stichting
mathematisch
centrum

MC

AFDELING INFORMATICA
(DEPARTMENT OF COMPUTER SCIENCE)

IW 119/79 SEPTEMBER

J.A. BERGSTRA, J. TIURYN & J.V. TUCKER

CORRECTNESS THEORIES AND PROGRAM EQUIVALENCE

Preprint

2e boerhaavestraat 49 amsterdam

Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O).

1980 Mathematics subject classification: 03C52, 03D75, 68B10

ACM-Computing Reviews-category: 5.24

Correctness theories and program equivalence*)

by

J.A. Bergstra[1], J. Tiuryn[2] & J.V. Tucker

ABSTRACT

We address technical issues to do with the mathematical investigation of the idea that a programming language semantics can be sensibly determined by proof rules for the "before-after" assertions required true of programs in the language. It is shown that the theory of program correctness based on first-order logic can in a natual way specify the basic semantical concept of program equivalence in the case of programs terminating throughout any given class of data structures, but that diverging programs require rather delicate investigations into the associated classes.

KEY WORDS & PHRASES: *program semantics specification, first-order correctness theories, program equivalence.*

---

*) This report will be submitted for publication elsewhere.

1) Department of Computer Science, University of Leiden, Wassenaarseweg 80, Postbus 9512, 2300 RA LEIDEN, The Netherlands.

2) Mathematics Institute, University of Warsaw, PkiN 00-901 WARSAW, Poland. Through the course of this collaboration visiting Department of Computer Science II, RWTH Aachen, Buchel 29-31, 5100 AACHEN, West Germany.

1. INTRODUCTION

Algorithms are written in a *definite* program formalism $\underline{\underline{P}}$ and are designed to compute functions in a *definite* class of data structures K. Central to the theory of computation are the basic semantical notions of the *termination, correctness, equivalence* and *isomorphism* of programs in $\underline{\underline{P}}$ as these are defined by the systems of K. Here we reveal, and attempt to clarify, certain perplexing technical problems which arise in the attempt to analyse program equivalence in classes $\underline{\underline{P}}$, intended to operate on classes K, by means of (proof systems founded on) the first-order logical assertion method so successfully developed for program correctness, see the survey of K.R. APT [1]. One reason for doing this is that in I. GREIF & A. MEYER's [10, 11] is to be found the beginnings of an interesting mathematical investigation of the thesis that a programming language semantics could be usefully specified by proof rules for the "before-after" assertions to be deemed true of programs in the language, an idea advocated by several writers: E.W. DIJKSTRA [6], C.A.R. HOARE [12, 13], R.W. FLOYD [7], Z. MANNA [17]; see also HOARE & WIRTH [15], HOARE & LAUER [14], a point of departure for [10]. And essential to a denotational semantics approach, such as Greif and Meyer's, is program equivalence. Another reason is to draw attention to the need of significantly deeper logical understanding of the relation of program equivalence - perhaps the most notable contribution to which is DE BAKKER's [2]; for example, on that relation must be founded any axiomatic work on program transformations in the fashion of V.K. SABELFELD's [26].

Henceforth we involve ourselves only with the following technical issues whose formulation derives from several interesting questions about partial correctness theories asked by A. MEYER [19] in connection with his interest in semantic specification: for these we wish to express our thanks to him.

Let P be some kind of program scheme over a finite signature Σ and let K be a class of relational systems or *data structures* of that type Σ. For each $A \in K$, P computes a partial function on A and one defines two programs P, Q to be K-*equivalent*, $P \equiv_K Q$, if for any $A \in K$ and every $a \in A^n$, $P(a) \simeq Q(a)$ (that is, either P(a), Q(a) are both defined and are equal or are both undefined).

Let $L = L(\Sigma)$ be the first-order logical language of Σ with equality.

The *total correctness theory of* P *in* L *with respect to* K is

$$TC_K(P) = \{(\alpha(x),\beta(y)) \in L^2 : \text{for all } A \in K,\ A \models \alpha(x) \to [P(x)\downarrow \wedge \beta P(x)]\}$$

and the *partial correctness theory of* P *in* L *with respect to* K is

$$PC_K(P) = \{(\alpha(x),\beta(y)) \in L^2 : \text{for all } A \in K,\ A \models \alpha(x) \to [(P(x)\downarrow \wedge \beta P(x)) \vee P(x)\uparrow]\}$$

where the $x = (x_1,\ldots,x_n)$ and y are the only free variables of α and β, and are determined by P; and the meanings of $P(x)\downarrow$ and $P(x)\uparrow$ are convergence and divergence although these are not first-order.

We enquire into the circumstances where

(1) $TC_K(P) = TC_K(Q)$ implies $P \equiv_K Q$

(2) $PC_K(P) = PC_K(Q)$ implies $P \equiv_K Q$

In §1 we quickly provide evidence of a paucity of such situations and so turn instead to these modified theories (cf. the discussion of correctness in MANNA [18, pp. 164-5]).

The *modified total correctness theory of* P *in* L *with respect to* K is

$$MTC_K(P) = \{(\alpha(x),\beta(x,y)) \in L^2 : \text{for each } A \in K,\ A \models \alpha(x) \to [P(x)\downarrow \wedge \beta(x,Px)]\}$$

and the *modified partial correctness theory of* P *in* L *with respect to* K is

$$MPC_K(P) = \{(\alpha(x),\beta(x,y)) \in L^2 : \text{for each } A \in K,\ A \models \alpha(x) \to [(P(x)\downarrow \wedge \beta(x,Px)) \vee P(x)\uparrow]\}.$$

For any always terminating programs over any K we find that

(3) $\quad MTC_K(P) = MTC_K(Q) \quad$ implies $P \equiv_K Q$,

(4) $\quad MPC_K(P) = MPC_K(Q) \quad$ implies $P \equiv_K Q$,

and so we begin, in §2, to examine arbitrary programs over various classes seeking this so-called *logical determinateness* for their semantics. Here $MTC_K(P)$ may be seen to factor out the distinction between $MPC_K(P)$ and $PC_K(P)$ and although (3) and (4) may fail even for a K a variety (§3) the behaviour of (4) can be intriguing: we study local classes and axiomatisable classes (§4) and complete and ω-categorial axiomatisable classes (§5), the class of all structures $ALG(\Sigma)$ (§7), and subclasses of Peano Arithmetic (§8 and Appendix 1). The need for determinateness in some general form is satisfied in §6 by extending the logic L to include some arithmetic.

Whatever the final resolution of the problem of determinateness and its rôle in investigations such as Greif and Meyer's, it seems to us that these mathematical studies persued further will provide considerable technical insight into the logical aspects of computation: one area of obvious importance is to calculate the affect of Peano Arithmetic on simple-minded programming on the natural numbers (see §8 and Appendix 1). Hopefully, there can be developed a rich, classically styled, model theory of programs.

For unexplained ideas in the theory of computation, logic or algebra we refer the reader to MANNA [18], CHANG & KEISLER [5], and MAL'CEV [16] respectively. The use of correctness theories is well established in the references previously cited, see also the text-book DE BAKKER [3]; the account of computations in algebraic systems in [29] may be useful for background material.

0. PROGRAMS ON ALGEBRAS

The concept of determinateness involves the three parameters of a program language $\underline{\underline{P}}$, a class of data structures, or *data type*, K and a logical language L. Of these the last two require most attention in the technical work which follows, for any of the common designs of (deterministic) program

schemes, appropriate for abstract structures, may serve as $\underline{P}$ providing they contain statements allowing the evaluation of basic operations and relations of the algebras, and are closed under if * then * else statements and composition; thus straight-line programs are about the weakest formalism for which all our results remain valid.

If we have a specific computing formulae in mind it is the various classes of *finite algorithmic procedures*, or *faps*, developed in FRIEDMAN [8], MOLDESTAD, STOLTENBERG-HANSEN & TUCKER [22, 23] and SHEPHERDSON [27]; or, possibly, the *effective definitional schemes* of FRIEDMAN [8], but this represents only one computational power equivalent to the *finite algorimic procedures with both stacking and counting on natural numbers* (*fapCS*'s), in [23], and to *finite algorithmic procedures with index regieters* (*fapirs*), in [27]. (In [24] six disparate methods of defining computability on algebras are explained and classified in terms of the fap formalism; the power of fapCS-computability turns out to be maximal among the truely constructive computing strengths possible in an abstract setting.) The semantics of faps are determined by them being assembler code for straightforward types of register machines generalized to abstract algebras, for details see [8, 23, 24, 27, 29]. However, the reader familiar with MANNA [18] or GREIBACH [9] should find no difficulty whatever in following all arguments here presented. Specific items we use all the time are these.

Given a program P applied to input $a_1,\ldots,a_n \in A$ we take as understood formulations of the *state descriptions* of the computation $P(a_1,\ldots,a_n)$ and its *length of computation* denoted $|P(a_1,\ldots,a_n)|$. It is also easy to show these facts, [29]:

0.1 LOCALITY OF COMPUTATION LEMMA

In any computation $P(a_1,\ldots,a_n)$ *the elements of* A *appearing in every state description of* $P(a_1,\ldots,a_n)$ *all lie within* $\langle a_1,\ldots,a_n\rangle$, *the subalgebra of* A *generated by* $a_1,\ldots,a_n$. *In particular, the output* $P(a_1,\ldots,a_n) \in \langle a_1,\ldots,a_n\rangle$.

0.2 UNIQUENESS OF COMPUTATION LEMMA

Let A *and* B *be algebras of signature* Σ *isomorphic by* ϕ. *For any program* P *over* Σ *and any input* $a_1,\ldots,a_n \in A$, $\phi P(a_1,\ldots,a_n) \simeq P(\phi a_1,\ldots,\phi a_n)$.

1. PROGRAMS WHICH ALWAYS TERMINATE

First of all it is trivial to see that over the natural numbers, ω, both (1) and (2) are true of any kind of program. This is an instance of the equally obvious fact that they hold when K consists of a single *prime* algebra A (that is, an A containing no proper subalgebras). For example, when K contains just one of the prime rings with identity, $\mathbb{Z}$ or $\mathbb{Z}_n$, or just one of the prime fields Q or $\mathbb{Z}_p$. The purpose of this first proposition is to generate some equally simple counter-examples.

