 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) $(\forall j \in \omega)[j \geq c \Rightarrow q_j = k_j]$.

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 $\langle k_0, \dots, k_{c-1} \rangle$. <u>End of Definition 8</u>

By now we have defined a semantics of program schemes.

<u>Remark</u>: 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 "<u>history</u>" 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 $ext(s_w, -) : T \to D$ from time points to data values D. This function is the "history" of the variable x_w during an <u>execution of the program p in the model $\mathfrak{M}$</u>. Def.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".

<u>About using Th.</u>: It might look counter-intuitive to execute programs in arbitrary elements of M_{td}. However, we can collect <u>all our postulates about time</u> into a set $Ax \subseteq F_{td}$ of axioms which this way would define the class $Mod(Ax) \subseteq M_{td}$ of all <u>intended interpretations</u> of P_d. Then traces of programs in $Mod(Ax)$ provide an intuitively acceptable semantics of program schemes. Such a set Ax of axioms will be proposed in Def.13. <u>If</u> 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 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 Def.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 Def.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 $DL_d \overset{d}{=} \langle DF_d, M_{td}, \models \rangle$ 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 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. E.g. $\Box(p,\psi)$ expresses that p is partially correct w.r.t. output condition ψ, and $\Diamond(p,\psi)$ expresses that p is totally correct w.r.t. output condition ψ in the weaker sense.

Convention 3 We shall use the model theoretic consequence relation $\models$ in the usual way. I.e. let $Th \subseteq DF_d$, $\varphi \in DF_d$ and $K \subseteq M_{td}$. Then

$$\mathfrak{M} \models \varphi \quad \text{iff} \quad (\forall g \in {}^{\omega}T)(\forall k \in {}^{\omega}D)(\forall r \in {}^{\omega}I)\ \mathfrak{M} \models \varphi[g,k,r],$$
$$\mathfrak{M} \models Th \quad \text{iff} \quad (\forall \varphi \in Th)\ \mathfrak{M} \models \varphi,$$
$$K \models Th \quad \text{iff} \quad (\forall \mathfrak{M} \in K)\ \mathfrak{M} \models Th,$$
$$Mod(Th) \overset{d}{=} Mod_{td}(Th) \overset{d}{=} \{\mathfrak{M} \in M_{td} : \mathfrak{M} \models Th\}, \quad \text{and}$$
$$Th \models \varphi \quad \text{iff} \quad Mod(Th) \models \varphi .$$

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 $Mod(Th) = Mod_h(Th)$.

DEFINITION 10 (proof concept [17])

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 $w \in F^{*})]$. Recall that we identify F^{*} with $\{H \in Sb(F) : |H| < \omega\}^{*}$.

The proof concept $(\vdash, Pr)$ 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 " $(\vdash,Pr)$ 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.

THEOREM 1 (strong completeness of DL_d)

There is a decidable proof concept $(\overset{N}{\vdash}, Prn)$ for the language DL_d such that for every $Th \subseteq DF_d$ and $\varphi \in DF_d$ we have $[Th \models \varphi$ iff $Th \overset{N}{\vdash} \varphi]$.

Proof: can be found in [[illegible]], as well as in [4]Thm.2 pp.30-38. QED

DEFINITION 11 (the proof concept $(\overset{N}{\vdash}, Prn)$ of DL_d)

By Thm.1 above there exists a decidable set $Prn \subseteq (DF_d)^*$ such that $(\forall Th \subseteq DF_d)(\forall \varphi \in DF_d)[\ Th \overset{N}{\vdash} \varphi$ iff $(\exists$ finite $H \subseteq Th)(\exists w)\ \langle H,w,\varphi\rangle \in Prn\ ]$.

The decision algorithm for Prn is rigorously constructed in [3] Thm.2, and [4]Thm.2,pp.30-38, and in [19].

From now on we shall use Prn as defined in the quoted papers. The only important properties of Prn we shall use are its decidability and its completeness for DL_d. End of Definition 11

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 Def.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 $\langle DL_d\ ,\ (\overset{N}{\vdash},Prn)\rangle$ where the proof concept $(\overset{N}{\vdash},Prn)$ is defined in Def. 11. By Theorem 1, First order Dynamic Logic $\langle DL_d,(\overset{N}{\vdash},Prn)\rangle$ is complete.

Given any logic, say $\langle DL_d, \overset{N}{\vdash}\rangle$, decidable sets $Ax \subseteq DF_d$ of formulas (i.e. theories Ax) give rise to new logics. We shall make this precise in Def.12 below.

DEFINITION 12 (new proof concepts $(Ax\overset{N}{\vdash})$ from old $\overset{N}{\vdash}$, $DL_d(Ax)$, $Dlog_d(Ax)$)

Let $Ax \subseteq DF_d$ be decidable but otherwise arbitrary.

(i) Let $Th \subseteq DF_d$ and $\varphi \in DF_d$ be arbitrary. We say that φ is $(Ax\overset{N}{\vdash})$-provable from Th iff $Th \cup Ax \overset{N}{\vdash} \varphi$. That is φ is provable by the proof concept $(Ax\overset{N}{\vdash})$ from Th iff $Th \cup Ax \overset{N}{\vdash} \varphi$. Thus $(Ax\overset{N}{\vdash})$ is a new recursively enumerable "provability" relation.

(ii) $pf(Ax\overset{N}{\vdash}) \overset{d}{=} \{\langle H,\langle L,w\rangle,\varphi\rangle \in (DF_d)^* : \langle H \cup L,w,\varphi\rangle \in Prn$ and $L \subseteq Ax\}$.

Clearly φ is $(Ax\overset{N}{\vdash})$-provable from Th iff $(\exists\langle H,w,\varphi\rangle \in Prn)\ H \subseteq Th \cup Ax$.

Clearly $pf(Ax\overset{N}{\vdash})$ is a decidable subset of $(DF_d)^*$.

(iii) We have defined a new proof concept $\langle(Ax\overset{N}{\vdash})\ ,\ pf(Ax\overset{N}{\vdash})\rangle$ where $pf(Ax\overset{N}{\vdash})$ is the decidable set of all $(Ax\overset{N}{\vdash})$-proofs. We shall always denote this new proof concept by $(Ax\overset{N}{\vdash})$. So whenever we write $(Ax\overset{N}{\vdash})$ we shall mean $\langle(Ax\overset{N}{\vdash}),pf(Ax\overset{N}{\vdash})\rangle$ but we shall not write it out explicitly.

(iv) We define the new language $DL_d(Ax)$ associated to $Ax \subseteq DF_d$ to be $DL_d(Ax) \overset{d}{=} \langle DF_d\ ,\ Mod_{td}(Ax)\ ,\ \models\rangle$.

(v) We define the new dynamic logic $Dlog_d(Ax)$ associated to Ax to be $Dlog_d(Ax) \stackrel{d}{=} \langle DL_d(Ax), (Ax \stackrel{N}{\vdash})\rangle$. End of Definition 12

On Figure 2, different proof concepts $(Ax_1 \stackrel{N}{\vdash})$, $(Ax_2 \stackrel{N}{\vdash})$ etc. will be compared with each other as well as with such classic proof concepts as Floyd's $\stackrel{F}{\vdash}$ and Rod Burstall's $\stackrel{mod}{\vdash}$.

DEFINITION 13 (Dax, Reasonable Dynamic Logic, $\stackrel{\omega}{\models}$)

In Def.s 14-17 below the axiom systems Ia, Tpa, Ex, $\{Axe\} \subseteq DF_d$ will be defined. We define the logical axioms of Reasonable Dynamic Logic to be $Dax \stackrel{d}{=} Ia \cup Tpa \cup Ex \cup \{Axe\}$.

We define Reasonable Dynamic Logic to be $Dlog_d(Dax)$. See Def.12 (v) above.

Let $Th \subseteq DF_d$ and $\varphi \in DF_d$. Then we define $[\, Th \stackrel{\omega}{\models} \varphi$ iff $(STM_d \cap Mod(Th)) \models \varphi\,]$. End of Definition 13

Note that $STM_d \models Dax$ is easy to prove.

Is our dynamic logic nihilistic or counterintuitive?:

We claim that the answer is no for our Reasonable Dynamic Logic $Dlog_d(Dax)$. To execute programs in arbitrary elements of M_{td} might look counterintuitive. However $Dlog_d(Dax)$ is a complete logic with decidable proof concept and there is nothing wrong with executing programs in elements of $Mod_{td}(Dax)$. See e.g. Prop.2 below, Thm.7 of [3], Thm.6 of [9]p.34 and Fig.2.

PROPOSITION 2 Let $\mathfrak{M} \models Dax$ and $p \in P_d$. Then (i)-(ii) below hold.

(i) To every input q there is exactly one trace of p in $\mathfrak{M}$ with input q.

(ii) Assume that the trace $s \in {}^m I$ of p in $\mathfrak{M}$ terminates at time $b \in T$. Then $(\forall a \in T)[\, b \leq a \Rightarrow (\forall i < m) ext(s_i, b) = ext(s_i, a)\,]$ and $(\exists a \in T)(\forall k \in T)[\,(s \text{ terminates } p \text{ at time } k) \Leftrightarrow a \leq k\,]$.

Proof: Detailed proofs can be found in [3]Thm.s 3-4, [4]Thm.s 3-4, pp. 42-45, except for the existence of traces in (i) which is proved in [20], but the idea of this proof is available in [3]proof of Thm.7.
QED(Proposition 2)

On Fig.2, different dynamic logics $Dlog_d(Ax)$ with various $Ax \subseteq DF_d$ will be compared with each other and with classical logics of programs like Floyd-Hoare Logic, Burstall's modal-dynamic logic etc.

§4. Comparing methods for program verification, the status of some well known ones

We shall show how to use our logic DL_d to compare powers of methods of program verification, as well as to generate new methods for program verification. We shall see that the program verification methods form a lattice, see Fig.2. It might be interesting and also useful to find out about well known program verification methods how they are situated in this lattice.

Three well known program verification methods we shall look at are Floyd's inductive assertions method $\overset{F}{\vdash}$, Burstall's time modalities method $\overset{mod}{\longmapsto}$ [7], and Future-enriched time modalities method $\overset{fum}{\longmapsto}$ [12]. Burstall's $\overset{mod}{\longmapsto}$ is often called intermittent-assertion method, see e.g. [16]. These methods will be defined rigorously, see Def.20 for $\overset{F}{\vdash}$, Def.18 for $\overset{mod}{\longmapsto}$, and Def.19 for $\overset{fum}{\longmapsto}$. The last one, $\overset{fum}{\longmapsto}$, is $\overset{mod}{\longmapsto}$ enriched with future tense and past tense. By spotting the precise locations of $\overset{F}{\vdash}$, $\overset{mod}{\longmapsto}$ and $\overset{fum}{\longmapsto}$ in the lattice of program verification methods we shall find a precise answer to the question asked at SRI in 1976: "Is sometime sometimes better than always?" [16].

We have to fix the criteria to be used when we compare program verification methods. We shall say that one method $\vdash_1$ is stronger than another $\vdash_2$ iff more programs can be proved to be partially correct by $\vdash_1$ than by $\vdash_2$. So we shall consider the reasoning power to prove partial correctness statements $\varphi \rightarrow \Box(p,\psi)$ to be the criterion to compare different methods. This choice has nothing to do with our logic DL_d, namely DL_d is suitable for proving total correctness of programs. It was proved in [3]Thm.7 and in Thm.7 of [4] that the Kfoury-Park[15] negative result on proving total correctness is <u>not</u> true for DL_d.

We shall consider program verification methods only with decidable proof concepts.

About generating new program verification methods by DL_d.: A safe way of dreaming up new sound program verification methods is to define a decidable set $Ax \subseteq DF_d$ of axioms such that $STM_d \models Ax$. Then $(Ax \overset{N}{\vdash})$ is a sound program verification method. A reasonable axiom system is e.g. Dax introduced in Def.13. Clearly $STM_d \models Dax$. Thus by Thm.1 we can be sure that whenever $Th \cup Dax \overset{N}{\vdash} \Box(p,\psi)$ then really $Th \overset{\omega}{\models} \Box(p,\psi)$ that is the proof method $(Dax \overset{N}{\vdash})$ is sound.

Below we shall introduce several such axiom systems Ax, with $STM_d \models Ax$. Later we shall compare them in Fig.2. One can consider these axiom systems as different candidates for being the logical axioms for dynamic logic. Or if we want to imitate what people do in modal logic then we could say that every recursively enumerable $Ax \subseteq DF_d$ such that $STM_d \models Ax$ is a dynamic logic and if $STM_d \models Ax_1$ and $STM_d \models Ax_2$ and $Ax_1 \not\models Ax_2$ then Ax_1 and Ax_2 are two different dynamic logics and if $Ax_2 \models Ax_1$ then Ax_2 is a dynamic logic stronger than Ax_1.

Usually, any axiom system, say Axname, introduced below will consist of two parts Tname and Iname such that Axname = Tname ∪ Iname. Tname consists of postulates about the time structure $\underset{\sim}{T}$ hence Tname $\subseteq F_t^Z$, see Def.16. Iname consists of induction axioms about the intensions, see Def.15. Typical examples are $\forall z(sc(z) \neq 0) \in$ Tname and $(x=ext(y,0) \wedge \forall z[x=ext(y,z) \rightarrow x=ext(y,sc(z))]) \rightarrow \forall z(x=ext(y,z)) \in$ Iname.

<u>DEFINITION 14</u> (ind(φ,z), IA, Ia, Lax)

Let d be a similarity type. Then td, F_{td} and Z were defined in Def.s 4 and 6 in §2. Let $z \in Z$ be arbitrary. Let $\varphi \in F_{td}$. We define the induction formula ind(φ,z) as follows:

$$ind(\varphi,z) \stackrel{d}{=} ([\varphi(0) \wedge \forall z(\varphi \rightarrow \varphi(sc(z))] \rightarrow \forall z \varphi) ,$$

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

The induction axioms are:

$IA \stackrel{d}{=} \{ ind(\varphi,z) : \varphi \in F_{td}$ and $z \in Z\}$.

$Lax \stackrel{d}{=} \{(j \neq k) :$ j and k are two different elements of Lab}.

$Ia \stackrel{d}{=} IA \cup Lax$. <u>End of Definition 14</u>

Clearly $IA \subseteq F_{td}$ since if $\varphi \in F_{td}$ and $z \in Z$ then $\varphi(0), \varphi(sc(z)) \in \in F_{td}$ because 0 and sc(z) are terms of sort t. It is important to stress here that φ may contain other free variables of all sorts. All the free variables of φ are also free in ind(φ,z) except for z. They are the "<u>parameters</u>" of the induction ind(φ,z).

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

Our strongest set of induction axioms is Ia. We shall distinguish various subsets of Ia.

<u>DEFINITION 15</u> (Iq, $I\Sigma_1$, $I\Pi_1$, If, I1, I′, Ict, Imd and Ifm)

$If \stackrel{d}{=} \{\varphi \in IA :$ φ contains no free variable of sort t or d} ∪ Lax.

$I1 \stackrel{d}{=} \{\varphi \in IA : (\forall i \in \omega)[i > 0 \Rightarrow z_i$ does not occur in φ neither free nor bound]} ∪ Lax.

$I' \overset{d}{=} \{ind(\varphi, z_0) \in I1 \;:\; \varphi \in F_{td}$ is such that "+" and "·" do not occur in φ and there is no subformula $\psi \in F_t^Z$ of $\varphi\} \cup Lax$.

$Ict \overset{d}{=} \{ind(\exists x_0 \ldots x_m[(\bigwedge_{i \leq m} x_i = ext(y_i, z_0)) \wedge \varphi], z_0) \;:\; m \in \omega$ and $\varphi \in F_d\} \cup Lax$.

Let $(\Sigma_{0,t}F_{td}) \overset{d}{=} \{\varphi \in F_{td} \;:\; \varphi$ contains no quantifier of sort t, that is $(\forall i \in \omega)[$"$\exists z_i$" does not occur in $\varphi]\}$.

$Iq \overset{d}{=} \{ind(\varphi, z_0) \;:\; \varphi \in (\Sigma_{0,t}F_{td})\} \cup Lax$.

$I\Sigma_1 \overset{d}{=} \{ind(\exists z_1 \ldots z_m \varphi, z_0) \;:\; \varphi \in (\Sigma_{0,t}F_{td})$ and $m \in \omega\} \cup Lax$.

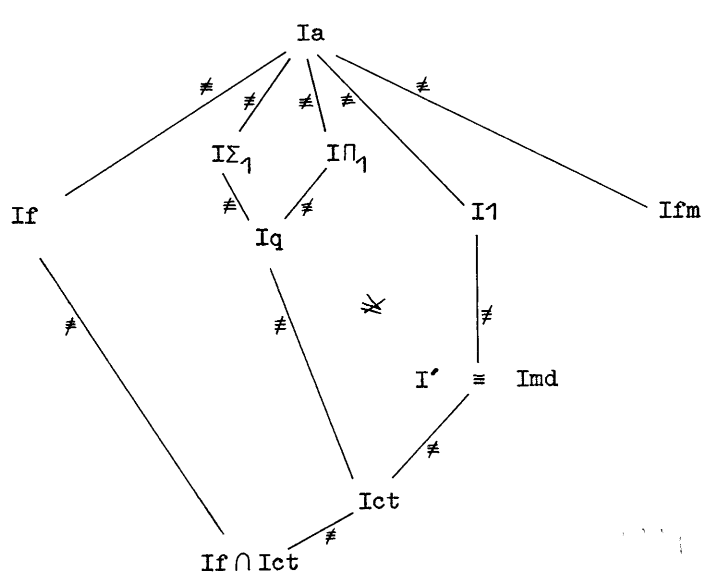
$I\Pi_1 \overset{d}{=} \{ind(\forall z_1 \ldots z_m \varphi, z_0) \;:\; \varphi \in (\Sigma_{0,t}F_{td})$ and $m \in \omega\} \cup Lax$.

$Imd \overset{d}{=} \{mod\,\varphi \;:\; \varphi \in IA^{mod}\}$, where mod and IA^{mod} will be defined in Def.18.

$Ifm \overset{d}{=} \{fum\,\varphi \;:\; \varphi \in Ifum\}$, where fum and Ifum will be defined in Def.19.

<u>End of Definition 15</u>

On Fig.1 we compare the sets of induction axioms introduced in Def. 15 above. <u>Warning</u>: As opposed to Fig.2, the comparison on Fig.1 is <u>not</u> modulo partial correctness of programs but instead it is absolute. That is, on Fig.1, $[I_1 \geqslant I_2$ iff $I_1 \models I_2]$ and $I_1 \equiv I_2$ means $(I_1 \leqslant I_2$ and $I_2 \leqslant I_1)$. The sign $\not\equiv$ indicates that the inequality in question is known to be proper, that is $I_2 \not\equiv I_1$. We shall discuss Fig.1 after the discussion of Fig.2 in §5.



<u>FIGURE 1</u>

DEFINITION 16 ($Ts \subseteq To \subseteq Tpres \subseteq Tpa \subseteq F_t^Z$ and Tfm)

Notation: $sc^0(z_0) \overset{d}{=} z_0$ and $(\forall n \in \omega) sc^{n+1}(z_0) \overset{d}{=} sc(sc^n(z_0))$.

$$Ts \overset{d}{=} \{ z_0 \neq 0 \leftrightarrow \exists z_1(z_0 = sc(z_1)),\ \ sc(z_0)=sc(z_1) \rightarrow z_0 = z_1,\ \ sc^n(z_0) \neq z_0 \ : \ : \ n \in \omega,\ n \neq 0 \}.$$

$$To \overset{d}{=} \{ (z_0 \leq z_1 \wedge z_1 \leq z_2) \rightarrow z_0 \leq z_2,\ \ (z_0 \leq z_1 \wedge z_1 \leq z_0) \rightarrow z_0 = z_1,\ z_0 \leq z_1 \vee z_1 \leq z_0,\ \ 0 \leq z_0,\ \ (z_0 \leq z_1 \wedge z_0 \neq z_1) \leftrightarrow sc(z_0) \leq z_1,\ 0 = z_0 \vee \exists z_1(z_0 = sc(z_1)) \}.$$

Tpres is the decidable set of Presburger's axioms for $\underset{\sim}{N}$:

$$Tpres \overset{d}{=} To \cup \{ z_0+0=z_0,\ \ z_0+sc(z_1)=sc(z_0+z_1),\ \ ind(\varphi, z_0) \ : \ \varphi \in F_t^Z \text{ and "}\cdot\text{" does not occur in } \varphi \}.$$

Tpa is the set of Peano's axioms formulated in the language F_t^Z about the similarity type t, see e.g. Example 1.4.11 in [8]p.42.:

$$Tpa \overset{d}{=} Tpres \cup \{ z_0 \cdot 0 = 0,\ \ z_0 \cdot sc(z_1) = z_0 \cdot z_1 + z_0,\ ind(\varphi, z_0) \ : \ \varphi \in F_t^Z \}.$$

$Tfm \overset{d}{=} \{ fum\varphi \ : \ \varphi \in Tfum \}$, where fum and Tfum will be defined in Def.19. End of Definition 16

Note that $Ts \subseteq To$ is not literally true but $To \models Ts$. We require $To \subseteq Tpa$ because we have the symbol $\leq$ in the similarity type t. We also note that $To \models Tfm$, and, clearly, $STM_d \models Tpa$. I.e. Fact 16.1 below holds.

FACT 16.1 $STM_d \models Tpa \models Tpres \models To \models Tfm$ and $To \models Ts$.

The set Ex of axioms introduced below are useful to prove total correctness, see Thm.7 of [3]Part II, and Thm.7 of [4].

DEFINITION 17 (Ex, Axe)

$$Ex \overset{d}{=} \{ [\forall z_0 \exists x_0 \varphi \ \rightarrow \ \exists y_0 \forall z_0 \exists x_0 (x_0 = ext(y_0, z_0) \wedge \varphi)] \ : \ \varphi \in F_{td} \text{ and } y_0 \text{ does not occur in } \varphi \}.$$

More intuitively, the formulas in Ex are of the form

$$\forall \bar{z} \forall \bar{x} \forall \bar{y} [\forall z_0 \exists x_0 \varphi(z_0, x_0, \bar{z}, \bar{x}, \bar{y}) \ \rightarrow \ \exists y_0 \forall z_0 \varphi(z_0, ext(y_0, z_0), \bar{z}, \bar{x}, \bar{y})]$$

where $\bar{z}, \bar{x}, \bar{y}$ are arbitrary sequences of variables not containing z_0, x_0, y_0.

Axe denotes the axiom of extensionality, i.e. Axe is

$(\forall y_0 \forall y_1 [\forall z_0 ext(y_0, z_0) = ext(y_1, z_0) \ \rightarrow \ y_0 = y_1]$. End of Definition 17

For the rest of this section, let $d = \langle H, d_1 \rangle$ be an arbitrary but fixed one-sorted similarity type, see Def.1.

A direct Kripke style semantics for DL_d^{mod} defined below can be found in [23]. Moreover, in [23] a direct Kripke style definition is given for the validity relation $\overset{mod}{\models}$ defined indirectly in Def.18.

DEFINITION 18 (modal dynamic language DL_d^{mod} of type d)

(i) Syntax DF_d^{mod} .:

T_d^{mod} is defined to be the smallest set satisfying (1)-(2) below:

(1) $\{x_n, y_n\} \subseteq T_d^{mod}$ for every $n \in \omega$.

(2) $f(\tau_1,\dots,\tau_n) \in T_d^{mod}$ for every $f \in H$ if $d(f)=n+1$ and $\{\tau_1,\dots,\tau_n\} \subseteq T_d^{mod}$.

DF_d^{mod} is defined to be the smallest set satisfying (3)-(5) below:

(3) $(\tau=\sigma) \in DF_d^{mod}$ for all $\tau, \sigma \in T_d^{mod}$.

(4) $R(\tau_1,\dots,\tau_n) \in DF_d^{mod}$ for every $R \in \mathrm{Dom}\, d_1$ if $R \notin H$, $d(R)=n$ and $\{\tau_1,\dots,\tau_n\} \subseteq T_d^{mod}$.

(5) $\{\mathrm{Alw}\varphi, \mathrm{First}\varphi, \mathrm{Next}\varphi, \exists x_n\varphi, \exists y_n\varphi, \neg\varphi, (\varphi\wedge\psi), \Box(p,\varphi)\} \subseteq DF_d^{mod}$ for all $n \in \omega$ and for all $\varphi,\psi \in DF_d^{mod}$ and all $p \in P_d$.

(ii) Translation function $mod : DF_d^{mod} \to DF_d$.:

The definition goes by recursion on the structure of DF_d^{mod}. Sometime we write $mod\varphi$ instead of $mod(\varphi)$. Let $n \in \omega$, $\tau_1,\dots,\tau_n \in T_d^{mod}$, $\varphi,\psi \in DF_d^{mod}$ and $p \in P_d$. Now

$mod(y_n) \overset{d}{=} ext(y_n,z_0)$, $mod(x_n) \overset{d}{=} x_n$,

$mod(\mathrm{Alw}\varphi) \overset{d}{=} \forall z_0(mod\varphi)$,

$mod(\mathrm{First}\varphi) = \exists z_0(z_0=0 \wedge mod\varphi)$,

$mod(\mathrm{Next}\varphi) \overset{d}{=} \exists z_1(z_1=sc(z_0) \wedge \exists z_0(z_0=z_1 \wedge mod\varphi)$,

$mod(g(\tau_1,\dots,\tau_n)) \overset{d}{=} g(mod\tau_1,\dots,mod\tau_n)$ if $g \in \mathrm{Dom}\, d_1$ is such that $d(g)=n+1$ in case $g \in H$ and $d(g)=n$ in case $g \notin H$,

$mod(\tau_1=\tau_2) \overset{d}{=} (mod\tau_1=mod\tau_2)$, $mod(\exists x_n\varphi) \overset{d}{=} \exists x_n mod\varphi$, $mod(\exists y_n\varphi) \overset{d}{=} \exists y_n mod\varphi$,

$mod(\neg\varphi) \overset{d}{=} \neg mod\varphi$, $mod(\varphi\wedge\psi) \overset{d}{=} (mod\varphi \wedge mod\psi)$, $mod(\Box(p,\psi)) \overset{d}{=} \Box(p,mod\psi)$.