THEOREM 1.1. *Let* $\underline{\underline{P}}$ *be a class of programs over* Σ *and* K *a class of data structures of type* Σ *satisfying these two properties: there is an* $f \in \underline{\underline{P}}$ *such that* (i) *there exist* $A \in K$, $a \in A$ *where* $f(a) \neq a$, *and* (ii) *for each* $A \in K$, f *computes an automorphism of* A. *Then implications* (1) *and* (2) *fail for* $\underline{\underline{P}}$ *and* K.

PROOF. Take P to be f and Q a program for the identity map. Hypothesis (i) asserts that $P \not\equiv_K Q$. We show $TC_K(P) = TC_K(Q)$ and since both P and Q are total this suffices to prove (1) and (2) fail. Assume for a contradiction that these sets do not coincide.

Case 1. There exist (α,β) such that $(\alpha,\beta) \in TC_K(P)$ but $(\alpha,\beta) \notin TC_K(Q)$. Using the termination of P, Q and the definition of Q we can write this precisely as

(a) $\quad \forall A \in K,\ A \models \alpha(x) \rightarrow \beta P(x)$

and

(b) $\quad \exists B \in K,\ B \not\models \alpha(x) \rightarrow \beta(x)$.

Given (b), choose $B \in K$ and $b \in B$ so that $B \not\models \alpha(b) \rightarrow \beta(b)$; hence $B \models \alpha(b) \wedge \neg\beta(b)$. From (a), $B \models \alpha(b) \rightarrow \beta P(b)$ and $B \models \beta P(b)$. Since $B \not\models \beta(b)$, $P(b) \neq b$

while $P(b) = f(b)$ entails $B \models \beta f(b)$. By hypothesis (ii) f computes an automorphism of B say ϕ and, since β is first-order, $\beta(x)$ if, and only if, $\beta\phi(x)$. Hence $B \models \beta\phi(b)$ entails $B \models \beta(b)$ which is a contradiction.
Case 2. there exist (α,β) such that $(\alpha,\beta) \in TC_K(Q)$ but $(\alpha,\beta) \notin TC_K(P)$. This leads to a contradiction in the same way. Q.E.D.

In applying 1.1 we desire $\underline{\underline{P}}$ to be as simple a programming formula as possible, in our examples $\underline{\underline{P}}$ is the class of straight-line programs in the appropriate signature.

ABELIAN GROUPS 1.2. Let $f_n(x) = nx = x + \ldots + x$ (n times). For any torsion-free divisible abelian group A and any $n \neq 0$, f_n is an automorphism of A. So take K to be any class of such groups for which one can choose an n such that f_n is not the identity on some group in K.

FINITE FIELDS 1.3. Let F be a finite field of characteristic p. Then $f(x) = x^p$ is a field automorphism of F. If F is not $\mathbb{Z}_p$ then f is not the identity. So take K to be any class of finite fields of characteristic p containing at least one $GF(p^n)$ for $n \neq 0,1$; in particular take $K = \{GF(p^n)\}$ for any $n \neq 0,1$. (Remember (1) and (2) hold for $K = \{\mathbb{Z}_p\}$.)

INVOLUTIONS 1.4. An *involution* * of a (not necessarily commutative) ring R is an automorphism such that for all $r \in R$, $r^{**} = r$. Take K to be any class of rings with involution containing at least one R where the involution is not the identity. For example, let K contain just the complex number field C with complex conjugation $a+ib \to a-ib$. Or, to cite an example from the theory of linear equations, use the ring of 2×2 matrices over a field with the symplectic involution defined $\begin{pmatrix} a & b \\ c & d \end{pmatrix}^* = \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$.

Of course in this section where programs always terminate there are no distinctions between (1), (2) and (3), (4); henceforth we explicitly mention only the total theories.

THEOREM 1.5. *Let $\underline{\underline{P}}$ be a class of programs over Σ and let K be a class of data structures of type Σ. If $P,Q \in \underline{\underline{P}}$ define total functions on each $A \in K$ then*

$$MTC_K(P) = MTC_K(Q) \quad \textit{implies } P \equiv_K Q.$$

PROOF. Assume $P \not\equiv_K Q$. Then there exists $A \in K$ and $a \in A^n$ such that $P(a) \neq Q(a)$ in A. Let $P(a)$ terminate in t steps. Since the entire computation of $P(a)$ takes place within the subsystem $\langle a \rangle$ of A (Lemma 0.1) there exists a Σ polynomial τ such that $P(a) = \tau(a)$ and, moreover, we can syntactically unfold the computation of $P(a)$ into a first-order boolean formula of n variables $\phi(x)$ which is true of $b \in B^n$, for $B \in K$, if, and only if, $P(b)$ follows precisely the same route as $P(a)$ in terminating in t steps with $P(b) = \tau(b)$. Define $\theta(x,y) \equiv \phi(x) \rightarrow y = \tau(x)$. It is easy to see that the pair $(x = x, \theta(x,y))$ lies in $MTC_K(P)$ but not in $MTC_K(Q)$. Q.E.D.

Despite this initial success of the modification to the correctness theories for total programs non-termination introduces elaborate technical difficulties as the following sections will illustrate. In hindsight it can only be to the advantage of everyone interested in these problems if further classification of the determinateness properties of the unimodified theories is made. R. Parikh has recently proved an attractive theorem in this connection, and has independently observed the possibility of non-determinateness in the case of Example 1.3, see [25].

We have one result worth mentioning here.

THEOREM 1.6. *Let Σ be a finite signature containing at least one function symbol and let K be the species of Σ, that is the class ALG(Σ) of all structures of type Σ. Let $\underline{\underline{P}}$ be a class of programs over Σ. If $P,Q \in \underline{\underline{P}}$ compute total functions throughout K then*

$$TC_K(P) = TC_K(Q) \quad \textit{implies } P \equiv_K Q.$$

PROOF. Suppose $P \not\equiv_K Q$ and let $A \in K$, $a \in A^n$ be such that $P(a) \neq Q(a)$ in A. The computations of $P(a)$ and $Q(a)$ lie within the subsystem $\langle a \rangle$ of A generated by a. So let those elements of $\langle a \rangle$ appearing in these computations be the list $\underset{\sim}{a} = (a_0,\ldots,a_N)$ where $a_0 = a$ and $a_N = P(a)$.

Let $D_{\underset{\sim}{a}}$ be the set of all open formulae of N+1 variables in L which are true of $\underset{\sim}{a}$ in A. The set $D_{\underset{\sim}{a}}$ is equivalent to a single formula $d_{\underset{\sim}{a}} \in L$. Notice

that $d = d_{\underset{\sim}{a}}$ is consistent with respect to K and that for any $B \in K$, if d is realized in B then P and Q are not equivalent over B. We propose to define from d a new formula $b = b_{\underset{\sim}{a}}$ such that

(i) $b(x) \to d(x)$;

(ii) $b(x)$ is consistent with respect to K;

(iii) there exist formulae $\phi_i(x_i)$, $0 \leq i \leq N$, such that if $B \in K$ realizes $b(x)$ at $\underset{\sim}{b} = (b_0,\ldots,b_N)$ then $\{b \in B: \phi_i(b)\} = \{b_i\}$ for $0 \leq i \leq N$.

From this the theorem follows because by (i) and (iii)

$$K \models b(x_0,\ldots,x_N) \to P(x_0)\downarrow \wedge \phi_N P(x_0),$$

but by (ii),

$$K \not\models b(x_0,\ldots,x_N) \to Q(x_0)\downarrow \wedge \phi_N Q(x_0).$$

b is defined as follows. First let $\mathcal{O}(y)$ be the statement "for each function (symbol) f of Σ, $y \notin \mathrm{im}(f)$". Since Σ is finite, $\mathcal{O}(y)$ is first-order expressible. Now choose some k-ary function f of Σ and define, for $0 \leq i \leq N$, the formula $\phi_i(x)$ to first-order express the statement "x is that unique element such that for *exactly* i+1 distinct elements z_j, $0 \leq j \leq i$, $\mathcal{O}(z_j)$ and $x = f(z_j,\ldots,z_j)$". Clearly, ϕ_i can be satisfied by *at most* one element of any K-algebra.

Set $b(x_0,\ldots,x_N) \equiv d(x_0,\ldots,x_N) \wedge \bigwedge_{i=0}^{N} \phi_i(x_i)$.

Conditions (i) and (iii) are immediate. To show (ii), starting from $a_0,\ldots,a_N \in A$, we construct a $B \in K$ satisfying $b(x)$. Let T[X] be the Σ-term algebra on the indeterminates $X_0,\ldots,X_N$. By the freeness property of T[X] for K, there is a congruence $\equiv_{\underset{\sim}{a}}$ on T[X] such that $T[X]/\equiv_{\underset{\sim}{a}}$ is isomorphic to $\langle \underset{\sim}{a} \rangle$ on which $d(x)$ is satisfied. Now for each $0 \leq j \leq N$ we take the j+1 indeterminates $Y^j_1,\ldots,Y^j_{j+1}$ and the equations $f(Y^j_i,\ldots,Y^j_i) = X_j$, for $1 \leq i \leq j+1$.

Let $T[X,Y] = T[X][Y^j_i: 1 \leq i \leq j+1, 0 \leq j \leq N]$ and divide it by the congruence $\equiv$ generated by $\equiv_{\underset{\sim}{a}}$ together with the equations. Setting $B = T[X,Y]/\equiv$ it is routine, if tedious, to verify that in B the only elements satisfying $\mathcal{O}$ are the Y^j_i and that the only elements satisfying ϕ_i and $d(x)$ are the X_i.

Q.E.D.

The class of *all* interpretations of a program formalism is of very limited semantical interest. One significant case where Theorem 1.6 is relevant is the class of all groupoids (that is, of all non-associative semigroups) which is the species of a single binary operation: the data structures on which one parses syntactic expressions are members of this category. A theroem about this class is also available for general programs, but with some considerable effort, see §7.