By the above, the function $mod : DF_d^{mod} \to DF_d$ is fully defined.

(iii) Validity relation $\overset{mod}{\models} \subseteq M_{td} \times DF_d^{mod}$.:

Let $\mathfrak{M} \in M_{td}$ and $\varphi \in DF_d^{mod}$. Then we define $\mathfrak{M} \overset{mod}{\models} \varphi$ iff $\mathfrak{M} \models mod\,\varphi$.

(iv) Axioms IA^{mod} of modal dynamic logic .:

$IA^{mod} \overset{d}{=} \{ ([\mathrm{First}\varphi \wedge \mathrm{Alw}(\varphi \to \mathrm{Next}\varphi)] \to \mathrm{Alw}\varphi) : \varphi \in DF_d^{mod}\} \cup Lax.$

(v) The language DL_d^{mod} of modal dynamic logic .:

$DL_d^{mod} \stackrel{d}{=} \langle DF_d^{mod}, Mod_{td}(IA^{mod}), \models^{mod} \rangle$, where for any $Th \subseteq DF_d^{mod}$ we define $Mod_{td}(Th) \stackrel{d}{=} \{\mathfrak{M} \in M_{td} : \mathfrak{M} \models^{mod} Th\}$. Let $Th \subseteq DF_d^{mod}$ and $\varphi \in DF_d^{mod}$. Then $Th \models^{mod} \varphi$ is defined to hold iff $Mod_{td}(IA^{mod} \cup Th) \models^{mod} \varphi$, see Convention 3. <u>End of Definition 18</u>

<u>PROPOSITION 3</u> (completeness of DL_d^{mod})
Let $Th \subseteq DF_d^{mod}$ and $\varphi \in DF_d^{mod}$. Then
$Th \models^{mod} \varphi$ iff $\{mod\psi : \psi \in Th \cup IA^{mod}\} \models^{N} mod\varphi$.

<u>The proof</u> of Prop.3 is immediate by the definitions and by the completeness theorem of DL_d, i.e. by Thm.1. <u>QED</u>

The modality symbol Alwfu used below intuitively means "<u>Alw</u>ays in the <u>fu</u>ture". Similarly Alwpaφ intuitively means "<u>Alw</u>ays in the <u>pa</u>st φ". In [12] "Alwfuφ" and "Nextφ" are denoted by "Fφ" and "Xφ" respectively.

<u>DEFINITION 19</u> (future enriched modal dynamic language DL_d^{fum} of type d)

(i) <u>Syntax</u>.: DF_d^{fum} is defined to be the smallest set satisfying (1)-(2) below:

(1) $DF_d^{mod} \subseteq DF_d^{fum}$.

(2) $\{Alwfu\varphi, Alwpa\varphi, Alw\varphi, First\varphi, Next\varphi, \exists x_n\varphi, \exists y_n\varphi, \neg\varphi, (\varphi \wedge \psi), \Box(p,\psi)\} \subseteq DF_d^{fum}$ for all $n \in \omega$, $\varphi,\psi \in DF_d^{fum}$ and all $p \in P_d$.

(ii) <u>Translation function</u> $fum : DF_d^{fum} \to DF_d$.:
The definition of fum goes by recursion on the structure of DF_d^{fum}. Sometime we write fumφ instead of fum(φ), i.e. $fum\varphi \stackrel{d}{=} fum(\varphi)$.
Let $\varphi \in DF_d^{mod}$. Then $fum(\varphi) \stackrel{d}{=} mod(\varphi)$, see Def.18(ii).
Let $n \in \omega$, $\varphi,\psi \in DF_d^{fum}$ and $p \in P_d$. Then
$fum(Alwfu\varphi) \stackrel{d}{=} \forall z_1[z_1 \geq z_0 \to \exists z_0(z_0 = z_1 \wedge fum\varphi)]$,
$fum(Alwpa\varphi) \stackrel{d}{=} \forall z_1[z_1 \leq z_0 \to \exists z_0(z_0 = z_1 \wedge fum\varphi)]$,
$fum(Alw\varphi) \stackrel{d}{=} \forall z_0(fum\varphi)$, $fum(Next\varphi) \stackrel{d}{=} \exists z_1[z_1 = sc(z_0) \wedge \exists z_0(z_0 = z_1 \wedge fum\varphi)]$,
$fum(First\varphi) \stackrel{d}{=} \exists z_0(z_0 = 0 \wedge fum\varphi)$, $fum(\exists x_n\varphi) \stackrel{d}{=} \exists x_n fum\varphi$, $fum(\exists y_n\varphi) \stackrel{d}{=} \exists y_n fum\varphi$
$fum(\neg\varphi) \stackrel{d}{=} \neg fum\varphi$, $fum(\varphi \wedge \psi) \stackrel{d}{=} ((fum\varphi) \wedge (fum\psi))$, $fum(\Box(p,\psi)) \stackrel{d}{=} \Box(p, fum\psi)$.
By the above the function $fum : DF_d^{fum} \to DF_d$ is fully defined.

(iii) <u>Validity relation</u> $\models^{fum} \subseteq M_{td} \times DF_d^{fum}$.:

Let $\mathfrak{M}\in M_{td}$ and $\varphi\in DF_d^{fum}$. Then we define $\mathfrak{M} \overset{fum}{\models} \varphi$ iff $\mathfrak{M} \models \mathrm{fum}\varphi$.

(iv) Abbreviations or shorthands: $(\mathrm{Som}\varphi) \overset{d}{=} (\neg\mathrm{Alw}\neg\varphi)$, $(\mathrm{Somfu}\varphi) \overset{d}{=} (\neg\mathrm{Alwfu}\neg\varphi)$, $\mathrm{Sompa}\varphi \overset{d}{=} (\neg\mathrm{Alwpa}\neg\varphi)$, and we use the usual shorthands $\forall x_n$, $\forall y_n$, $\vee$, $\rightarrow$, $\Diamond$, etc. introduced below the definitions of DL_d and DL_d^{mod}.

(v) Axioms. (v)1 Induction axioms:

$$\mathrm{Ifum} \overset{d}{=} \{([\varphi\wedge\mathrm{Alwfu}(\varphi \rightarrow \mathrm{Next}\varphi)] \rightarrow \mathrm{Alwfu}\varphi) : \varphi\in DF_d^{fum}\} \cup \mathrm{Lax}.$$

(v)2 Time-structure axioms:

$$\begin{aligned}\mathrm{Tfum} \overset{d}{=} \{ &\mathrm{First}(\mathrm{Alwfu}\varphi \rightarrow \mathrm{Alw}\varphi),\ \mathrm{First}(\varphi \leftrightarrow \mathrm{Alwpa}\varphi),\\ &(\varphi \rightarrow \mathrm{Sompa}\varphi\wedge\mathrm{Somfu}\varphi),\ ([\mathrm{Alwpa}\varphi\wedge\mathrm{Alwfu}\varphi] \rightarrow \mathrm{Alw}\varphi),\\ &(\mathrm{SomfuSomfu}\varphi \rightarrow \mathrm{Somfu}\varphi),\ (\mathrm{SompaSompa}\varphi \rightarrow \mathrm{Sompa}\varphi),\\ &(\mathrm{Alwfu}\varphi \leftrightarrow [\varphi\wedge\mathrm{NextAlwfu}\varphi]),\ (\mathrm{NextAlwpa}\varphi \leftrightarrow [\mathrm{Next}\varphi\wedge\mathrm{Alwpa}\varphi])\\ &: \varphi\in DF_d^{fum}\}.\end{aligned}$$

(vi) Future enriched modal dynamic language is defined to be $DL_d^{fum} \overset{d}{=} \langle DF_d^{fum}, DM_d^{fum}, \overset{fum}{\models}\rangle$ where $DM_d^{fum} \overset{d}{=} \{\mathfrak{M}\in M_{td} : \mathfrak{M} \overset{fum}{\models} \mathrm{Ifum}\cup\mathrm{Tfum}\}$. We use $\mathrm{Th} \overset{fum}{\models} \varphi$ etc. in accordance with Convention 3, i.e.

$\mathrm{Th} \overset{fum}{\models} \varphi$ iff $(\forall\mathfrak{M}\in DM_d^{fum})[\mathfrak{M} \overset{fum}{\models} \mathrm{Th} \Rightarrow \mathfrak{M} \overset{fum}{\models} \varphi]$. End of Definition 19

Remark: Note that $\varphi \overset{fum}{\models} \mathrm{Alw}\varphi$ for all $\varphi\in DF_d^{fum}$ since $\mathfrak{M} \overset{fum}{\models} \varphi$ implies $\mathfrak{M} \overset{fum}{\models} \mathrm{Alw}\varphi$ by definition. Also note that

$$\begin{aligned}\mathrm{Ifum}\cup\mathrm{Tfum} \overset{fum}{\models} \{&([\mathrm{First}\varphi\wedge\mathrm{Alw}(\varphi \rightarrow \mathrm{Next}\varphi)] \rightarrow \mathrm{Alw}\varphi),\\ &\mathrm{Alw}([\mathrm{First}\varphi\wedge\mathrm{Alwpa}(\varphi \rightarrow \mathrm{Next}\varphi)] \rightarrow \mathrm{NextAlwpa}\varphi) :\\ &: \varphi\in DF_d^{fum}\}.\end{aligned}$$

PROPOSITION 4 (completeness of DL_d^{fum})

Let $\mathrm{Th}\subseteq DF_d^{fum}$ and $\varphi\in DF_d^{fum}$. Then

$\mathrm{Th} \overset{fum}{\models} \varphi$ iff $\{\mathrm{fum}\psi : \psi\in\mathrm{Th}\cup\mathrm{Ifum}\cup\mathrm{Tfum}\} \overset{N}{\vdash} \mathrm{fum}\varphi$.

Proof: By Thm.1 and Def.19. QED

COROLLARY 5 There are decidable proof concepts $\overset{mod}{\vdash}$ and $\overset{fum}{\vdash}$ such that $\langle DL_d^{mod}, \overset{mod}{\vdash}\rangle$ and $\langle DL_d^{fum}, \overset{fum}{\vdash}\rangle$ are complete logics.

<u>DEFINITION 20</u> (Floyd-Hoare logic $\langle HFL_d , (\overset{F}{\vdash} , Prf)\rangle$)

(i) The set HF_d of <u>Floyd-Hoare statements</u> of type d is an important sublanguage of DF_d.:

$HF_d \overset{d}{=} \{ (\varphi \to \Box(p,\psi)) \; : \; p\in P_d$ and $\varphi,\psi \in F_d\}$. Clearly $HF_d \subseteq DF_d$.

(ii) <u>Floyd-Hoare language</u> HFL_d is defined to be:

$HFL_d \overset{d}{=} \langle HF_d \cup F_d , Mod_{td}(Iq) , \vDash\rangle$.

(iii) The relation $\overset{F}{\vdash} \subseteq \{Th : Th \subseteq F_d\} \times HF_d$ was defined in a rigorous manner in [3]Def.17, [4]Def.17,p.55, [6],[2]p.118. We shall use this definition of $\overset{F}{\vdash}$ without reformulating it, but we note that in the quoted papers there is a <u>decidable</u> set $Prf \subseteq (HF_d \cup F_d)^*$ such that $(\forall Th \subseteq F_d)(\forall \varrho \in HF_d)[Th \overset{F}{\vdash} \varrho$ iff $(\exists$ finite $H \subseteq Th)(\exists w)\langle H,w,\varrho\rangle \in Prf]$. Hence Prf is the set of $\overset{F}{\vdash}$-proofs and Prf is decidable. Cf. Def. 10. According to Def.10, $(\overset{F}{\vdash},Prf)$ is a decidable proof concept for the Floyd-Hoare language HFL_d. <u>End of Definition 20</u>

<u>The lattice of proof methods for partial correctness of programs</u>

Instead of "proof method for program verification" we shall simply say "proof method". By a <u>proof method</u> we understand a proof concept $(X \overset{Y}{\vdash})$ in the sense of Def.12 or one in the sense of Def.10. Thus e.g. $\overset{F}{\vdash}$ and $(Dax \overset{N}{\vdash})$ are proof methods. When we call $(X \overset{N}{\vdash})$ a proof method for program verification then what we <u>intuitively</u> have in mind is the proof concept $(X \overset{N}{\vdash})$ as a device for proving properties of programs. We shall concentrate on the powers of proof methods $(X \overset{Y}{\vdash})$ to prove partial correctness of programs.

We define a pre-ordering $\leq$ on the proof methods as follows: $(X \overset{Y}{\vdash}) \leq (Y \overset{Z}{\vdash})$ is defined to hold iff $[(Th \cup X \overset{Y}{\vdash} \varrho) \Rightarrow (Th \cup Y \overset{Z}{\vdash} \varrho)]$ for every similarity type d, $Th \subseteq F_d$ and $\varrho \in HF_d$.

The relation $\leq$ induces an equivalence relation $\equiv$ defined as: $(X \overset{Y}{\vdash}) \equiv (Y \overset{Z}{\vdash})$ iff $[(X \overset{Y}{\vdash}) \leq (Y \overset{Z}{\vdash})$ and $(Y \overset{Z}{\vdash}) \leq (X \overset{Y}{\vdash})]$.

A straight line $X \overset{Y}{\vdash} \diagup Y \overset{Z}{\vdash}$ on Fig.2 indicates the relation $(X \overset{Y}{\vdash}) \leq (Y \overset{Z}{\vdash})$. A line with $\neq$ added like $X \overset{Y}{\vdash} \overset{\neq}{\diagup} Y \overset{Z}{\vdash}$ indicates the strict relation $<$ that is $[(X \overset{Y}{\vdash}) \ngeq (Y \overset{Z}{\vdash})$ and $(X \overset{Y}{\vdash}) \leq (Y \overset{Z}{\vdash})]$. A line with =? added like $X \overset{Y}{\vdash} \overset{=?}{\diagup} Y \overset{Z}{\vdash}$ indicates that $(X \overset{Y}{\vdash}) \leq (Y \overset{Z}{\vdash})$ but we <u>do not know</u> whether $(X \overset{Y}{\vdash}) \geq (Y \overset{Z}{\vdash})$ holds or not. Broken line $X \overset{Y}{\vdash} \overset{\nleq}{\diagdown} Y \overset{Z}{\vdash}$ with $\nleq$ indicates that $(X \overset{Y}{\vdash}) \nleq (Y \overset{Z}{\vdash})$ (but we do not know

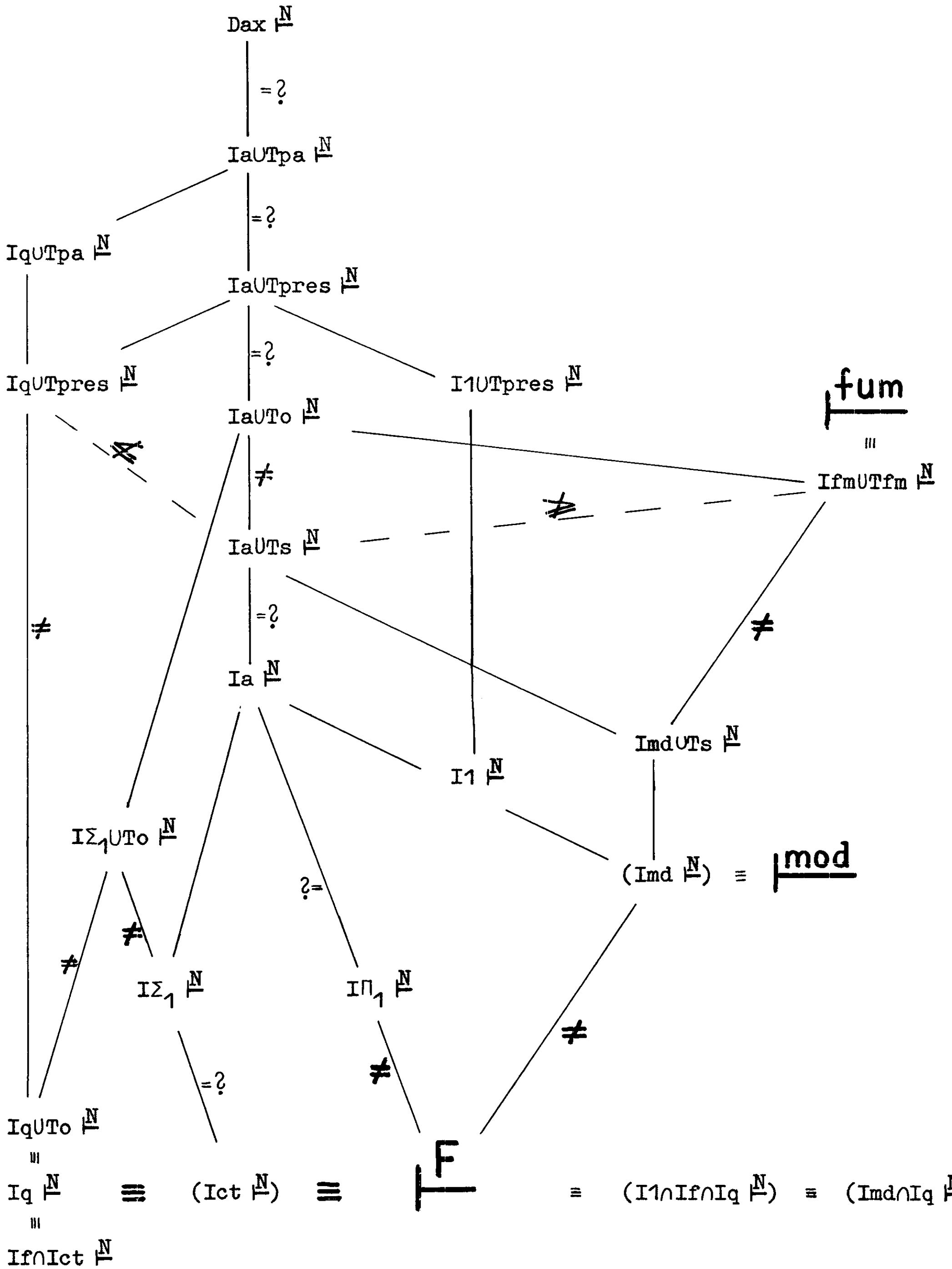

Dax ⊨N
=?
IaUTpa ⊨N
IqUTpa ⊨N
=?
IaUTpres ⊨N
IqUTpres ⊨N
=?
I1UTpres ⊨N
IaUTo ⊨N
⊢fum
≡
IfmUTfm ⊨N
≠
IaUTs ⊨N
≠
=?
Ia ⊨N
ImdUTs ⊨N
I1 ⊨N
IΣ1UTo ⊨N
(Imd ⊨N) ≡ ⊢mod
?=
≠
IΣ1 ⊨N
IΠ1 ⊨N
=?
IqUTo ⊨N
≡
Iq ⊨N ≡ (Ict ⊨N) ≡ ⊢F ≡ (I1∩If∩Iq ⊨N) ≡ (Imd∩Iq ⊨
≡
If∩Ict ⊨N

FIGURE 2

whether $(X \overset{Y}{\vdash}) \geq (Y \overset{Z}{\vdash})$ holds or not). If $(X \overset{Y}{\vdash}) \nleq (Y \overset{Z}{\vdash})$ is not indicated (either by $\neq$ or by $\nleq$) then we do not know whether or not $(X \overset{Y}{\vdash}) \leq$ $\leq (Y \overset{Z}{\vdash})$. Hence "=?" is used only to stress that we do not know whether equivalence holds. 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 $(Iq \cup TPres \overset{N}{\vdash}) \leq (Ia \cup To \overset{N}{\vdash})$ holds or not. Note that the fact $Iq \not\vdash Iq \cup To$ does <u>not</u> imply $(Iq \overset{N}{\vdash}) \nleq (Iq \cup To \overset{N}{\vdash})$ since proof methods here are compared only w.r.t. $Th \subseteq F_d$ and $\rho \in HF_d$.

§5. Proofs and discussions of Figures 2,1

We shall prove that the inclusions $(X \overset{Y}{\vdash}) \leq (Y \overset{Z}{\vdash})$ as well as the inequalities $(X \overset{Y}{\vdash}) \nleq (Y \overset{Z}{\vdash})$ indicated on Fig.2 all do hold. First, in Thm.6 below, we prove one inequality $(Ia \cup To \overset{N}{\vdash}) \nleq (Ia \cup Ts \overset{N}{\vdash})$ and then after proving Thm.6 we shall proving the rest of Fig.2.

Thm.6 below is in contrast with the result $(Iq \cup To \overset{N}{\vdash}) \equiv (Iq \cup Ts \overset{N}{\vdash})$ indicated on Fig.2.

<u>THEOREM 6</u> There are a finite d and $\Box(p,\psi) \in HF_d$ such that $Ia \cup To \overset{N}{\vdash} \Box(p,\psi)$ but $Ia \cup Ts \overset{N}{\not\vdash} \Box(p,\psi)$.

<u>Proof</u>. Let $d \overset{d}{=} \langle \{su, zero\}, \{\langle su,2\rangle, \langle zero,1\rangle, \langle R,1\rangle, \langle S,1\rangle\}\rangle$, i.e. d is a similarity type which has a unary function symbol su, a constant symbol zero and two relation symbols R and S.

Let $0' \overset{d}{=}$ zero and $(\forall n \in \omega)(n+1)' \overset{d}{=} su(n')$. Let $Lab \overset{d}{=} \{n' : n \in \omega\}$.

Let $p \in P_d$ be the program represented on Fig.3. Note that in defining p we use fewer labels than required in the formal definition of P_d, but it is easy to see that this change is <u>not essential</u> while it considerably <u>simplifies</u> the traces of p.

Let $\psi(x_0,x_1) \in F_d$ be the formula $(\neg S(x_0) \rightarrow x_0 = x_1)$.

We shall show that $Ia \cup Ts \not\vdash \Box(p,\psi)$ while $Ia \cup To \overset{N}{\vdash} \Box(p,\psi)$. To this end, first we construct a model $\mathfrak{M} \in M_{td}$.

<u>6.0. The definition of $\mathfrak{M} \in M_{td}$.:</u>

Z denotes the set of all integers such that $\omega \subseteq Z$ is the set of nonnegative members of Z.

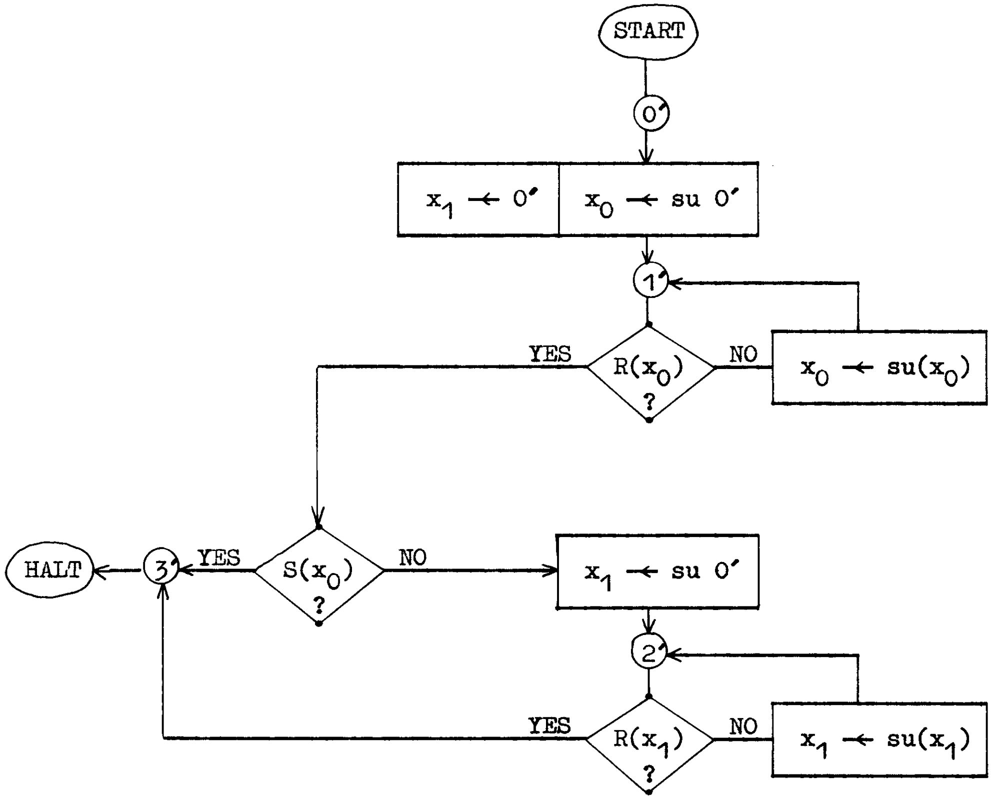


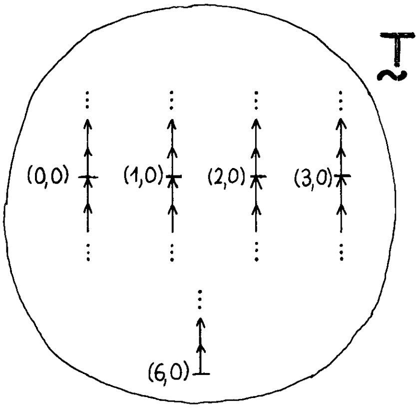
FIGURE 3

Let $A \stackrel{d}{=} (6\times Z) \cup (\{6,7\}\times\omega)$. We often write (i,n) instead of $\langle i,n\rangle$. Note that if $a\in A$ then $a=(i,n)$ for some $i\in 8$ and $n\in Z$. Let $suc : A \to A$ be defined by $suc(i,n) \stackrel{d}{=} (i,n+1)$ for every $(i,n)\in A$.

6.0.1. Let $\underset{\sim}{T}\in M_t$ be the following model of type t. (See Fig.4.)

$\underset{\sim}{T} \stackrel{d}{=} \langle T,Q\rangle$ where $T = (\{6\}\times\omega)\cup(4\times Z)$ and $Q(0) = (6,0) \stackrel{d}{=} 0^T$, $Q(sc) = suc$, $Q(\leq) = 0$ and $Q(+) = Q(\cdot) = T\times T\times\{0^T\}$. See Def.s 1 and 2.