2. SOME GENERAL PROPERTIES OF PARTIAL CORRECTNESS THEORIES

Theorem 1.5 rests on the simple technical advantage of the modified correctness theories that they may contain first-order expressions of the structure and output of any computation of program P which runs for a fixed time t. Formally, to each program P, considered as having n input variables ($n \geq 0$), there corresponds sequences of first-order formulae $\phi_t^P(x_1,\ldots,x_n)$ and polynomials $\tau_t^P(x_1,\ldots,x_n)$ such that for all $t \in \omega$ and any input $a_1,\ldots,a_n$ from data structure A

$$A \models \phi_t^P(a_1,\ldots,a_n) \qquad \text{if, and only if } |P(a_1,\ldots,a_n)| = t$$

and,

$$\text{if } A \models \phi_t^P(a_1,\ldots,a_n) \text{ then } P(a_1,\ldots,a_n) = \tau_t^P(a_1,\ldots,a_n).$$

In particular,

$$A \models P(a_1,\ldots,a_n) \qquad \text{if, and only if } A \models \bigvee_{t\in\omega} \phi_t^P(a_1,\ldots,a_n).$$

It is important to observe that (gödel numbers for) the formulae and polynomials can be recursively calculated uniformly in the (codes for) programs. And whilst there is a single program S over ALG(Σ) (necessarily of the power of finite algorithmic procedures with stacking and counting) which decides ϕ_t^P on inputs from A, uniformly in codes e_P for programs P and run times t,

$$\begin{aligned} S(e_P,a,t) &= 0 \qquad \text{if } A \models \phi_t^P(a) \\ &= 1 \qquad \text{otherwise} \end{aligned}$$

(see [23] where step counting is proved to be a *complexity measure* for fapCS computation on abstract algebras), the business of deciding for *fixed* P and t whether or not given $a \in A^n$, $A \models \phi_t^P(a)$, can be accomplished by a straight-line program.

The formal proofs of there observations we leave as an instructive exercise to the reader. This section collects together some formal relationships between partial correctness theories and their modifications and concludes with an important technical lemma.

Let $S = \{(\alpha(x),\beta(y)): \alpha,\beta \in L\}$. Then for any program P over any class,

$$PC_K(P) = MPC_K(P) \cap S$$

and so

LEMMA 2.1. *For any programs* P, Q *over any class* K, *if* $MPC_K(P) = MPC_K(Q)$ *then* $PC_K(P) = PC_K(Q)$.

That the converse is false, even for total programs, follows from 1.1 and 1.5, of course. However,

THEROEM 2.2. *For any programs* P,Q *over any class* K, *if* $MTC_K(P) = MTC_K(Q)$ *then* $PC_K(P) = PC_K(Q)$ *if, and onlu if,* $MPC_K(P) = MPC_K(Q)$.

PROOF. From the hypothesis on total correctness we deduce that

$$PC_K(P) = PC_K(Q) \quad \text{implies } MPC_K(P) = MPC_K(Q)$$

Contrapositively, suppose $(\alpha,\beta) \in MPC_K(P) - MPC_K(Q)$. Then

$$K \models \alpha(x) \rightarrow [(P(x)\downarrow \wedge \beta(x,P(x))) \vee P(x)\uparrow]$$

$$K \not\models \alpha(x) \rightarrow [(Q(x)\downarrow \wedge \beta(x,Q(x))) \vee Q(x)\uparrow]$$

Choose $A \in K$ and $a \in A^n$ such that

$$A \models \alpha(a) \wedge \neg[(Q(a)\downarrow \wedge \beta(a,Q(a))) \vee Q(a)\uparrow].$$

Then $A \models \alpha(a) \wedge Q(a)\downarrow \wedge \neg\beta(a,Q(a))$. Now we express the computation of $Q(a)$ in the formula ϕ_t^Q and polynomial τ_t^Q, for $t = |Q(a)|$, so that $A \models \phi_t^Q(a)$ and $A \models \phi_t^Q(a) \rightarrow Q(a) = \tau_t^Q(a)$; notice that $(\phi_t^Q(x), y = \tau_t^Q(x)) \in MTC_k(Q)$ and consider the pair $(\phi_t^Q(x) \wedge \neg\beta(x,\tau_t^Q(x)), y \neq y)$.

This pair does not lie in $PC_K(Q)$ because $A \models \phi_t^Q(a) \wedge \neg\beta(a,\tau_t^Q(a)) \wedge Q(a)\downarrow$. However, it does lie in $PC_K(P)$. To see this let $B \in K$ and $b \in B^n$ such that $B \models \phi_t^Q(b) \wedge \neg\beta(b,\tau_t^Q(b))$. As $MTC_K(P) = MTC_K(Q)$, $(\phi_t^Q(x), y = \tau_t^Q(x)) \in MTC_K(P)$ and we have $B \models P(b) = \tau_t^Q(b)$, and hence that $B \not\models P(b)\downarrow \wedge \beta(b,P(b))$ and $B \models P(b)\uparrow$. This establishes for any $B \in K$ and any $b \in B^n$

$$B \models \phi_t^Q(b) \wedge \neg\beta(b,\tau_t^Q(b)) \rightarrow [(P(b)\downarrow \wedge P(b) \neq P(b)) \vee P(b)\uparrow].$$

That is, $(\phi_t^Q(x) \wedge \neg\beta(x,\tau_t^Q(x)), y \neq y) \in PC_K(P)$. Q.E.D.

It is convenient to remark at this point that properties of terminations of general programs are not determined by partial correctness although the proof requires work of later sections:

PROPOSITION 2.3. *There is a class* K *and programs* P, Q *for which*

$$MPC_K(P) = MPC_K(Q) \qquad \textit{but} \qquad MTC_K(P) \neq MTC_K(Q)$$

PROOF. This anticipates a construction of the appendix: choose P, Q and K as in example A.3. To see that $MTC_K(P) \neq MTC_K(Q)$ observe that $(x = x, x = x) \in MTC_K(P) - MTC_K(Q)$ because otherwise Q would be everywhere total on K. Q.E.D.

We now prove a technical result which in localizing the semantics of computations across classes of structures pin points the source of certain difficulties in obtaining logical determinateness; it also enables us to avoid repeating some patterns of argument later on.

Let A be an algebra of signature Σ and let Γ be a signature extending Σ, so that $\Sigma \subset \Gamma$. The Σ reduct of algebra B of signature Γ is the structure $B|_\Sigma$ with domain that of B and whose operations and relations are those of B named in Σ; B is said to be a Γ *expansion* of A if $B|_\Sigma = A$.

If K is a class of structures of signature Σ and $\Sigma \subset \Gamma$ then by $K(\Gamma)$ we

denote the class of all Γ expansions of all K-algebras. In particular, if $\Gamma = \Sigma \cup \{\underline{c}_1,\ldots,\underline{c}_n\}$ where $\underline{c}_1,\ldots,\underline{c}_n$ are new constant symbols relative to Σ then $K(\Gamma)$ consists of all algebras of the form $(A,a_1,\ldots,a_n)$ where $A \in K$ and $(a_1,\ldots,a_n) \in A^n$. The sort of localization of semantics we have in mind is suggested by this obvious equivalence: for any first-order formula $\phi(x_1,\ldots,x_n)$ over Σ,

$$K = \forall x_1,\ldots,x_n.\phi(x_1,\ldots,x_n) \quad \text{if, and only if, } K(\Gamma) = \phi(\underline{c}_1,\ldots,\underline{c}_n),$$

where $\phi(\underline{c}_1,\ldots,\underline{c}_n)$ is a first-order sentence over Γ.

Call a program *closed* if it has no input variables.

LOCALIZATION LEMMA 2.4. *Let* K *be a class of algebras of signature* Σ. *The following statements are equivalent:*

(i) *for all programs* P, Q *over* Σ,

$$MPC_K(P) = MPC_K(Q) \qquad \textit{implies } P \equiv_K Q$$

(ii) *for all finite extensions of* Σ *by constants to* Γ, *and for each closed program* P *over* Γ, *if* P *diverges on some* $A \in K(\Gamma)$ *then there is a sentence* θ, *first-order over* Γ, *which is consistent with* $K(\Gamma)$ *and such that* $K(\Gamma) = \theta \to P\uparrow$.

PROOF. Consider (i) implies (ii). Let P be a closed program over $\Gamma \supset \Sigma$ involving constants $\underline{c}_1,\ldots,\underline{c}_n \in \Gamma - \Sigma$. Obviously one can take $P = P_0(\underline{c}_1,\ldots,\underline{c}_n)$ where $P_0(x_1,\ldots,x_n)$ is a program over Σ. Suppose $A \models P\uparrow$ for some $A \in K(\Gamma)$. We must make a trivial technical case distinction: assume $n \neq 0$. Let Q, R be these programs over Σ:

$$Q(x_1,\ldots,x_n) = x_1$$

$$R(x_1,\ldots,x_n) = \underline{\text{if}}\ P_0(x_1,\ldots,x_n)\downarrow\ \underline{\text{then}}\ x_1\ \underline{\text{else}}\ \uparrow\ \underline{\text{fi}}.$$

Clearly, $Q \not\equiv_K R$ since Q is everywhere convergent whereas R is not. By their definition, $MPC_K(Q) \subset MPC_K(R)$ while hypothesis (i) entails there is

$(\alpha,\beta) \in MPC_K(R) - MPC_K(Q)$. We show we can take the following first-order formulae over Γ to be a $\mathcal{O}$ which satisfies (ii):

$$\mathcal{O} = \alpha(\underline{c}_1,\ldots,\underline{c}_n) \wedge \neg\beta(\underline{c}_1,\ldots,\underline{c}_n,\underline{c}_1).$$

First examine consistency. Since $(\alpha,\beta) \in MPC_K(Q)$ there is a $B \in K$ and $b = (b_1,\ldots,b_n) \in B^n$ such that $B \not\models \alpha(b) \rightarrow [Q(b)\downarrow \wedge \beta(b,Q(b))] \vee Q(b)\uparrow$. So

$$B \models \alpha(b_1,\ldots,b_n) \wedge Q(b_1,\ldots,b_n) \wedge \neg\beta(b_1,\ldots,b_n,b_1)$$

and $(B,b_1,\ldots,b_n) \in K(\Gamma)$ is a structure on which $\mathcal{O}$ is satisfied under the interpretation of $\underline{c}_i$ as b_i, $1 \leq i \leq n$.

Next suppose $B \models \mathcal{O}$ for some $B \in K(\Gamma)$. As $(\alpha,\beta) \in MPC_K(R)$ we see that

$$B|_\Sigma \models \alpha(x_1,\ldots,x_n) \rightarrow [R(x_1,\ldots,x_n)\downarrow \wedge \beta(x_1,\ldots,x_n,x_1)] \vee$$

$$\vee R(x_1,\ldots,x_n)\uparrow$$

Since R converges precisely where P converges we can deduce

$$B \models \alpha(\underline{c}_1,\ldots,\underline{c}_n) \rightarrow [P\downarrow \wedge \beta(\underline{c}_1,\ldots,\underline{c}_n,\underline{c}_1)] \vee P\uparrow$$

which together with $B \models \mathcal{O}$ implies $P\uparrow$ on B.