We shall sloppily identify $\underset{\sim}{T}$ with the structure $\langle T,suc,0^T\rangle$. At two places above we should have written $(T\times T)\cap suc$ instead of suc but we hope that context helps to understand that we meant e.g. $Q(sc) \stackrel{d}{=} (T\times T)\cap suc$. We shall commit this kind of sloppiness in the future too.



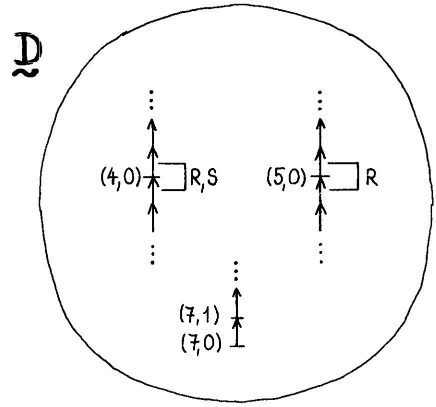


FIGURE 4

FIGURE 5

6.0.2. Let $\underset{\sim}{D}\in M_d$ be the following model of type d. (See Fig.5.)
$\underset{\sim}{D} \overset{d}{=} \langle D,G\rangle$ where $D = (\{7\}\times\omega)\cup(\{4,5\}\times Z)$ and $G(\text{zero}) = (7,0)$, $G(\text{su}) = \text{suc}$, $G(R) = \{(4,0),(5,0)\}$, $G(S) = \{(4,0)\}$.

Notation: Let $n\in\omega$. We shall identify n' with $(7,n)$ since $(7,n)$ is the value of the term n' in $\underset{\sim}{D}$.

6.0.3. Next we define three functions $f,h,g : T \to D$ illustrated on Fig.s 6–8.

$$f \overset{d}{=} \{\langle(6,n),n'\rangle,\langle(0,-n),(4,-n)\rangle,\langle(0,n),(4,0)\rangle,\langle(1,-n),(5,-n)\rangle, \langle(1,n),(5,0)\rangle,\langle(i,z),(5,0)\rangle \ : \ n\in\omega,\ i\in\{2,3\},\ z\in Z\}.$$

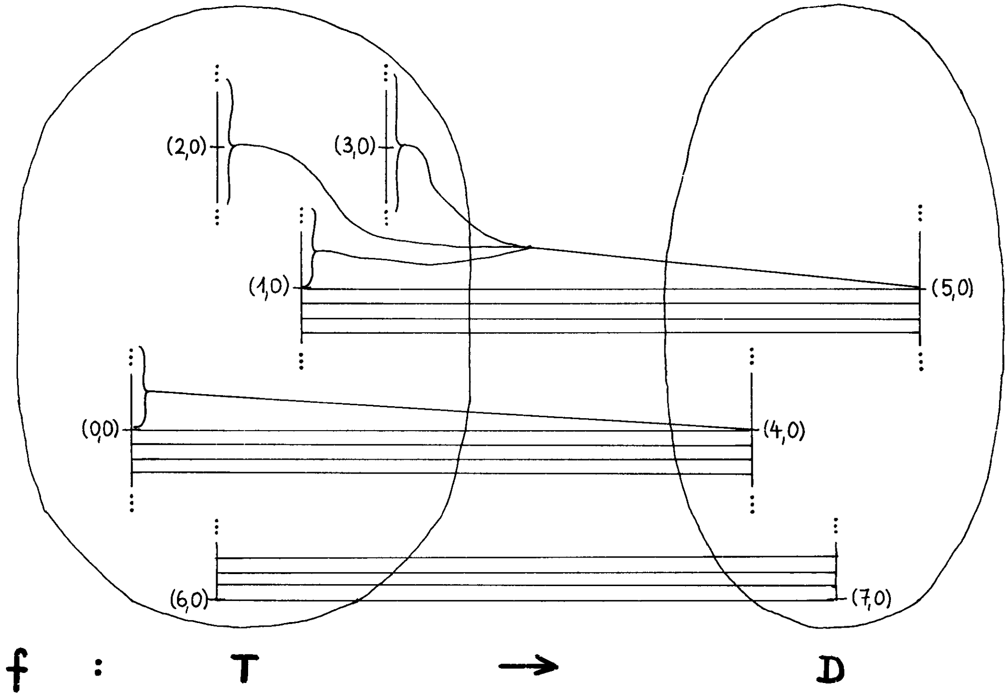
$$h \overset{d}{=} \{\langle(6,n),0'\rangle,\langle(0,z),0'\rangle,\langle(1,-n),0'\rangle,\langle(1,n),n'\rangle,\langle(2,-n),(4,-n)\rangle, \langle(2,n),(4,0)\rangle,\langle(3,-n),(5,-n)\rangle,\langle(3,n),(5,0)\rangle \ : \ n\in\omega,\ z\in Z\}.$$

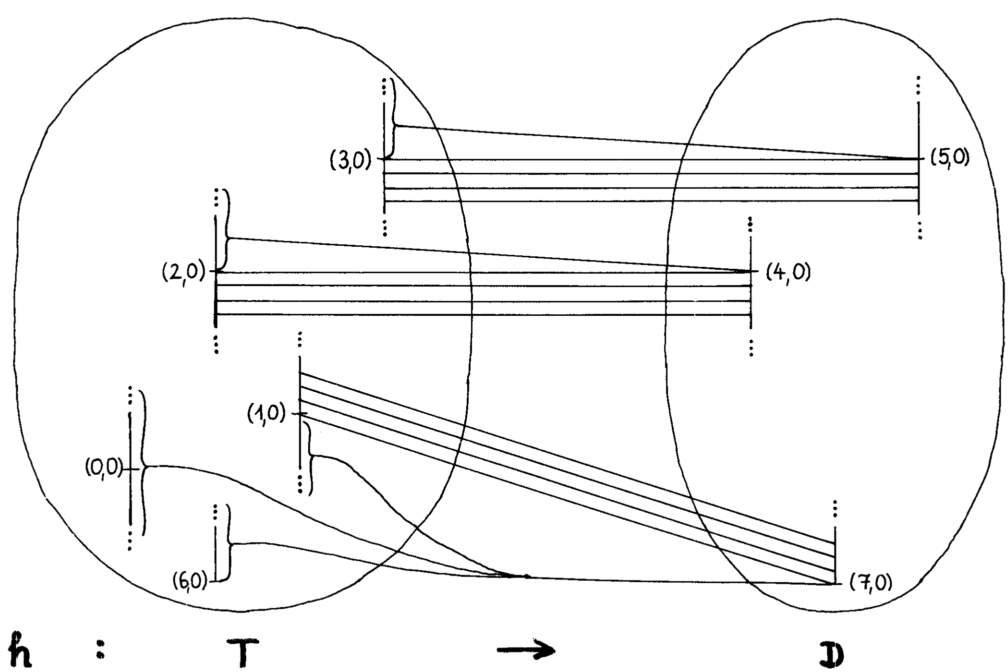
$$g \overset{d}{=} \{\langle(6,0),0'\rangle,\langle(6,n+1),1'\rangle,\langle(0,-n),1'\rangle,\langle(0,n+1),3'\rangle,\langle(1,-n),1'\rangle, \langle(1,n+1),2'\rangle,\langle(2,-n),2'\rangle,\langle(2,n+1),3'\rangle,\langle(3,-n),2'\rangle,\langle(3,n+1,3'\rangle \ : \ : \ n\in\omega\}.$$

6.0.4. Let $I \overset{d}{=} \{f,h,g\}$, $\text{valueof} \overset{d}{=} \langle k(a) : \langle k,a\rangle\in I\times T\rangle$ and $\mathfrak{M} \overset{d}{=} \langle \underset{\sim}{T}, \underset{\sim}{D}, I, \text{valueof}\rangle$. We have defined the model $\mathfrak{M}\in M_{td}$.

CLAIM 6.1. $\mathfrak{M} \models \text{Ia}\cup\text{Ts}$.

Proof. Clearly, $\mathfrak{M} \models \text{Ts}\cup\text{Lax}$. To prove $\mathfrak{M}\models \text{IA}$ we shall use an ultraproduct construction. Let F be a nonprincipal ultrafilter on ω and let $\mathfrak{M}^+ \overset{d}{=} \langle \underset{\sim}{T}^+,\underset{\sim}{D}^+,I^+,\text{ext}\rangle \overset{d}{=} {}^{\omega}\mathfrak{M}/F$ be the usual ultrapower of $\mathfrak{M}$. Let $\delta : \mathfrak{M} \to \mathfrak{M}^+$ be the usual diagonal embedding. For every $i\in\omega$





FIGURES 6-7

let $(i\uparrow) \stackrel{d}{=} \langle (i,n) : n\in\omega\rangle/F$ and $(i\downarrow) \stackrel{d}{=} \langle (i,-n) : n\in\omega\rangle/F$. Let $M^+ \stackrel{d}{=} T^+\cup D^+\cup I^+$. Hence M^+ is the universe of $\mathfrak{M}^+$, more precisely M^+ is the disjoint union of all the universes of $\mathfrak{M}^+$.

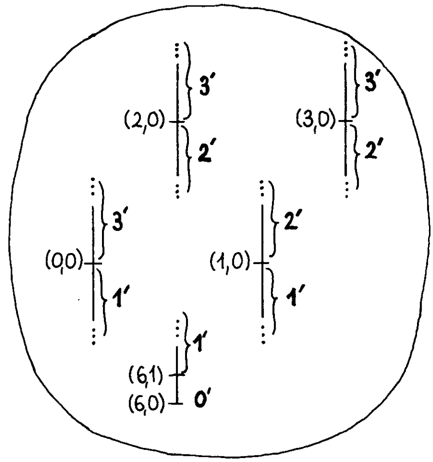


FIGURE 8

Notations: $\mathrm{Id} \stackrel{d}{=} \langle m : m\in M^+\rangle$. Then $\mathrm{Id} : M^+ \rightarrowtail\!\!\!\twoheadrightarrow M^+$ is the identity mapping. For any sets X,Y and functions k,q we define:

$X\sim Y \stackrel{d}{=} \{a\in X : a\notin Y\}$,

$X\upharpoonright k \stackrel{d}{=} (X\times \mathrm{Rng}\ k)\cap k$ and

$k\circ q \stackrel{d}{=} \langle k(q(x)) : x\in \mathrm{Dom}(q) \text{ and } q(x)\in \mathrm{Dom}(k)\rangle$. That is, $X\upharpoonright k$ is the function k domain-restricted to the set X, $k\circ q$ is the composition of k and q. Then $X\upharpoonright \mathrm{Id}\subseteq k$ means that k is identity on X.

CLAIM 6.2. There are automorphisms $P : \mathfrak{M}^+ \rightarrowtail\!\!\!\twoheadrightarrow \mathfrak{M}^+$ and $Q : \mathfrak{M}^+ \rightarrowtail\!\!\!\twoheadrightarrow \mathfrak{M}^+$ of $\mathfrak{M}^+$ such that $P\circ\delta = Q\circ\delta = \delta$, $P(6\uparrow)=(1\downarrow)$, $P(1\uparrow)=(3\downarrow)$ and $Q(6\uparrow)=(0\downarrow)$, $Q(1\uparrow)=(2\downarrow)$.

Proof of Claim 6.2.: Let $B \stackrel{d}{=} T^+\cup D^+$. Then $\delta : A \rightarrowtail B$. Let $suc : B\to B$ be the natural one, i.e. $\langle B,suc\rangle \stackrel{d}{=} {}^{\omega}\langle A,suc\rangle/F$. Let $(\forall n\in\omega)(\forall b\in B)\ suc^0(b)\stackrel{d}{=}b$ and $suc^{n+1}(b)\stackrel{d}{=}suc\,suc^n(b)$. We define $(\forall b\in B)$ $L(b) \stackrel{d}{=} \{suc^n(b) : n\in\omega\} \cup \{a\in B : (\exists n\in\omega)suc^n(a)=b\}$. Let $H_6 \stackrel{d}{=} L(6\uparrow)\cup L(7\uparrow)\cup L(1\uparrow)$, $H_1 \stackrel{d}{=} L(1\downarrow)\cup L(5\downarrow)\cup L(3\downarrow)$ and $H_0 \stackrel{d}{=} L(0\downarrow)\cup L(4\downarrow)\cup L(2\downarrow)$. See Fig.9! Clearly,

(*) there is an isomorphism $p : \langle H_6,suc\rangle \rightarrowtail\!\!\!\twoheadrightarrow \langle H_1,suc\rangle$ such that $p(6\uparrow)=(1\downarrow)$, $p(7\uparrow)=(5\downarrow)$ and $p(1\uparrow)=(3\downarrow)$.

Let $P \stackrel{d}{=} p\cup p^{-1}\cup (M^+\sim(H_6\cup H_1))\upharpoonright \mathrm{Id}$, where $p^{-1} \stackrel{d}{=} \{\langle b,a\rangle : \langle a,b\rangle\in p\}$ is the usual inverse of p. We show that P is an automorphism of $\mathfrak{M}^+$. For illustration of the proof see Fig.9.

Below we shall omit some straightforward details, but we shall be glad to send [20], which contains all the details of the present proof

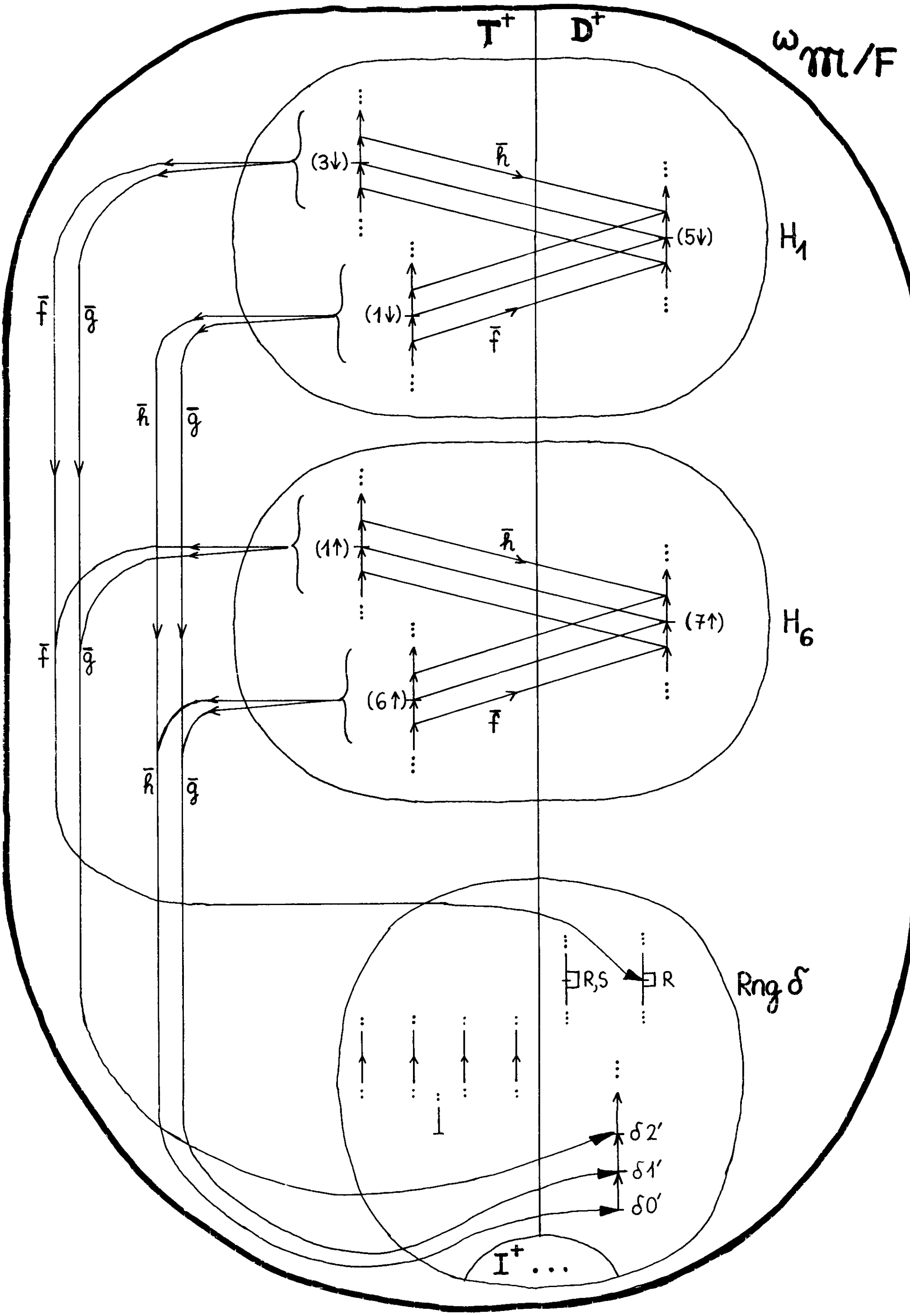

T^+
D^+
$\omega\mathfrak{M}/F$
H_1
$(3\downarrow)$
$(5\downarrow)$
$(1\downarrow)$
$\bar{h}$
$\bar{f}$
$\bar{g}$
H_6
$(1\uparrow)$
$(7\uparrow)$
$(6\uparrow)$
Rng δ
R,S
R
$\delta 2'$
$\delta 1'$
$\delta 0'$
$I^+ \ldots$

FIGURE 9

to anybody asking for it. It is easy to check the following

(⁎⁎) H_6 , H_1 , H_0 and $\mathrm{Rng}\delta$ are pairwise disjoint.

By (⁎⁎), P is a function on M^+ and $\mathrm{Rng}\delta \restriction P \subseteq \mathrm{Id}$. By (⁎), it is easy to check that $P : T^+ \rightarrowtail\!\!\!\!\twoheadrightarrow T$ and $P : D^+ \rightarrowtail\!\!\!\!\twoheadrightarrow D^+$ i.e. P is a permutation both of T^+ and D^+. Since I, $R^{\mathfrak{M}}$ and $S^{\mathfrak{M}}$ are finite, we have $I^+ \cup R^{\mathfrak{M}^+} \cup S^{\mathfrak{M}^+} \subseteq \mathrm{Rng}\,\delta$ (see Convention 2). Thus $P : I^+ \rightarrowtail\!\!\!\!\twoheadrightarrow I^+$ is a permutation of I^+ and P preserves R, S and the constants 0 and zero (since $\{0^{\mathfrak{M}^+}, \mathrm{zero}^{\mathfrak{M}^+}\} \subseteq \mathrm{Rng}\,\delta$). P preserves sc and su by (⁎) since $B \sim H_i$ is closed under suc (and clearly P preserves $+, \cdot$ and $\leq$ by their definitions). All what remains to show is that P preserves the binary function ext.

The only <u>really binary</u> operation of $\mathfrak{M}^+$ is ext : $I^+ \times T^+ \to D^+$. But by $I^+ \subseteq \mathrm{Rng}\delta$ we have $I^+ \restriction P \subseteq \mathrm{Id}$, hence the first arguments of ext are fixed points of P. Hence from the point of view of P, ext behaves like three unary functions. More precisely, let $(\forall k \in I)\ \bar{k} \overset{d}{=}$ $\overset{d}{=} \langle \mathrm{ext}(\delta k, a) : a \in T^+ \rangle$. Note that $I^+ = \{\delta f, \delta g, \delta h\}$. Then to see that P preserves ext it <u>is</u> enough to check that $(\forall k \in I)[P$ preserves $\bar{k}]$. Thus we reduced $\mathfrak{M}^+$ to a unary model $\mathfrak{M}' = \langle M^+, \bar{f}, \bar{g}, \bar{h} \rangle$ and we have to show that P is an automorphism of $\mathfrak{M}'$. Now we are going to show that P preserves $\bar{f}$, $\bar{g}$, and $\bar{h}$.

$\mathrm{Su}\,\mathfrak{M}'$ denotes the set of all subuniverses of $\mathfrak{M}'$, i.e. subsets of M^+ closed under $\bar{f}$, $\bar{g}$, and $\bar{h}$.

Let $N_i \overset{d}{=} H_i \cup \mathrm{Rng}\,\delta$, for every $i \in \{6,1,0\}$. Now we claim statements $(⁎^3)$-$(⁎^5)$ below for every $i \in \{6,1,0\}$:

$(⁎^3)$ $N_i \in \mathrm{Su}\,\mathfrak{M}'$.

$(⁎^4)$ $P : \langle N_6, \bar{f}, \bar{g}, \bar{h} \rangle \rightarrowtail\!\!\!\!\twoheadrightarrow \langle N_1, \bar{f}, \bar{g}, \bar{h} \rangle$ is an isomorphism.

$(⁎^5)$ $(M^+ \sim H_i) \in \mathrm{Su}\,\mathfrak{M}'$.

To check $(⁎^3)$-$(⁎^5)$ above, we use Los lemma and the definitions of f, g,h , see Fig.9. The detailed proof is in [20]. We omit this proof because it is straightforward. By (⁎⁎) we have that P is identity on $N_6 \oplus N_1 \overset{d}{=} (N_6 \cap N_1) \cup (M^+ \sim (N_6 \cup N_1))$ i.e. $(N_6 \oplus N_1) \restriction P \subseteq \mathrm{Id}$. By $(⁎^5)$, $(N_6 \oplus N_1) \in \mathrm{Su}\,\mathfrak{M}'$. These facts together with $(⁎^3)$-$(⁎^5)$ imply that $P : \mathfrak{M}' \rightarrowtail\!\!\!\!\twoheadrightarrow \mathfrak{M}'$ is an automorphism.

So far, we have seen that $P : \mathfrak{M}^+ \rightarrowtail\!\!\!\!\twoheadrightarrow \mathfrak{M}^+$ is an automorphism.

Clearly P satisfies the conditions of Claim 6.2. The construction of Q is obtained from the above proof by substituting Q, H_0, N_0, $(2\downarrow)$, $(4\downarrow)$ and $(0\downarrow)$ into the places of P, H_1, N_1, $(3\downarrow)$, $(5\downarrow)$ and $(1\downarrow)$ respectively, everywhere. <u>QED(Claim 6.2.)</u>

We turn to the proof of $\mathfrak{M} \models IA$. Let $\varphi(z_0)\in F_{td}$ be any formula possibly with parameters from M. More precisely, let $m\in\omega$, $p\in {}^mM$ and let $\varphi(z_0)$ be the formula $\varphi(z_0,p)$ that is $\varphi(z_0,p_0,\dots,p_{m-1})$. We assume that $\varphi(z_0,p)$ is obtained from some $\varphi(z_0,\bar{z},\bar{x},\bar{y})\in F_{td}$ by substituting p in place of $\langle\bar{z},\bar{x},\bar{y}\rangle$ such that everything belongs to the appropriate sort, e.g. if p_0 is substituted for z_1 then $p_0\in T$. Assume that $\varphi(z_0,p)$ has no free variable other than z_0. Let $b\in T$ be arbitrary. Then $\mathfrak{M} \models \forall z_0\varphi(z_0,p)$ and $\mathfrak{M} \models \varphi(b,p)$ have their obvious meanings, see e.g. Def.1.3.14-15 of [8]p.28 where $\varphi(b,p)$ and $\forall z_0\varphi(z_0,p)$ are denoted by $\varphi[b,p]$ and $(\forall z_0\varphi)[p]$ respectively.

We want to prove $\mathfrak{M} \models \mathrm{ind}(\varphi,z_0)$. <u>Assume</u>

(C1) $\mathfrak{M} \models \varphi(0,p)$ and $\mathfrak{M} \models \forall z_0(\varphi(z_0,p) \to \varphi(sc(z_0),p))$.

Then $(\forall n\in\omega)\,\mathfrak{M}\models\varphi(\langle 6,n\rangle,p)$ since $\langle 6,n\rangle=sc^n(0)$ in $\mathfrak{M}$. Then

(C2) $\mathfrak{M}^+ \models \varphi((6\uparrow),\delta\circ p)$ holds by Los lemma.

Let P,Q be the automorphisms the existence of which is claimed in 6.2. Since P is an automorphism, by (C2) we have $\mathfrak{M}^+ \models \varphi(P(6\uparrow),P\circ\delta\circ p)$, hence $\mathfrak{M}^+ \models \varphi((1\downarrow),\delta\circ p)$ by $P(6\uparrow)=(1\downarrow)$ and by $P\circ\delta=\delta$. By the Los lemma there is $V\in F$ such that $(\forall n\in V)\,\mathfrak{M}\models\varphi(\langle 1,-n\rangle,p)$. Since F is nonprincipal, V is infinite which implies by (C1) that $(\forall z\in Z)$ $\mathfrak{M} \models \varphi(\langle 1,z\rangle,p)$. Then $\mathfrak{M}^+ \models \varphi((1\uparrow),\delta\circ p)$. Using Claim 6.2, $P(1\uparrow)=(3\downarrow)$, $Q(1\uparrow)=(2\downarrow)$, Los lemma and (C1) as above we obtain $(\forall z\in Z)[\mathfrak{M}\models\varphi(\langle 3,z\rangle,p)$ and $\mathfrak{M}\models\varphi(\langle 2,z\rangle,p)]$. By (C2) and $Q(6\uparrow)=(0\downarrow)$ we have $\mathfrak{M}^+ \models \varphi((0\downarrow),\delta\circ p)$. Then as above, by (C1) we conclude $(\forall z\in Z)\,\mathfrak{M}\models\varphi(\langle 0,z\rangle,p)$. We have proved $(\forall b\in T)\,\mathfrak{M}\models\varphi(b,p)$ which means $\mathfrak{M} \models \forall z_0\varphi(z_0,p)$. Thus $\mathfrak{M} \models \mathrm{ind}(\varphi(z_0,p),z_0)$. Since the choice of p was arbitrary, this means $\mathfrak{M} \models \forall\bar{z}\forall\bar{x}\forall\bar{y}\,\mathrm{ind}(\varphi(z_0,\bar{z},\bar{x},\bar{y}),z_0)$. Since $\varphi\in F_{td}$ was chosen arbitrarily, we proved $\mathfrak{M} \models IA$. <u>QED(Claim 6.1.)</u>

<u>CLAIM 6.3.</u> $\mathfrak{M} \not\models \Box(p,\psi)$.