In the case $n = 0$ we have only to consider the case where Σ contains a constant $\underline{c}$ where the argument above works for Q, R redefined as $Q = \underline{c}$ and $R = \underline{\text{if}}\ P\downarrow\ \underline{\text{then}}\ \underline{c}\ \underline{\text{else}}\ \uparrow\ \underline{\text{fi}}$.

Consider (ii) implies (i). Suppose $P \not\equiv_K Q$. We must show $MPC_K(P) \neq MPC_K(Q)$ on the basis of (ii). There are essentially two cases.

First, for some $A \in K$ and $a \in A^n$, P(a) and Q(a) converge but $P(a) \neq Q(a)$. Here we can use the first-order expression of their computations: if $|P(a)| = t$ and $|Q(a)| = s$ then the pair $(\phi_t^P(x) \wedge \phi_s^Q(x) \wedge \tau_t^P(x) \neq \tau_s^Q(x), y = \tau_t^P(x))$ lies in $MPC_K(P)$ but not in $MPC_K(Q)$.

Secondly, for some $A \in K$ and $a = (a_1,\ldots,a_n) \in A^n$, P(a) converges but Q(a) diverges. (The third case exchanges the hypothesis between P and Q and follows *mutatis mutandis*.)

Let $|P(a)| = t$ and define a new program R by

$$R(x) = \underline{\text{if}}\ \phi_t^P(x)\ \underline{\text{then}}\ x_1\ \underline{\text{else}}\ Q(x)\ \underline{\text{fi}}$$

Notice that R does not require programming features beyind those assumed for P and Q depends upon the observation that a straight-line program can be written to decide any given $\phi_t^P(x)$.

From our hypothesis it follows that $R(a)\uparrow$. Let $B = (A,a_1,\ldots,a_n)$ and $\Gamma = \Sigma \cup \{\underline{c}_1,\ldots,\underline{c}_n\}$ where the $\underline{c}_i$ are constant symbols new to Σ. Clearly, $B \models R(\underline{c}_1,\ldots,\underline{c}_n)\uparrow$ as $\underline{c}_i$ is interpreted as a_i, $1 \leq i \leq n$. By statement (ii) there is a sentence $\mathcal{O}$, first-order over Γ, which is consistent with $K(\Gamma)$ and such that $K(\Gamma) \models \mathcal{O} \rightarrow R(\underline{c}_1,\ldots,\underline{c}_n)\uparrow$. Let $\mathcal{O}_0$ be $\mathcal{O}$ with variables x_i replacing constants $\underline{c}_i$, $1 \leq i \leq n$. We claim that $(\mathcal{O}_0(x), y \neq y) \in MPC_K(Q) - MPC_K(P)$.

The pair cannot lie in $MPC_K(P)$ because $K \models \mathcal{O}_0(x) \rightarrow P(x)\downarrow$ and $y \neq y$ is false. On the other hand the pair does lie in $MPC_K(Q)$ because $K \models \mathcal{O}_0(x) \rightarrow [R(x)\uparrow \wedge \phi_t^P(x)]$ and $K \models [R(x)\uparrow \wedge \phi_t^P(x)] \rightarrow Q(x)\uparrow$. Q.E.D.

3. A COUNTER-EXAMPLE TO DETERMINATENESS ON A VARIETY

Whilst the modified correctness theories worked perfectly for always terminating programs they fail to characterize partial programs:

THEOREM 3.1. *Let Σ be a signature containing two unary functions* f, g *let* K *be the variety of algebras of type Σ defined by the equation*

$$fg(x) = gf(x) = x.$$

Then there exist flowchart programs P *and* Q *such that*

$$MPC_K(P) = MPC_K(Q) \qquad \text{but } P \not\equiv_K Q$$

PROOF. Let P compute the two argument projection function $P(x,y) = x$ throughout K. For Q we require

$$\begin{aligned} Q(x,y) &= x \quad \text{if } \langle x\rangle \text{ or } \langle y\rangle \text{ is finite or } x \in \langle y\rangle \text{ or } y \in \langle x\rangle, \\ &= \uparrow \quad \text{otherwise.} \end{aligned}$$

Given the equation defining K, it is straightforward to design a flow chart program which is such a Q. Clearly $P \not\equiv_K Q$.

Assume for a contradiction that $MPC_K(P)$ and $MPC_K(Q)$ are distinct. Clearly, $MPC_K(P) \subset MPC_K(Q)$ so let $\alpha,\beta \in L$ such that

$$K \models \alpha(x,y) \rightarrow [(Q(x,y)\downarrow \wedge \beta(x,y,Q(x,y)) \vee Q(x,y)\uparrow]$$

$$K \not\models \alpha(x,y) \rightarrow [(P(x,y)\downarrow \wedge \beta(x,y,P(x,y)) \vee P(x,y)\uparrow].$$

Applying the known properties of P, Q these expressions simplify to

$$K \models \alpha(x,y) \rightarrow [(Q(x,y)\downarrow \wedge \beta(x,y)) \vee Q(x,y)\uparrow]$$

$$K \models \alpha(x,y) \rightarrow \beta(x,y).$$

Let $\gamma(x,y) \equiv \alpha(x,y) \wedge \neg\beta(x,y)$ and observe that $K \models \gamma(x,y) \rightarrow Q(x,y)\uparrow$ and so for $A \in K$ if $A \models \gamma(x,y)$ then $\langle x\rangle$ and $\langle y\rangle$ are infinite and $x \notin \langle y\rangle$, $y \notin \langle x\rangle$. Choose $A \in K$ and $a,b \in A$ such that $A \models \gamma(a,b)$; to this step we produce a contradiction.

To L we add a constant symbol $\underline{a}$ to obtain $L_{\underline{a}}$, then $A \models \gamma(\underline{a},b)$ and $A \models \gamma(\underline{a},y) \rightarrow Q(\underline{a},y)$. Let $T = Th(A,a)$, the set of all sentences of $L_{\underline{a}}$ true in A with $\underline{a}$ assigned a. To make an elementary embedding of A into a certain $B \in K$ we add a new constant symbol $\underline{c}$ to $L_{\underline{a}}$ and define the subset of $L_{\underline{a},\underline{c}}$

$$T' = \{\neg\gamma(\underline{a},\underline{c}),\ \langle\underline{c}\rangle \text{ is infinite},\ \underline{c} \notin \langle\underline{a}\rangle\}$$

where it is easy to express "$\langle\underline{c}\rangle$ is infinite" and "$\underline{c} \notin \langle\underline{a}\rangle$" in first-order terms, given the special definition of K.

By a routine application of the *Compactness Theorem* [5, p. 67], the set of sentences $T \cup T'$ can be shown to have a model $B \in K$. And clearly in such B there are a, b, c such that

$$B \models \gamma(a,b) \qquad \text{and} \qquad B \models \neg\gamma(a,c).$$

We now use the following fact, which is easy to prove from the specifications

of K: if $A \in K$ and $a,b,c \in A$ are such that $\langle b \rangle$, $\langle c \rangle$ are infinite, and a, b, c do not appear in oneanother's subalgebras, then there exists $\phi \in Aut(A)$ for which $\phi(a) = a$ and $\phi(b) = c$; Therefore $b,c \in B$ can be exchanged, by an automorphism fixing a, in the pair of valid formulae above to reveal the sought for contradiction. Q.E.D.

4. DETERMINATENESS FOR LOCAL CLASSES

Let K be a class of algebras. An algebra A is *locally a K-algebra* if each finite subset of A is contained within a subalgebra of A which belongs to K; write L(K) for the class of all locally K-algebras.

PROPOSITION 4.1. *For any programs* P, Q *over any class* K,

$$P \equiv_{L(K)} Q \quad \textit{if, and only if,} \quad P \equiv_K Q.$$

PROOF. Now $P \equiv_{L(K)} Q$ implies $P \equiv_K Q$ because $L(K) \supset K$. Conversely, assume $P \equiv_K Q$. Let $A \in L(K)$ and consider an arbitrary computation of P, Q on $a \in A^n$. If $B \in K$ is a subalgebra of A containing (the components of) a then $P(a) \simeq Q(a)$ in A if, and only if, $P(a) \simeq Q(a)$ in B, by the Locality of Computation Lemma 0.1. So $P \equiv_K Q$ implies $P \equiv_{L(K)} Q$. Q.E.D.

PROPOSITION 4.2. *Let* K *be a first-order axiomatizable class and let* K_0 *be the class consisting of its countable structures. Then for any programs* P, Q *over* K

$$P \equiv_{K_0} Q \quad \textit{if, and only if,} \quad P \equiv_K Q.$$

Moreover, for any program P *over* K, $MPC_{K_0}(P) = MPC_K(P)$.

PROOF. Obviously, $P \equiv_K Q$ implies $P \equiv_{K_0} Q$ as $K_0 \subset K$. By Proposition 4.1, $P \equiv_{K_0} Q$ implies $P \equiv_{L(K_0)} Q$: we show $K \subset L(K_0)$. Let $A \in K$ and $a_1,\ldots,a_n \in A$. By a *Downward Löwenheim-Skolem Theorem* (for example, Theroem 3.1.6 in CHANG & KEISLER [5, p. 109]), there is a countable elementary substructure A_0 of A containing $a_1,\ldots,a_n$ which is a K-algebra as K is axiomatizable since $A_0 \in K_0$, $A \in L(K)$.

Now $MPC_K(P) \subset MPC_{K_0}(P)$ since $K_0 \subset K$. Assume for a contradiction that $(\alpha,\beta) \in MPC_{K_0}(P) - MPC_K(P)$. Then there exists $A \in K$ such that $A \not\models \alpha(x) \rightarrow (P(x)\downarrow \wedge \beta(x,P(x))) \vee P(x)$. So there is $a \in A^n$ such that $A \models \alpha(a)$ and $P(a)\downarrow$ but $A \not\models \beta(a,P(a))$. Since $P(a)\downarrow$ we can first-order express this computation and assert $A \models \phi_t^P(x) \rightarrow \neg\beta(x,\tau_t^P(x))$. Again by *Löwenheim-Skolem,* there is a countable elementary substructure A_0 of A so that $A_0 \models \phi_t^P(x) \rightarrow \neg\beta(x,\tau_t^P(x))$. From this it follows, from propositional manipulation and the locality of computations, that $(\alpha,\beta) \notin MPC_{K_0}(P)$, the required contradiction. Q.E.D.

We may now easily deduce this corollary.