<u>Proof</u>. Let $s \stackrel{d}{=} \langle f,h,g\rangle$. Then s is a trace of p in $\mathfrak{M}$. To see this fact observe that $g=s_2$ is the history of the control variable of p, see Fig.s 6-8. Let $b \stackrel{d}{=} \langle 2,0\rangle$. Then s terminates p in $\mathfrak{M}$

at time b since $s_2(b) = g(b) = 3'$ is the label of the HALT command of p. The output $\langle s_0(b), s_1(b)\rangle$ of p at time b does not satisfy ψ in $\mathfrak{M}$ since $\neg S(\langle 5,0\rangle)$ and $s_0(b) = f(b) = (5,0) \neq (4,0) = h(b) = s_1(b)$. Thus $\langle (5,0),(4,0)\rangle$ is a possible output of p in $\mathfrak{M}$ but $\underset{\sim}{D} \not\models \psi(x_0,x_1)[(5,0),(4,0)]$. <u>QED(Claim 6.3.)</u>

By Thm.1, 6.1 and 6.3 above we have the following

<u>COROLLARY 6.4.</u> $Ia \cup Ts \overset{N}{\not\models} \Box(p,\psi)$.

<u>CLAIM 6.5.</u> $Ia \cup To \overset{N}{\models} \Box(p,\psi)$.

<u>Proof.</u> Let $Ax \overset{d}{=} Ia \cup To$. Let "$(\forall z_1 < z_0)\varphi$" stand for the formula $\forall z_1[(z_1 \leq z_0 \wedge z_1 \neq z_0) \rightarrow \varphi]$. Similarly for "$(\forall z_1 \geq z_0)\varphi$" etc. For every $\varphi(z_0) \in F_{td}$ we define $\text{first}(\varphi, z_0)$ to be the formula $[(\forall z_1 < z_0)\neg\varphi(z_1) \wedge \varphi(z_0)]$.

<u>CLAIM 6.6.</u> Let $\varphi \in F_{td}$. Then $Ax \models (\exists z_0 \varphi(z_0) \rightarrow \exists z_0 \text{first}(\varphi, z_0))$.

<u>Proof.</u> Let $\psi(z_2)$ be the formula $[(\exists z_0 \leq z_2)\varphi(z_0) \rightarrow (\exists z_0 \leq z_2)\text{first}(\varphi,z_0)]$. Then $To \models \psi(0) \wedge \forall z_2[\psi(z_2) \rightarrow \psi(sc(z_2))]$ is easy to prove. By $\text{ind}(\psi(z_2),z_2) \in Ia$ we conclude $Ax \models \forall z_2 \psi(z_2)$. Then obviously $Ax \models [\exists z_0 \varphi(z_0) \rightarrow \exists z_0 \text{first}(\varphi, z_0)]$. <u>QED(Claim 6.6.)</u>

For any $\varphi(z_0) \in F_{td}$ let $\text{hyp}(\varphi, z_2)$ be the formula $(\varphi(z_2) \wedge (\forall z_0 \geq z_2)[\varphi(z_0) \rightarrow \varphi(sc(z_0))]$.

<u>CLAIM 6.7.</u> Let $\varphi(z_0) \in F_{td}$. Then $Ax \models \forall z_2[\text{hyp}(\varphi, z_2) \rightarrow (\forall z_0 \geq z_2)\varphi(z_0)]$.

<u>Proof.</u> $To \models [\text{hyp}(\varphi,z_2) \rightarrow \neg\exists z_0 \text{first}([\neg\varphi(z_0) \wedge z_0 \geq z_2], z_0)$. By 6.6. then $Ax \models (\text{hyp}(\varphi, z_2) \rightarrow \neg\exists z_0[\neg\varphi(z_0) \wedge z_0 \geq z_2])$. <u>QED(Claim 6.7.)</u>

Let $\mathfrak{M} = \langle \underset{\sim}{T}, \underset{\sim}{D}, I, ext\rangle \in Mod_{td}(Ax)$ be arbitrary. Let $s \in {}^3I$ be an arbitrary trace of p in $\mathfrak{M}$.

<u>Notations</u>: Throughout, instead of the term $ext(s_i, z_j)$ we shall write $s_i(z_j)$. Let $b \in T$. Then $\bar{s}(b) \overset{d}{=} \langle s_i(b) : i \in 3\rangle$ and $\bar{\bar{s}}(b) \overset{d}{=} \langle s_0(b), s_1(b)\rangle$.

<u>CLAIM 6.8.</u> (i) $\mathfrak{M} \models [s_2(z_0) \in \{2', 3'\} \rightarrow (\forall z_1 \geq z_0) s_0(z_1) = s_0(z_0)]$.
(ii) $\mathfrak{M} \models (s_2(z_0) = 2' \rightarrow (\exists z_1)[z_1 \leq z_0 \wedge s_2(z_1) = 1' \wedge s_0(z_1) = s_1(z_0)])$.

Proof. Proof of (i): Let $b \in T$ be such that $s_2(b) \in \{2', 3'\}$. Let $\gamma(z_1)$ be the formula $[s_2(z_1) \in \{2', 3'\} \wedge s_0(z_1) = s_0(b)]$. Clearly, $\mathfrak{M} \models \gamma(b)$. Also $\mathfrak{M} \models \gamma(z_1) \rightarrow \gamma(sc(z_1))$ because s is a trace of p. Hence $\mathfrak{M} \models (\forall z_1 \geq b)\gamma(z_1)$, by 6.7. Thus $\mathfrak{M} \models (\forall z_1 \geq b) s_0(z_1) = s_0(b)$.

Proof of (ii): Let $\mathcal{K}(z_0, z_1)$ be the formula $[z_1 \leq z_0 \wedge s_2(z_1) = 1' \wedge \wedge s_0(z_1) = s_1(z_0)]$ and let $\varphi(z_0)$ be the formula $[s_2(z_0) = 2' \rightarrow \rightarrow \exists z_1 \mathcal{K}(z_0, z_1)]$. We have to prove $\mathfrak{M} \models \forall z_0 \varphi(z_0)$.

Let $b \in T$. Assume $\mathfrak{M} \models \varphi(b)$. If $s_2(sc(b)) \neq 2'$ then $\varphi(sc(b))$ is obviously true. Assume therefore $s_2(sc(b)) = 2'$.

Case 1 $s_2(b) \neq 2'$. Then, since s is a trace, $s_2(b) = 1'$. Then $s_2(sc(b)) = 2'$ implies $\mathcal{K}(sc(b), sc(0))$. I.e. $\mathfrak{M} \models \varphi(sc(b))$ holds.

Case 2 $s_2(b) = 2'$. Then by $\varphi(b)$, there exists $a \in T$ with $\mathcal{K}(b, a)$. Since s is a trace of p and $s_2(b) = s_2(sc(b)) = 2'$ we have $\neg R(s_1(b))$. Hence by $\mathcal{K}(b, a)$ we have $s_2(sc(a)) = 1'$ and $s_0(sc(a)) = su(s_0(a)) = su(s_1(b)) = s_1(sc(b))$. We have $sc(a) \leq sc(b)$ since $a \leq b$ by $\mathcal{K}(b, a)$. Thus $\mathcal{K}(sc(b), sc(a))$ proving $\mathfrak{M} \models \varphi(sc(b))$.

We proved $\mathfrak{M} \models \forall z_0(\varphi(z_0) \rightarrow \varphi(sc(z_0)))$. Since $\varphi(0)$ is obviously true, by IA we proved $\mathfrak{M} \models \forall z_0 \varphi(z_0)$. QED(Claim 6.8.)

Now we turn to the proof of $\mathfrak{M} \models \Box(p, \psi)$. Let $\langle a, d \rangle \in {}^2D$ be any possible output of p in $\mathfrak{M}$. Then there are a trace $s \in {}^3I$ of p and $e \in T$ such that $\bar{s}(e) = \langle a, d, 3' \rangle$. If $\underset{\sim}{D} \models S(a)$ then $\underset{\sim}{D} \models \psi[a, d]$ is obvious. Assume therefore $\underset{\sim}{D} \models \neg S(a)$. By 6.6. there is $c \in T$ such that that $\text{first}(\bar{s}(sc(c)) = \bar{s}(e), c)$ holds (since $e \neq 0$). Let this c be fixed. Then $\bar{s}(c) \neq \bar{s}(sc(c))$, hence $s_2(c) \neq 3'$. Since s is a trace of p, by $s_2(sc(c)) = 3'$ we have $\bar{\bar{s}}(c) = \bar{\bar{s}}(sc(c)) = \langle a, d \rangle$. Then $\neg S(a)$ implies $s_2(c) \neq 1'$ proving $s_2(c) = 2'$. By $s_2(sc(c)) = 3'$ then we have $R(d)$. By 6.8(ii) we have $(\exists b < c)(\exists x \in D)\ \bar{s}(b) = \langle d, x, 1' \rangle$. By $R(d)$ we have $s_2(sc(b)) \in \{2', 3'\}$ and $s_0(sc(b)) = s_0(b)$. Then by 6.8(i) and $sc(b) \leq c$ we have $d = s_0(b) = s_0(sc(b)) = s_0(c) = a$. We proved $\underset{\sim}{D} \models \psi[a, d]$. By the choices of e, s, and $\mathfrak{M}$ we proved $Ia \cup To \models \Box(p, \psi)$. Then by Thm.1 we have $Ia \cup To \overset{N}{\models} \Box(p, \psi)$. QED(Theorem 6.)

PROOF OF THE REST OF FIGURE 2 :

1) Proofs of the inequalities (all these proofs use ultraproducts):

(1.1) Sketchy proofs of $(Iq \cup Tpres \overset{N}{\models}) \nleq (Iq \cup To \overset{N}{\models})$ and $(Iq \cup Tpres \overset{N}{\models}) \nleq \overset{F}{\models}$ are Thm.9(iv $\not\Rightarrow$ i) in Part II of [3] and [4]p.93

together with pp.60-65 Claim 9.1 there. Detailed proof is available from the author.

(1.2) $(\mathrm{Iq} \cup \mathrm{Tpres} \overset{N}{\models}) \nleq (\mathrm{Ia} \cup \mathrm{Ts} \overset{N}{\models})$ is proved in [20]. The proof is a modification of the above proof of Thm.6: it uses Corollary 6.4 unchanged and the only part that is changed is formulation and proof of Claim 6.5. See also the $\mathrm{Iq} \cup \mathrm{Tpres} \models \Box(p,\psi)$ part of proof of (1.1) above.

(1.3) $(\mathrm{Ifm} \cup \mathrm{Tfm} \overset{N}{\models}) \nleq (\mathrm{Ia} \cup \mathrm{Ts} \overset{N}{\models})$ is proved in [20]. The proof is a modification of the above proof of Thm.6; it uses Corollary 6.4 unchanged.

(1.4) $(\mathrm{Imd} \overset{N}{\models}) \nleq \overset{F}{\models}$ is proved in detail in Thm.9(v $\not\Rightarrow$ i) of [4]pp. 59-93, see also Thm.s 11/e - 11/g of [4]pp.100-107, and [23].

(1.5) The proof of $(\mathrm{Ia} \cup \mathrm{To} \overset{N}{\models}) \nleq \overset{F}{\models}$ is very easy! See Thm.10 in [3]Part II. In the proof of Thm.9(v $\not\Rightarrow$ i) in [3] a partial correctness statement $\rho \in HF_d$ and a finite $Th \subseteq F_d$ are selected and an easy ultraproduct proof is outlined to show $Th \overset{F}{\not\models} \rho$. It is very easy to show $Th \cup \mathrm{Ia} \cup \mathrm{To} \overset{N}{\models} \rho$ by using the proof methods of Thm.s 3-4 in [3] for that Th and ρ.

(1.6) $(\mathrm{Ia} \cup \mathrm{To} \overset{N}{\models}) \nleq (\mathrm{Ia} \cup \mathrm{Ts} \overset{N}{\models})$ is Thm.6 proved above in the present paper.

(1.7) $(\mathrm{I}\Pi_1 \overset{N}{\models}) \nleq \overset{F}{\models}$ and $(\mathrm{I}\Sigma_1 \cup \mathrm{To} \overset{N}{\models}) \nleq (\mathrm{I}\Sigma_1 \overset{N}{\models})$ are proved in [20]. The proof of the Π_1-part is a modification of the proof of Thm.9 (v $\not\Rightarrow$ i) of [4]pp.59-93 where only Claim 9.4 (and its proof) is modified. For the Σ_1-part, the proofs of Thm.s 3-4 in [3] and in [4]pp.42-45 are also used. Actually using these proofs it is not very hard to modify the present proof of Thm.6 to prove $(\mathrm{I}\Sigma_1 \cup \mathrm{To} \overset{N}{\models}) \nleq (\mathrm{I}\Sigma_1 \overset{N}{\models})$.

(1.8) All the other inequalities indicated by $\neq$ or by $\nleq$ on Fig.2 are immediate consequences of (1.1)-(1.7) above and of the inclusions "$\leq$" and equivalences "$\equiv$" indicated there (which we turn to prove now).