THEOREM 4.3. *Let* K *be a first-order axiomatizable class and let* K_0 *be the class consisting of its countable structures. Then for all programs* P, Q *over* K *the following are equivalent:*

(i) $MPC_K(P) = MPC_K(Q)$ *implies* $P \equiv_K Q$

(ii) $MPC_{K_0}(P) = MPC_{K_0}(Q)$ *implies* $P \equiv_{K_0} Q$.

5. DETERMINATENESS AND COMPLETE AXIOMATIZABLE CLASSES

By a *complete axiomatizable class* K we mean that K is the class of all models of a complete first-order axiomatizable theory.

THEOREM 5.1. *Let* K *be a complete axiomatizable class and let* P, Q *be programs over* K. *Then the following properties are equivalent.*

(1) $MPC_K(P) = MPC_K(Q)$;

(2) *for some countable* $A \in K$, $P \equiv_A Q$;

(3) *for some countable* $A \in K$, $MPC_A(P) = MPC_A(Q)$.

PROOF. First we prove (1) implies (2). Now for $A \in K$, $P \equiv_A Q$ iff for *no* $a \in A^n$ any one of the following are true:

(i) for some t, s, $\phi_t^P(a) \wedge \phi_s^Q(a) \wedge \tau_t^P(a) \neq \tau_s^Q(a)$;

(ii) for some t, $\phi_t^P(a)$ and for all s, $\neg\phi_s^Q(a)$;

(iii) for some s, $\phi_s^Q(a)$ and for all t, $\neg\phi_t^P(a)$.

By now, the reader should recognize that case (i) is irrelevant for, in the presence of the hypothesis $MPC_K(P) = MPC_K(Q)$, when both P and Q converge their outputs must coincide. So we rephrase the situation thus:

Let $T_t^P = \{\phi_t^P(x), \neg\phi_s^Q(x): s \in \omega\}$ and $T_s^Q = \{\phi_s^Q(x), \neg\phi_t^P: t \in \omega\}$. Then for any $A \in K$, $P \equiv_A Q$ iff no $a \in A$ satisfies or *realizes one of the types* (to use the terminology of mathematical logic) T_t^P, T_s^Q $(t,s \in \omega)$. To prove (2) we look for some countable $A \in K$ which *omits* these types. Because K is complete and we can apply the *Extended Omitting Types Theorem,* CHANG & KEISLER [5, p. 82], it is sufficient to prove K locally omits all these types.

Suppose, for a contradiction, that T_t^P is locally realized, so there is a formula $\mathcal{O}$ consistent with K and such that $K \models \mathcal{O}(x) \to \phi_t^P(x)$ and $K \models \mathcal{O}(x) \to \neg\phi_s^Q(x)$ for all $s \in \omega$. We claim that $(\mathcal{O}(x), y \neq y) \in MPC_K(Q) - MPC_K(P)$; this is easy to see. Let $A \in K$, $a \in A^n$. If $A \models Q(a)$ then $A \models \Lambda_{s\in\omega} \neg\phi_s^Q(a)$ and $Q(a)\uparrow$. Hence $A \models \mathcal{O}(a) \to [Q(a)\downarrow \wedge Q(a) \neq Q(a)] \vee Q(a)\uparrow$. And $A \models \mathcal{O}(a)$ implies $A = \phi_t^P(a)$ and $P(a)\downarrow$ and so $(\mathcal{O}, y \neq y) \notin MPC_K(P)$.

Applying the same argument to T_s^Q shows all the types are locally omitted and the implication is proved.

That (2) implies (3) is immediate.

Thirdly, that (3) implies (1) follows from this fact:

LEMMA 5.2. *Let* P *be a program over the complete axiomatizable class* K. *Then for each* $A \in K$

$$MPC_A(P) = MPC_K(P)$$

PROOF. Since $\{A\} \subset K$, $MPC_K(P) \subset MPC_A(P)$. For the reverse inclusion, suppose, for a contradiction, that $(\alpha(x), \beta(x,y)) \in MPC_A(P) - MPC_K(P)$. Thus there exists $B \in K$ and $b \in B^n$ such that $B \models \alpha(b) \wedge P(b)\downarrow \wedge \neg\beta(b,P(b))$. Let $|P(b)| = t$ and set $\mathcal{O}(x) = \alpha(x) \wedge \phi_t^P(x) \wedge \neg\beta(x,\tau_t^P(x))$; clearly $B \models \mathcal{O}(b)$ and $B \models \exists x.\mathcal{O}(x)$. Since K is defined by a complete first-order axiomatic theory T, $T \vdash \exists x.\mathcal{O}(x)$ and $A \models T$ implies $A \models \exists x.\mathcal{O}(x)$. Whence it is easy to see that this contradicts $(\alpha,\beta) \in MPC_A(P)$. Q.E.D.

Determinateness for a complete axiomatizable class of K is not always possible (see Example A.4, Appendix 1), but the property in this case can

be neatly expressed in terms of a *logic of effective definitions*, LED, developed in [4] and [28], where it is equivalent to the class being $\forall$-LED complete.

By an *ω-categorial class* K we mean a class containing a countable structure and having the property that any two countable K-algebras are isomorphic.

COROLLARY 5.3. *Let* K *be an ω-categorial axiomatisable class. Then for any programs* P, Q *over* K,

$$MPC_K(P) = MPC_K(Q) \quad \textit{implies} \quad P \equiv_K Q.$$

PROOF. Assume $MPC_K(P) = MPC_K(Q)$. Since K is complete and axiomatizable (Proposition 3.1.10 CHANG & KEISLER [5, p. 113]) we can apply Theorem 5.1 to obtain a countable K-algebra A such that $P \equiv_A Q$. Let K_0 be the class of all K-algebras. Then since each structure in K is isomorphic to A, $P \equiv_{K_0} Q$. Whence by Proposition 4.2, $P \equiv_K Q$. Q.E.D.

6. DETERMINATENESS VIA EXTENDED SEMANTICS

Let $\Sigma_p = \{0,S,+,\cdot\}$ and let Σ be a second signature disjoint from Σ_P. Set $\Sigma_\omega = \Sigma \cup \Sigma_P$ and recall from §2 the meaning of a Σ_ω expansion of a Σ structure. A Σ_ω structure A is called a *standard expansion* if $A|_{\Sigma_P} = (\omega;0, +1,+,\cdot)$, the algebra of natural numbers with constant zero and operations successor, addition and multiplication.

If K is a class of Σ algebras then set $K(\Sigma_\omega)$ to be the class of all Σ_ω expansions of K algebras.

THEOREM 6.1. *Let* K *be a class of countable structures of signature* Σ. *Let* P, Q *be programs over* K. *Then*

$$MPC_{K(\Sigma_\omega)}(P) = MPC_{K(\Sigma_\omega)}(Q) \quad \textit{implies} \quad P \equiv_K Q.$$

PROOF. The first step is to formulate an arithmetization of the $\phi^P_t(x)$ in the first-order logic over Σ_ω. For $t \in \omega$, we denote by $\underline{t}$ the term $S^t(0)$ over Σ_ω.

REPRESENTATION LEMMA 6.2. *Let* $\{Q_t(x): t \in \omega\}$ *be a recursively enumerable sequence of open formulae of first-order logic over* Σ. *Then there exists a sentence* ψ *and a formula* $Q(y,x)$ *in the first-order language over* Σ_ω *such that* (i) ψ *is true in all standard* Σ_ω *structures and* (ii) *for each* $t \in \omega$, $\psi \vdash Q_t(x) \leftrightarrow Q(\underline{t},x)$.

We do not stop to prove this lemma as it is a reasonably straightforward adaptation of the usual proof of the representation of recursive functions in arithmetic.

Next, observe that $P \equiv_K Q$ iff $P \equiv_{K(\Sigma_\omega)} Q$ so it is sufficient to prove determinateness for $K(\Sigma_\omega)$. By the Localisation Lemma 2.4, it is sufficient to consider closed programs over finite extensions of Σ_ω by constants. Let P be a closed program over such a signature extension Γ and containing reference to only these constants new to Σ_ω and the constants and operations of Σ. Suppose $A \in K(\Gamma)$ and $A \models P\uparrow$. Without loss of generality assume A is standard: if A were not standard we can choose $B \in K(\Gamma)$ such that $B|_\Sigma = A|_\Sigma$ and $B|_{\Sigma_P}$ is standard arithmetic.

Since $A \models P\uparrow$ we have for all $t \in \omega$, $A \models \neg\phi_t^P$. Write $Q_t = \neg\phi_t^P$ and apply Lemma 6.2 so as to choose appropriate ψ and $Q(y)$. Thus, as A is standard, $A \models \psi$ and as $A \models Q_t(x) \leftrightarrow Q(\underline{t})$ we get $A \models Q(\underline{t})$, for each $t \in \omega$. Let $\mathcal{O} = \psi \wedge \forall t Q(t)$ a first-order formula over Γ. Again because A is standard we have $A \models \mathcal{O}$ so $\mathcal{O}$ is consistent.

Now suppose $B \in K(\Gamma)$ and $B \models \mathcal{O}$. Then $B \models \psi$ and for all $t \in \omega$, $B \models Q(\underline{t})$. Using $\psi \vdash \neg\phi_t^P \leftrightarrow Q(\underline{t})$ we deduce $B \models \bigwedge_{t\in\omega} \neg\phi_t^P$ which, of course, means $B \models P\uparrow$.

Q.E.D.

Notice that if K is any axiomatizable class then one obtains determinateness for K from $K(\Sigma_\omega)$ in view of Theorem 4.3.

7. DETERMINATENESS FOR ALG(Σ)

By an *algebraic signature* we mean a signature without relations. The purpose of this section is to prove this single theorem.

THEOREM 7.1. *Let* Σ *be an algebraic signature. Let* $K = ALG(\Sigma)$ *be the class of all structures of signature* Σ. *Then for all programs* P, Q *over* Σ,

$$MPC_K(P) = MPC_K(Q) \quad \textit{implies} \quad P \equiv_K Q.$$

Actually, our methods can be used to establish determinateness for ALG(Σ) for Σ possessed of relations *except* in the case of Σ containing any number of constants but one unary function and all its relations unary; in this exceptional case we conjecture there is no determinacy forthcoming from the modified theories.

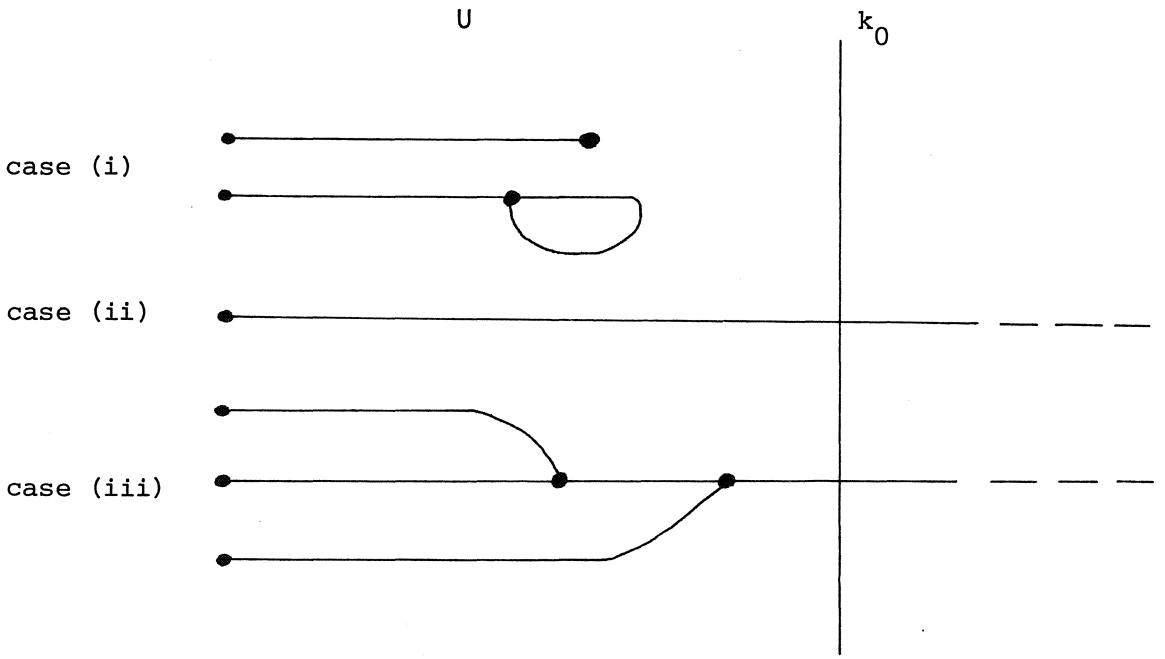
Meyer has announced [20] that he has obtained similar determinacy theorems for ALG(Σ) which are to appear in MEYER & HELPERN [21].

The argument for Theorem 7.1 is quite involved technically and we have decided to assume a greater measure of familiarity with mathematical logic on the part of the reader.

PROOF OF THEOREM 7.1. If Σ contains only constants then computations in K trivialize and determinateness is easily checked. The argument proper divides into the singular case, where Σ contains one unary function and some constants, and the general one of those algebraic signatures which remain. The theroem is proved by applying the Localisation Lemma 2.4 to the Lemmas 7.2 and 7.3 below.

LEMMA FOR SINGULAR CASE 7.2. *Let Σ be (a finite extension by constants of) an algebraic signature containing one unary function and let K be any class of Σ algebras which is closed under taking subalgebras. Then for any closed program P over Σ, if for some A ∈ K, A ⊨ P↑ then there is a sentence 0, first-order over Σ, which is consistent with K and such that K ⊨ 0 → P↑.*

PROOF. Assume A and $P = P(\underline{c}_1,\ldots,\underline{c}_n)$ are given and A ⊨ P↑. We make a special decomposition of the subalgebra $\langle c_1,\ldots,c_n\rangle$ of A. For f the unary function in Σ and c ∈ A, k ∈ ω define $orb_k(f,c) = \{a \in A: \exists i < k,\ f^i(c) = a\}$ and $orb(f,c) = \bigcup_{k\in\omega} orb_k(f,c)$. Then $\langle c_1,\ldots,c_n\rangle = orb(f,c_1) \cup \ldots \cup orb(f,c_n)$. There arises just a few possible types of *orbit* in this decomposition of interest to us, illustrated in the figure below: (i) $orb(f,c_i)$ is finite; (ii) $orb(f,c_i)$ is infinite and meets no other orbit; (iii) $orb(f,c_i)$ is infinite but intersects some $orb(f,c_j)$. In this third case notice that if $a \in orb(f,c_i) \cap orb(f,c_j)$ then $f^r(c_i) = a = f^s(c_j)$ for some r, s and hence for all k, $f^{r+k}(c_i) = f^{s+k}(c_j)$.



Choose k_0 so large as to bound the cardinalities of the finite orbits and the finite parts of intersecting orbits which remain distinct; set $U = U_{k\leq k_0}\ i\leq n\ arb_k(f,c_i)$. We aim to represent this subalgebra structure in a first-order sentence over Σ.

Let U be defined by the formula $U(x) = \vee_{i\leq k_0}\ \vee_{j\leq n}\ x = f^i(\underline{c}_j)$.

Let R define all equalities and inequalities in U in this way: set

$$T(i,j,p,q) = f^i(\underline{c}_p) = f^j(\underline{c}_q) \qquad \text{if } f^i(c_p) = f^j(c_q) \text{ in A}$$

$$= f^i(\underline{c}_p) \neq f^j(\underline{c}_q) \qquad \text{otherwise.}$$

Then

$$R = \bigwedge_{\substack{i,j\leq k_0\\ p,q\leq n}} T(i,j,p,q).$$

Let

$$S = (\forall x)[\neg U(x) \rightarrow f(x) \neq x \wedge \neg U(f(x)) \wedge (\forall y,z)(f(y) = f(z) = x \rightarrow y = z)]$$

And choosing those $c_{\lambda_1},\ldots,c_{\lambda_t}$ such that for all $a \in U$, $f(a) \neq c_{\lambda_i}$ $(1 \leq i \leq t)$ we define $V = \wedge_{i\leq t}\ (\forall x)[f(x) \neq \underline{c}_{\lambda_i}]$

Let $\mathcal{O} = R \wedge S \wedge V$. We claim $\mathcal{O}$ to be K-consistent and that $K \models \mathcal{O} \rightarrow P\uparrow$. The consistency of $\mathcal{O}$ follows from its construction from $\langle c_1,\ldots,c_n\rangle$ and the hypothesis that subalgebras of K-algebras are again K-algebras. To obtain $K \models \mathcal{O} \rightarrow P\uparrow$ one proceeds as follows. Let $B \in K$ and $B \models \mathcal{O}$. Let B' be the subalgebra of B generated by the elements named by the constants in P. One can now show that $B' \models \mathcal{O}$ implies B' is isomorphic to $\langle c_1,\ldots,c_n\rangle$ whence $P\uparrow$ on B', by Lemma 0.2, and so $B \models P\uparrow$, by Lemma 0.1. The proof of the isomorphism we leave to the reader. Q.E.D.

LEMMA FOR USUAL CASES 7.3. *Let Σ be (a finite extension by constants of) an algebraic signature containing at least two unary functions or at least function of arity greater than one and let* $K = ALG(\Sigma)$. *Then for any closed program* P *over* Σ, *if for some* $A \in K$, $A \models P\uparrow$ *then there is a sentence* $\mathcal{O}$, *first-order over* Σ, *which is consistent with* K *and such that* $K \models \mathcal{O} \rightarrow P\uparrow$.

PROOF. We begin with some general machinery which will handle the various possible signatures in a uniform way.

Let Γ be a one sorted signature. This we expand to a two sorted signature Γ_2 by adding to Γ a new sort called SETS, and renaming as DOM the sort of Γ, together with the binary relations ε on SETS $\times$ SETS and MAP on SETS $\times$ DOM. Given an algebra A of type Γ_2 we denote by $A|_{DOM}$ the reduct to the Γ structure of A and by $A|_{SETS}$ the reduct to the $\{SETS,\varepsilon\}$ structure of A.

Assume the first-order languages $L(\Gamma)$ and $L(\Gamma_2)$, over Γ and Γ_2 respectively, include only the connectives $\neg$, $\vee$, $\exists$; $L(\Gamma_2)$ has two kinds of variables x_i^D, x_i^S $(i \in \omega)$ ranging over DOM and SETS although we drop the superscripts whenever confusion seems unlikely.

Suppose we are given four formulae from $L(\Gamma)$, the list $t = Q_D(x), Q_S(x), Q_\varepsilon(x,y), Q_M(x,y)$. Such a list determines an interpretation H^t of $L(\Gamma_2)$ into $L(\Gamma)$ in an obvious way.

$$H^t(x_i^S) = x_{2i+1}$$

$$H^t(x_i^0) = x_{2i}$$

$$H(f(\tau_1,\ldots,\tau_k)) = f(H^t(\tau_1),\ldots,H^t(\tau_k))$$

$$H^t(x_i^S \in x_j^S) = Q_\varepsilon(H^t(x_i^S),H^t(x_j^S))$$

$$H^t(MAP(x_i^S,\tau)) = Q_M(H^t(x_i^S),M^t(\tau))$$

$$H^t(\phi \vee \psi) = H^t(\phi) \vee H^t(\psi)$$

$$H^t(\neg\phi) = \neg H^t(\phi)$$

$$H^t(\exists x_i^S \phi) = \exists x_{2i+1}(\phi\ (x_{2i+1}) \wedge H^t(\phi))$$

$$H^t(\exists x_i^D \phi) = \exists x_{2i}(Q_D(x_{2i}) \wedge H^t(\phi)$$

where f is a k-ary operation of Γ; $\tau,\tau_1,\ldots,\tau_k$ are Γ terms and ϕ, ψ are formulae of $L(\Gamma_2)$.

LEMMA 7.4. *Let $\mathcal{O}$ be a sentece of $L(\Gamma)$ and let H^t be an interpretation of $L(\Gamma_2)$ into $L(\Gamma)$ which together satisfy these two conditions:*

(i) *given any closed program P over Γ which diverges on some Γ structure, there exists a Γ structure A where $A \models \mathcal{O}$ and $A \models P\uparrow$.*

(ii) *for any sentence ψ of $L(\Gamma_2)$, whenever $\mathcal{O} \wedge \psi$ is consistent with respect to $ALG(\Gamma_2)$ then $H^t(\mathcal{O} \wedge \psi)$ is consistent with respect to $ALG(\Gamma)$.*

Then given any closed program P which diverges somewhere in $K = ALG(\Gamma)$, there exists a K-consistent sentence $\phi \in L(\Gamma)$ with $K \models \phi \to P\uparrow$.