<u>2) Proofs of equivalences</u> $(X \overset{Y}{\models}) \equiv (Y \overset{Z}{\models})$:

(2.1) $(\mathrm{Ict} \cap \mathrm{If} \overset{N}{\models}) \geq \overset{F}{\models}$ is proved in proofs of Prop.12 and Thm.9 (i $\Rightarrow$ ii) in Part II of [3], and also in [4]pp.57-58 and p.111. The detailed proof is given in proving Thm.9(i $\Rightarrow$ ii) in both quoted papers.

(2.2) By Fig.1, all the induction axiom systems Iname introduced in this paper are $\geq \mathrm{Ict} \cap \mathrm{If}$. Hence $(\mathrm{Iname} \overset{N}{\models}) \geq \overset{F}{\models}$ follows from (2.1) with the only exception of Ifm. It is not hard to check that $(\mathrm{Ifm} \overset{N}{\models}) \ngeq \overset{F}{\models}$.

(2.3) A <u>simple proof</u> of all the remaining equivalences $\equiv$ in

Fig.2 under the restriction that Th contains the Peano axioms is found in [6] which was first published in 1977 in Hungarian, see [1]. Even under this strong restriction, the question whether $(Ex \cup Ia \cup Tpa \overset{N}{\models}) \equiv$ $\equiv (Ia \cup Tpa \overset{N}{\models})$ remains an open problem.

(2.4) $(Iq \cup To \overset{N}{\models}) \leq \overset{F}{\models}$ is Thm.9(iii $\Rightarrow$ i) in Part II of [3] and in [4]p.56. A detailed proof arises if one reads Prop.7 of [9]p.121 together with [10].

(2.5) All the statements $(X \overset{Y}{\models}) \leq (Y \overset{Z}{\models})$ implicit in Fig.2 are easy consequences of (2.4) and (2.1) above. END of proofs of Fig.2.

ON THE INTUITIVE MEANING OF FIGURE 2

One of the central themes of Nonclassical Logic is the study of the lattice of the various modal logics. This activity turned out to be a rather fruitful part of modal logic providing much insight into the nature of modal reasoning. Analogously, on Fig.2, we investigate the lattice of the various dynamic logics $Dlog_d(Ax)$ for various $Ax \subseteq F_{td}$. We hope this might provide insight into the nature of reasoning about programs (or more generally, reasoning about consequences of actions).

For example, Thm.6 says that if the set of logical axioms Ax of our Dlog(Ax) contain full induction Ia over time then it does matter whether or not time instances can be compared by the "later than" relation. In this case the dynamic logic $Dlog(Ia \cup To)$ in which we can say "z_0 is later than z_1" is stronger (modulo HF_d) than the one $Dlog(Ia \cup Ts)$ in which we cannot.

As a contrast, if the logical axioms contain only restricted induction Iq over time then the logic $Dlog(Iq \cup To)$ with "later than" is not stronger than the one Dlog(Iq) without it. However, here the logic $Dlog(Iq \cup Tpres)$ in which we can perform addition on time is stronger than the one $Dlog(Iq \cup To)$ in which we cannot. Intuitively $z_0=z_1+z_2$ means that "z_0 is z_2 time after z_1".

Now we turn to the question "is sometime sometimes better..." in the title of [16]. The formulas in $(\Sigma_{0,t}F_{td})$ can be considered to be the formulas without time modalities "Sometime" and "Always". Hence Iq is time induction over all the formulas without time modalities (time induction over the non-modal formulas). The result $(Imd \overset{N}{\models}) > (Iq \cup To \overset{N}{\models})$ in Fig.2 can be interpreted to say that the logic Dlog(Imd) in which "Sometime" is available is indeed stronger than the one $Dlog(Iq \cup To)$ without "Sometime". But this result implies only that "Sometime" is better <u>if</u> we allow arbitrarily complex time-modality prenexes "Sometime$\exists x_0(x_0=y_0 \wedge$ Always$\exists x_1(x_1=y_1 \wedge$ Sometime$\varphi))$" see the definition of DF^{mod} (Def.18). This was not mentioned in the title of [16]. So a finicky interpretation of the quoted question might lead us to the "pure sometime logic" $Dlog(I\Sigma_1)$ in which we can perform time-induction over Sometimeφ with $\varphi \in (\Sigma_{0,t}F_{td})$ but we cannot do time-induction over "$\neg$Sometimeφ" or over "Sometime$\exists x_0(x_0=y_0 \wedge$ Always$\varphi)$". Thus the result $(I\Sigma_1 \cup To \overset{N}{\models}) > (Iq \cup To \overset{N}{\models})$ and the problem whether or not $(I\Sigma_1 \overset{N}{\models}) \equiv (Iq \overset{N}{\models})$ both in Fig.2 are relevant to a more careful analysis of the quoted question.

By another part of Fig.2, future tense "Sometime in the futureφ" as used e.g. in [12] adds to the reasoning power of dynamic logic $Dlog(Ia \cup Ts)$ with full time-induction. The rest of Fig.2 can be interpreted in this spirit, to investigate what kinds of logical constructs do increase the reasoning power (-s of which versions) of dynamic logic. Such logical constructs are "later than", "at z_0 time after z_1 it is the case that φ", "Sometimeφ" etc. By passing we note that it clearly shows on Fig.2 that the well known dynamic logics $\langle HFL_d, \overset{F}{\models} \rangle$, $\langle DL_d^{mod}, \models^{\underline{mod}} \rangle$, and $\langle DL_d^{fum}, \models^{\underline{fum}} \rangle$ are strictly increasing in this order in reasoning power modulo partial correctness of programs, i.e. modulo HF_d. That is $\overset{F}{\models} < \models^{\underline{mod}} < \models^{\underline{fum}}$.

We believe that Fig.2 is much more important for computer science than Fig.1, therefore we shall be sketchy in proving Fig.1.

<u>ON THE PROOFS OF FIGURE 1</u>

<u>The inclusions</u> indicated on the figure are straghtforward, except for $I' \models Imd$ and $Imd \models I'$. $I' \models Imd$ can be seen by observing that $mod(\varphi)$ is semantically equivalent to an element of I', for every $\varphi \in DF^{mod}$. The idea of the proof of $Imd \models I'$ is to translate I' into Imd. Instead of giving here the definition, we show the idea on an example. Let $\varphi \overset{d}{=} R(s_0, ext(y_0, sc(0)), ext(y_0, sc(z_0))))$. Then $\varphi' \overset{d}{=} \exists x_1 \exists x_2 [FirstNext(x_1=y_0) \wedge NextNext(x_2=y_0) \wedge R(x_0,x_1,x_2)]$. Now the translation of $ind(\varphi, z_0)$ is defined to be $[First\varphi' \wedge Alw(\varphi' \rightarrow Next\varphi')] \rightarrow \rightarrow Alw\varphi'$.

<u>On the inequalities</u> indicated on Fig.1.: $I' \not\models I1$ can be checked

by showing $I' \not\models \mathrm{ind}(R(\mathrm{ext}(y_0,z_0+z_0)),z_0)$ or $I' \not\models \mathrm{ind}(\mathrm{sc}(z_0)\neq 0,z_0)$. (These are proved in detail in [20]. In the proofs, models $\mathfrak{M}$ are constructed such that $\mathfrak{M} \models I'$. The proofs of $\mathfrak{M} \models I'$ are simplified versions of the proof of Claim 6.2 in the present paper.) By Fig.2 we have that $(\exists p \exists \psi)[\mathrm{Imd} \models \Box(p,\psi)$ but $\mathrm{Iq} \cup \mathrm{To} \not\models \Box(p,\psi)]$. Therefore $\mathrm{Iq} \not\models \mathrm{Imd}$, that is $\mathrm{Imd} \not\leq \mathrm{Iq}$ and hence $\mathrm{I1} \not\leq \mathrm{Iq}$. An easy argument shows that $\mathrm{I1} \not\models \mathrm{Iq}$, i.e. I1 and Iq are not comparable. By Fig.2, $\mathrm{Iq} \not\models I\Sigma_1$ and $\mathrm{Iq} \not\models I\Pi_1$. $I\Pi_1 \not\models \mathrm{I1}$ and $I\Sigma_1 \not\models \mathrm{I1}$ can be proved by [20] roughly by considering $\langle \underset{\sim}{T}, \underset{\sim}{T}, \{Id\}, \text{valueof} \rangle$ (but we did not check the details carefully). The remaining inequalities on Fig.1 are not hard. $I\Sigma_1 \not\models I\Pi_1$ and $I\Pi_1 \not\models I\Sigma_1$ are in [20]. <u>End of proof of Fig.1.</u>

<u>Intuitive motivation</u> for the second part of the present paper is a section entitled "Intuitive ... of Fig.2" in §5 immediately below the end of proof of Fig.2. To this we add that our Fig.2 is analogous with Fig.1 of the monograph [6 b] on first order modal logic and Kripke models. For the lattice of modal logics see e.g. [6 a], we point out this because the main result proved in the present paper concerns the lattice of dynamic logics.

R E F E R E N C E S

[1] Andréka,H. and Németi,I., Completeness of Floyd's program verification method w.r.t. nonstandard time models, Seminar Notes, Math. Inst.H.A.Sci.-SZKI 1977 (in Hungarian). This was abstracted in [2].

[2] Andréka,H. and Németi,I., Completeness of Floyd Logic, Bull. Section of Logic Wroclaw Vol 7, No 3, 1978, pp.115-121.

[3] Andréka,H. Németi,I. and Sain,I., A complete logic for reasoning about programs via nonstandard model theory. Part I, Part II. Theoret.Comput.Sci. 17(1982) no.2 and no.3.

[4] Andréka,H. Németi,I. and Sain,I., A complete first order dynamic logic. Preprint No. 810318, Math.Inst.H.A.S., Budapest, 1980.

[5] Andréka,H. Németi,I. and Sain,I., Henkin-type semantics for program schemes to turn negative results to positive. In: Fundamentals of Computation Theory'79 (Proc.Conf. Berlin 1979), Ed.: L. Budach, Akademie Verlag Berlin 1979. Band 2. pp.18-24.

[6] Andréka,H. Németi,I. and Sain,I., A characterization of Floyd provable programs. In: Mathematical Foundations of Computer Science'81 (Proc.Conf. Strbské Pleso Czechoslovakia 1981)Lecture Notes in Computer Science, Springer Verlag, 1981.

[6 a] Blok,W.J., The lattice of modal logics. J. Symbolic Logic. To appear.

[6 b] Bowen,K.A., Model theory for modal logic. D.Reidel Publ.Co., Boston 1979, x+127 pp.

[7] Burstall,R.M., Program proving as hand simulation with a little induction. IFIP Congress, Stockholm, August 3-10, 1974.

[8] Chang,C.C. and Keisler,H.J., Model Theory. North-Holland, 1973.

[9] Csirmaz,L., A survey of semantics of Floyd-Hoare derivability. CL&CL - Comput.Linguist.Comput.Lang. 14(1980)pp.21-42.

[10] Csirmaz,L., On the completeness of proving partial correctness. Acta Cybernet. To appear.

[11] Csirmaz,L. and Paris,J.B., A property of 2-sorted Peano models and program verification. Preprint Math.Inst.H.A.S. Budapest, 1981.

[12] Gabbay,D. Pnuely,A. Shelah,S. and Stavi,J., On the temporal analysis of fairness. Preprint, Weizmann Inst. of Science, Dept. of Applied Math., May 1981.

[13] Gergely,T. and Úry,L., Time models for programs. In: Mathematical Logic in Computer Science (Proc.Coll.Salgótarján 1978)Colloq.Math. Soc.J.Bolyai 26 Ed.s: Gergely,T. Dömölki,B. North-Holland, 1981. pp.359-427.

[14] Hájek,P., Making dynamic logic first-order. In: Mathematical Foundations of Computer Science'81 (Proc.Conf. Strbské Pleso Czechoslovakia 1981) Lecture Notes in Computer Science, Springer Verlag, 1981.

[15] Kfoury,D.J. and Park,D.M.R., On the termination of program schemas. Information & Control 29(1975),pp.243-251.

[16] Manna,Z. and Waldinger,R., Is "Sometime" sometimes better than "Always"? Intermittent assertions in proving program correctness. Preprint No. Z173, Stanford Research Inst., Menlo Park, June 1976.

[17] Monk,J.D., Mathematical Logic. Springer Verlag, 1976.

[18] Németi,I., Nonstandard runs of Floyd provable programs. Preprint, Math.Inst.H.A.S., Budapest, 1980.

[19] Németi,I., Hilbert style axiomatization of nonstandard dynamic logic. Preprint, Math.Inst.H.A.S., Budapest, 1980.

[20] Németi,I., Results on the lattice of dynamic logics. Preprint, Math.Inst.H.A.S., Budapest, 1981.

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

[22] Sain,I., There are general rules for specifying semantics: Observations on abstract model theory. CL&CL - Comput.Linguist.Comput. Lang. 13(1979),pp.251-282.

[23] Sain,I., First order dynamic logic with decidable proofs and workeable model theory. In: Fundamentals of Computation Theory'81 (Proc. Conf. Szeged 1981)Lecture Notes in Computer Science, Springer Verlag, 1981.