PROOF. Let Γ_ω be Γ with the signature of arithmetic adjoined as in §6. Suppose $B \models \mathcal{O}$ and $B \models P\uparrow$ and let A be a countable elementary subalgebra of B so that $A \models \mathcal{O} \wedge P\uparrow$. Let A_ω be some standard Γ_ω expression of A. Then just as in the proof of Theorem 6.1 we can find a sentence $\phi' \in L(\Gamma_\omega)$ such that $A_\omega \models \phi'$ and $ALG(\Gamma_\omega) \models \phi' \to P\uparrow$.

We here state a technical lemma whose proof is a tedious exercise in axiomatic set theory which we take the liberty of omitting.

LEMMA 7.5. *Let* Δ *be a finite signature and let* δ *be a sentence of* $L(\Delta)$. *Then there is a sentence* p *of the first-order language of Zermelo-Fraenkel set theory,* L(ZF) *and a formula* p'(x) *of* L(ZF) *such that*

(i) $ZF \vdash p$

(ii) *if* $B \models p$ *then for* $b \in B$, $B \models p'(b)$ *iff* b *is a* Δ *structure which satisfies the sentence* δ.

To continue the proof of Lemma 7.4, set $\Delta = \Gamma_\omega$ and $\delta = Q'$ in Lemma 7.5 to get appropriate p and p'(x).

Let ψ be a sentence of $L(\Gamma_2)$ which expresses the following of a Γ_2 structure B: if p and $\exists x.p'(x)$ hold for B then for some b of type SET in B, p'(b) holds and MAP restricted to $b \times B|_{DOM}$ is the graph of a function $b \to B|_{DOM}$ which is a monomorphism of Γ structures.

We set $\phi = H(0 \wedge p \wedge \exists x.p'(x) \wedge \psi) \in L(\Gamma)$ and aim to show ϕ is K-consistent and $K \models \phi \to P\uparrow$. Consider the latter property first. Suppose $B \models \phi$. Define

$$SET(B) = \{b \in B: B \models Q_S(b)\}$$

$$DOM(B) = \{b \in B: B \models Q_D(b)\}$$

$$\varepsilon(B) = \{(b,b') \in B^2: B \models Q_\varepsilon(b,b')\}$$

$$MAP(B) = \{(b,b') \in B^2: B \models Q_S(b) \wedge Q_D(b') \wedge Q_M(b,b')\}$$

Notice that $DOM(B) \models 0$ and $SET(B) \models P \wedge \exists x.p'(x)$. Morover, let $B_{D,S}$ be the two sorted Γ_2 structure determined by these formulae by taking DOM(B) as a Γ structure and adding a disjoint copy S of SET(B) as a structure of sort SET. If $h: S \to SET(B)$ is the copying bijection then ε and MAP are defined: $a \in a' \iff B \models Q_\varepsilon(h(a),h(a'))$ and $MAP(a,a') \iff B \models Q_M(h(a),a')$. It should be clear from the nature of H^t that $B_{D,S} \models 0 \wedge p \wedge \exists x.p'(x) \wedge \psi$. We conclude that for some $b \in B_{D,S}|_{SET}$, b is a Γ_ω structure which satisfies ϕ' and that MAP restricted to $b \times DOM(B)$ (= domain of $B_{D,S}$) is the graph of an injective Γ homomorphism from b to $B_{D,S}|_{DOM}$ (= DOM(B)). As $b \models \phi'$ and $\models \phi' \to P\uparrow$ we have $b \models P\uparrow$. Now all values occurring in the computation P lie within the prime Γ subalgebra of b that is embedded in DOM(B). Therefore $DOM(B) \models P\uparrow$ and hence $B \models P\uparrow$ and we are done.

To prove consistency: let A be a countable Γ algebra such that $A \models \mathcal{O} \wedge P\uparrow$. Expand A to a standard Γ_ω structure A' with $A' \models \phi$ (remember any standard expansion of A will satisfy ϕ'). Now add to A' a model of ZF, which contains an isomorphic copy A" of A', to make the two sorted Γ_2 structure B wherein MAP is a relation which restricted to A" $\times$ A' is precisely the graph of a Γ isomorphism between them. It is easy to check that this Γ_2 structure B satisfies $\mathcal{O} \wedge p \wedge \exists x.p'(x) \wedge \psi$ and so it follows immediately from clause (ii) of Lemma 7.4 that $\phi = H^t(\mathcal{O} \wedge p \wedge \exists x.p'(x) \wedge \psi)$ is consistent. This completes the proof of Lemma 7.4. Q.E.D.

To complete the proposition's proof is a matter of defining interpretations H^t for the various signatures and proving true of them the two hypotheses of Lemma 7.4. We will give two representative cases:

(a) Σ contains one binary function f and constant c. Here take
$\mathcal{O} = (\exists x,y)[x \neq y \wedge \forall x.\exists y(f(y,y) = x)]$ and

$$Q_S(x) \equiv \neg\exists y.f(y,y) = x$$

$$Q_D(x) \equiv \exists y.f(y,y) = x$$

$$Q_\varepsilon(x,y) \equiv x \neq y \wedge f(x,y) = c$$

$$Q_M(x,y) \equiv x = f(y,y).$$

(b) Σ contains two unary functions f, g. Here take
$\mathcal{O} = \forall x\exists y.f(y) = x$. Let $Q_c(x) = \neg\exists y.f(y) = x$ and define

$$Q_S(x) \equiv \exists y(Q_c(y) \wedge f(y) = x)$$

$$Q_D(x) \equiv \neg Q_c(x) \wedge \neg Q_S(x)$$

$$Q_\varepsilon(x,y) \equiv \exists z.(Q_c(z) \wedge f(z) = x \wedge g(z) = y)$$

$$Q_M(x,y) \equiv f(x) = y$$

8. ARITHMETIC PROGRAMS

We take as the standard model of arithmetic the algebra $N = (\omega;0,+1,+,\cdot,\leq)$ whose signature we denote Σ; We shall consider programs over Σ applied to certain types of models of *Peano arithmetic* over Σ. Let PA denote the class of *all* models of Peano arithmetic. The following fact is easy to see:

LEMMA 8.1. *For any programs* P, Q *over* Σ,

$$MPC_{PA}(P) = MPC_{PA}(Q) \quad \textit{implies} \quad P \equiv_N Q.$$

The question of determinateness for PA,

$$MPC_{PA}(P) = MPC_{PA}(Q) \quad \text{implies} \quad P \equiv_{PA} Q$$

we cannot yet answer, and we offer it as an open problem. We can provide, however, the following theorem.

Let $\Pi_1(N)$ be the set of all *universal* first-order sentences over Σ true in the standard model N. Let K be the class of all models of the theory defined by Peano arithmetic plus $\Pi_1(N)$.

THEROEM 8.2. *For any programs* P, Q *over* Σ

$$MPC_{PA}(P) = MPC_{PA}(Q) \quad \textit{implies} \quad P \equiv_K Q.$$

Before proving Theorem 8.2 we prove an interesting observation which illustrates that quite basic information about computation on N can be read off from information about computations on K. In particular, this next theorem shows that *programs on* N *equivalent up to any denotational semantics determined by* N *can be detected as operationally distinct on* N *from their denotational inequivalence on* K.

PROPOSITION 8.3. *Let* P, Q *be programs over* Σ. *Suppose that* $P \equiv_N Q$ *but* $P \not\equiv_K Q$. *Then the relative run times of* P *and* Q *over* N *are unbounded in the sense that for any* k *there exists an input* $a \in N^n$ *such that*

$$\frac{|P(a)|}{|Q(a)|} + \frac{|Q(a)|}{|P(a)|} > k.$$

PROOF. First suppose that for some $A \in K$, $a \in A^n$ it is the case that $P(a)\downarrow$, $Q(a)\downarrow$ but $P(a) = Q(a)$. Let $|P(a)| = t$ and $|Q(a)| = s$ so that $A \models \phi^P_t(a) \wedge \phi^Q_s(a) \wedge \tau^P_t(a) \neq \tau^Q_s(a)$. Now since all of $\Pi_1(N)$ is satisfied in A, $N \models \exists x.\phi^P_t(x) \wedge \phi^Q_s(x) \wedge \tau^P_t(x) \neq \tau^Q_s(x)$ from whence it follows P and Q differ somewhere on N; this contradicts $P \equiv_N Q$. So we amy assume that for some $A \in K$, $a \in A^n$ it is the case that $P(a)\downarrow$ but $Q(a)\uparrow$ (say).

Let $\theta^s_t(x) = \phi^P_t(x) \wedge \Lambda_{i \leq S} \neg\phi^Q_i(x)$. If again $|P(a)| = t$ then for each s, $A \models \exists x.\theta^s_t(x)$.

As this is an existential sentence and $A \models \Pi_1(N)$ we deduce $N \models \exists x.\theta^s_t(x)$ for each s. Given k choose any s, t such that $s > tk$ and choose $a \in N^n$ such that $N \models \theta^s_t(a)$. Then

$$\frac{|P(a)|}{|Q(a)|} + \frac{|Q(a)|}{|P(a)|} \geq \frac{|Q(a)|}{|P(a)|} \geq \frac{s}{t} > \frac{tk}{t} = k. \qquad \text{Q.E.D.}$$

PROOF OF THEOREM 8.2. Contrapositively, assume $P \not\equiv_K Q$. If $P \not\equiv_N Q$ then we are done because of Lemma 8.1; so assume $P \equiv_N Q$, the hypotheses of Proposition 8.3. From the proof of 8.3 we can further assume that somewhere in K, P converges whilst Q diverges and, moreover, we can choose t such that for all s, $N \models \exists x.\theta^s_t(x)$ where θ^s_t is as defined in the argument of 8.3.

Let $\phi(z)$ be a formula such that $\exists z.\phi(z)$ is satisfied somewhere in PA and for any $M \in PA$, $a \in M$ if $M \models \phi(a)$ then a is a non-standard element of M; this exists by *Gödel's Incompleteness Theorem.*

There are now two cases to the proof, one of which must hold since $\exists z.\phi(z)$ is consistent with PA.

(1) $\quad \exists z.(\phi(z) \wedge \exists x[\phi^P_t(x) \wedge (\forall y < x).\neg\phi^Q(y,x)])$ $\quad$ is satisfied in PA.

Then we claim

$$(\exists z(\phi(z) \wedge \phi^P_t(x) \wedge (\forall y < z).\neg\phi^Q(y,x)),\ y \neq y) \in MPC_{PA}(Q) - MPC_{PA}(P).$$

To see the pair is in $MPC_{PA}(Q)$ is to notice the precondition can be satisfied in which case it implies for all standard k, $\neg\phi^Q(k,x) \equiv \neg\phi^Q_k(x)$ and so $Q(x)\uparrow$. Whereas to see the pair does not lie in $MPC_{PA}(P)$ is to note the precondition implies the convergence of P(x).

(2) $\forall z.(\phi(z) \rightarrow \forall x[\phi_t^P(x) \rightarrow \forall y < z.\phi^Q(y,x)])$ is satisfied in PA.

Let H(x) stand for the least y such that $\phi^Q(y,x)$ if any such exist. Assume $\phi(z)$ holds then $\phi_t^P(x) \rightarrow H(x) \leq z$ and $\sup\{H(x): \phi_t^P(x)\}$ exists; let this be defined by the formula $\gamma_t(w)$. Set $\psi(x) = \exists z.\phi(z) \wedge \exists w.(\gamma_t(w) \wedge (\forall y < w).\neg\phi^Q(y,x))$. We claim $(\psi(x), y \neq y) \in MPC_{PA}(Q) - MPC_{PA}(P)$.

To see the pair lies in $MPC_{PA}(Q)$ notice that $\gamma_t(w)$ implies w exceeds the lengths of all computations of Q under the condition $\phi_t^P(x)$; in particular w exceeds all standard numbers as these computations may have arbitrarily large standard lengths or standard inputs. It follows that for all standard k, $\neg\phi^Q(k,x) \equiv \phi_k^Q(x)$ which entails $Q(x)\uparrow$.

On the other hand, the pair fails to lie in $MPC_{PA}(P)$ because of the consistency of $\exists x.\psi(x)$ and the fact that $\psi(x)$ implies the convergence of P(x). Q.E.D.

REFERENCES

[1] APT, K.R., *Ten years of Hoare's logic, a survey,* pp. 1-44 of F.V. Jensen, B.H. Mayoh & K.K. Møller (eds.), *Proceedings from 5th Scandinavian Logic Symposium,* Aalborg University Press, Aalborg, 1979.

[2] DE BAKKER, J.W., *Recursive procedures,* Mathematical Centre Tracts 24, Mathematical Centre, Amsterdam, 1973.

[3] _____, *Mathematical theory of program correctness,* to appear.

[4] BERGSTRA, J.A. & J. TIURYN, *Implicit definability of algebraic structures by means of program properties,* to appear in *Fundamentals of Computation Theory, 1979.*

[5] CHANG, C.C. & H.J. KEISLER, *Model theory,* North-Holland, Amsterdam, 1973.

[6] DIJKSTRA, E.W., *A discipline of programming,* Prentice-Hall, Englewood Cliffs, New Jersey, 1976.

[7] FLOYD, R.W., *Assigning meaning to programs,* pp. 19-32 of J.T. Schwartz (ed.), *Mathematical aspects of computer science,* American Mathematical Society, Providence, Rhode Island, 1967.

[8] FRIEDMAN, H., *Algorithmic procedures, generalized Turing algorithms, and elementary recursion theory*, pp. 316-389 of R.O. Gandy & C.M.E. Yates (eds.), *Logic colloquium, '69*, North-Holland, Amsterdam, 1971.

[9] GREIBACH, S.A., *Theory of program structures: schemes, semantics, verification*, Springer-Verlag, Berlin, 1975.

[10] GREIF, I. & A.R. MEYER, *Specifying the semantics of while-programs: a tutorial and critique of a paper by Hoare and Lauer*, Laboratory for Computer Science Technical Report, M.I.T., Cambridge, 1979.

[11] _____, *Can partial correctness assertions specify programming language semantics?*, pp. 25-26 of K. Weihrauch (ed.), *Theoretical Computer Science: 4th GI Conference*, Springer-Verlag, Berlin, 1979.

[12] HOARE, C.A.R., *An axiomatic basis for computer programming*, Communications ACM 12 (1967), 576-580.

[13] _____, *Procedures and parameters: an axiomatic approach*, pp. 102-116 of E. Engeler (ed.), *Symposium on the semantics of algorithmic languages*, Springer-Verlag, Berlin, 1971.

[14] HOARE, C.A.R. & P. LAUER, *Consistent and complementary formal theories of the semantics of programming languages*, Acta Informatica 3 (1974), 135-155.

[15] HOARE, C.A.R. & N. WIRTH, *An axiomatic definition of the programming language PASCAL*, Acta Informatica 2 (1973), 335-355.

[16] MAL'CEV, A.I., *Algebraic systems*, Springer-Verlag, Berlin, 1973.

[17] MANNA, Z., *The correctness of programs*, Journal of Computer and System Sciences 3 (1969), 119-127.

[18] _____, *Mathematical theory of computation*, McGraw-Hill, New York, 1974.

[19] MEYER, A.R., Letter to J. Tiuryn, 6th April 1979.

[20] _____, Letter to J. Tiuryn, 16th May 1979.

[21] MEYER, A.R. & J. HELPERN, *Specifying programming language semantics with before-after assertions,* to appear.

[22] MOLDESTAD, J., V. STOLTENBERG-HANSEN & J.V. TUCKER, *Finite algorithmic procedures and inductive definability,* to appear in Mathematica Scandanavia.

[23] _____, *Finite algorithmic procedures and computation theories,* to appear in Mathematica Scandanavia.

[24] MOLDESTAD, J. & J.V. TUCKER, *On the classification of computable functions in an abstract setting,* in preparation.

[25] PARIKH, R., *Some applications of topology to program semantics,* to appear in Mathematical Systems Theory.

[26] SABELFELD, V.K., *Äquialente Transformationen für Flussdiagramme,* Acta Informatica, 10 (1978), 127-155.

[27] SHEPHERDSON, J.C., *Computation over abstract structures: serial and parallel procedures and Friedman's effective definitional schemes,* pp. 445-513 of H.E. Rose & J.C. Shepherdson (eds.), *Logic colloquium '73,* North-Holland, Amsterdam, 1975.

[28] TIURYN, J., *Algebraic aspects of logic based on term algebras of r.e. trees,* Warsaw University Research Report, Warsaw, 1978.

[29] TUCKER, J.V., *Computing in algebraic systems,* Matematisk institutt, Universitetet i Oslo, Preprint Series, No. 12 (ISBN 82-553-0358-8), Oslo, 1978.

APPENDIX I

Here we shall illustrate the complicated behaviour of the modified correctness theories for a fixed pair of programs over various classes. Starting with a signature Σ_1 containing the unary function symbol g and constant 0, consider the programs P, Q abbreviated

$$P(x) \equiv 0$$

$$Q(x) \equiv \underline{\text{while}}\ x \neq 0\ \underline{\text{do}}\ x := g(x)\ \underline{\text{od}}$$

where we think of g as a predecessor function. Obviously, for any class K of systems of type Σ_1 we have $MPC_K(P) \subset MPC_K(Q)$; on the other hand in many K, $P \not\equiv_K Q$ because Q need not always terminate. Let $K_1 = ALG(\Sigma_1)$ the entire species of systems of type Σ_1.

(A1) $\quad MPC_{K_1}(P) \neq MPC_{K_1}(Q).$

PROOF. Let $\alpha(x) \equiv x \neq 0 \wedge g(x) = x$ and $\beta(x,y) \equiv 0 \neq 0$. Then $(\alpha,\beta) \in MPC_{K_1}(Q) - MPC_{K_1}(P)$ because $K_1 \models \alpha(x) \rightarrow Q(x)\uparrow$. Q.E.D.

Extend Σ_1 to Σ_2 by adding a unary function symbol f and set K_2 to be the class of systems of type Σ_2 satisfying

$$g(0) = 0$$

$$\forall x(gf(x) = x)$$

$$\forall x(x \neq 0 \rightarrow fg(x) = x)$$

$$\forall x(x \neq 0 \rightarrow f(x) \neq x \wedge g(x) \neq x)$$

$$\forall x(f(x) \neq 0)$$

So we think of f in the rôle of successor.

(A2) $\quad MPC_{K_2}(P) = MPC_{K_2}(Q).$

PROOF. Use the proof of Theorem 3.1.

Extend Σ_2 to Σ_3 by adding symbols for the binary functions and symbols $-$, $+$, $\cdot$, $\leq$ and set K_3 to be the class of systems of type Σ_3 which satisfy Peano's axioms for arithmetic.

(A3) $\quad MPC_{K_3}(P) \neq MPC_{K_3}(Q).$

PROOF. The programs P and Q differ on precisely the non-standard number systems of K_3. By *Godel's Incompleteness Theorem,* there is a formula $\alpha(x)$, consistent with Peano arithmetic - the class of K_3 - such that $K_3 \models \alpha(x) \rightarrow$ "x is non-standard". Let $\beta(x,y) \equiv 0 \neq 0$. It follows that

$$K_3 \models \alpha(x) \rightarrow [(Q(x)\downarrow \wedge \beta(x,Q(x))) \vee Q(x)\uparrow]$$

but

$$K_3 \not\models \alpha(x) \rightarrow [(P(x)\downarrow \wedge \beta(x,P(x))) \vee P(x)\uparrow]. \qquad Q.E.D.$$

Let K_4 be the subclass of K_3 of all structures elementary equivalent to the standard model of Peano arithmetic, N.

(A4) $\quad MPC_{K_4}(P) = MPC_{K_4}(Q).$

PROOF. Suppose not; it is easily seen that there is a formula $\alpha(x)$, consistent with $T = Th(N)$ such that $T \models \alpha(x) \rightarrow Q(x)\uparrow$. As T is complete, $T \vdash \exists x.\alpha(x)$ and $N \models \exists x.\alpha(x)$ and so there is $n \in N$ such that $Q(n)\uparrow$; by definition of Q this cannot be the case. Q.E.D.

Notice that as T is a complete theory, K_4 satisfies the hypotheses of Theorem 5.1.

Finally, extend the signature Σ_4 to Σ_5 by adding a binary relation symbol R and let K_5 be the class of all structures of K_4 augmented by arbitrarily chosen binary relations (which are to interpret R). As still $P \not\equiv_{K_5} Q$, Theorem 5.1 proves that

(A5) $\quad MPC_{K_5}(P) \neq MPC_{K_5}(Q).$

We think it a useful task to construct an analysis of such *bad* phenomena as the alternations A.1 - A.5 in general algebraic terms with a view to understanding the regularities involved in non-termination of programs.