1972-02-00 A

INSTITUT FÜR NUMERISCHE MATHEMATIK UND
ELEKTRONISCHE INFORMATIONSVERARBEITUNG
UNIVERSITÄT INNSBRUCK

A basic problem in the theory of programming languages

Bruno Buchberger

Bericht Nr. 72 - 1 Februar 1972

A- 6020 Innsbruck, Innrain 52

SUMMARY

We give an exact formulation of a basic problem in the theory of programming languages, which, on an intuitive level, would read as follows: Find conditions on the instruction set of a language such that all partial recursive functions can be expressed in that language. We derive two theorems which may be regarded as a first step in a thorough study of this problem.

Notation and abbreviations:

N ... set of all natural numbers including o.

p.r.f. ... partial recursive function(s).

P_n ... set of all n-ary p.r.f.

R_n ... set of all n-ary recursive (i. e. partial recursive, total) functions.

"Gödel numbering" ... Gödel numbering of the unary p.r.f. (Rogers [12]).

If f,g, denote functions, fg(x) often stands for f(g(x)).

If f denotes a function, then $f^{(t)}(x)$ is defined as follows:

$$f^{(o)}(x) = x,$$

$$f^{(t+1)}(x) = f\, f^{(t)}(x) \quad (\text{for } t \geq o).$$

$|C| < \infty$ ... C is a finite set.

$|C| = \infty$ ... C is infinite and denumerable.

$\wedge$, $\vee$, $\neg$, $\rightarrow$, (x), (Ex) ... conjunction, disjunction, negation, implication, universal quantification of x, existential quantification of x.

Quantification is always over N.

σ_n, σ_{n1}, ..., σ_{nn} ... recursive "pairing" functions for the n-tuples of natural numbers.

1. Informal preliminaries

In this section we want to give a detailed exposition of our intuitive concept on which the formal study is based. The reader who is not interested in formal considerations may well skip this section, except Definition 1.1, and immediately continue with section 2.

First, we adopt the point of view that the theory of programming languages should be identified or at least be a part of the theory of Gödel numberings (Rogers [12]), since the fundamental requirements for Gödel numberings, i. e. the recursive "translatability" between these numberings and one of the historical "standard" numberings, is a property we would not like programming languages to dispense from. Also all the consequences of this fundamental requirement (for instance, the recursiveness of certain operations on p.r.f. etc.), when interpreted for programming languages, fit our intuition very well.

However, it turned out that it is necessary to add some basic notions to the general theory of Gödel numberings for studying some special properties of "programs". So, by adding the notion of a "complexity measure", Blum [1] derived a big number of interesting theorems about the complexity of computations. Since there are complexity measures for all Gödel numberings, the addition of this notion, on the other hand, does not restrict the class of numberings considered.

Similarly, we would like to add the notion of a "P-decomposition" of a Gödel numbering to derive theorems on other aspects of programming

languages. Again, this will not restrict the applicability of the theory since there exist P-decompositions for every Gödel numbering (Proposition 4.3).

We shall, first, give an exact definition what we mean by a "P-decomposition" of a p.r.f. and, later, make some remarks to explain why we think that this notion could have some importance for Gödel numberings (programming languages, computability formalisms, computer concepts etc.).

Let $\{\rho_i\}$ be some Gödel numbering of the unary p.r.f., $\Psi \in P_2$ the function that "describes" $\{\rho_i\}$ (see Rogers [12], p.332), i. e.

(1.1) $(i,x)(\Psi(i,x) = \rho_i(x))$.

(We sometimes also shall speak simply about the "Gödel numbering Ψ ").

Further, we introduce the operator of "conditioned iteration" P, which for all functions $f,g \in P_1$ yields a function $[f,g]^P \in P_1$ by the following recursive definition

$$(1.2) \qquad [f,g]^P(x) = \begin{cases} x, & \text{if } g(x) = 0 \\ [f,g]^P f(x) & \text{otherwise.} \end{cases}$$

Definition 1.1: Let $\Psi \in P_2$. If $\gamma \in R_2$, $\bar{\Psi}, \kappa, \rho \in R_1$, and

(1.3) $(i,x)\ (\Psi(i,x) = \rho [\bar{\Psi},\kappa]^P \gamma(i,x))$,

then we say that " ρ, $\bar{\Psi}$, κ, γ determine a P-decomposition of Ψ".

Remarks:

1. Discussion of the operator P: Given some $x \in N$, $[f,g]^P(x)$ is evaluated by successive evaluation of x, $f(x)$, $f^{(2)}(x)$, ... until, for some t, $gf^{(t)}(x) = o$. In this case $[f,g]^P(x) = f^{(t)}(x)$. If no such t can be found (this is the case also if some of the $f^{(t)}(x)$, $gf^{(t)}(x)$ which appear during the evaluation of $[f,g]^P(x)$ are undefined), $[f,g]^P(x)$ is undefined.

   If f is the state transition function of some automaton A (with countably many states) and g its "termination criterion" (i.e. "$g(x) = o$" is interpreted as "x is a terminal state") then $[f,g]^P$ is the function which, for any state x, defines the terminal state which A, eventually, reaches if it starts a computation with x. Also, by a suitable definition of the notion "state" every type of deterministic automaton can be viewed as a special case of this general type, which is fully defined by the two functions f,g.

2. Reviewing the entire literature on computability formalisms (Turing's formalism, Kleene's formalism, Markov's formalism etc., including the literature on special programming languages and computers) one can easily check that without any exception all these formalisms were introduced according to the following scheme:

   a. One defines what a momentary state of a computation in the alleged formalism F (an "F-state") should be. In Turing's formalism this is, for instance, the quadruple [Turing table, tape expression, internal configuration, scanned square]. Notice that, in this example, we let the Turing table (the "program") be a part of the "state".

This is necessary since, with other formalisms (for instance machine codes of computers), it is not "objectively" possible to separate the "program" from "the rest of the state".

The set X_F of all F-states always is an (intuitively) decidable set, which can be brought into an (intuitively) effective 1-1 correspondence with N. Hence, without any loss of generality, we simply set $X_F = N$ (cf. Rogers [12], p.332).

b. Given any F-state $x \in N$ one says what is done in the next step of an "F-computation" to yield the "F-successor state of x", i. e. one defines a function

(1.4) $$\begin{array}{ccc} \bar{\Psi}_F : & N & \longrightarrow N \\ & \cup & \cup \\ & x & \longmapsto x' \\ & \text{"state"} & \text{"F-successor state"}. \end{array}$$

$\bar{\Psi}_F$ always is total (or at least can be made to be total without changing the formalism), and (intuitively) computable. Thus, one might say that $\bar{\Psi}_F \in R_1$ for all the formalisms in the literature. (In the historical cases, most $\bar{\Psi}_F$ were given in natural languages. Only when defining "universal" elements in the formalism F, the various $\bar{\Psi}_F$ were given formally).

c. One defines an (intuitively) decidable predicate "x is a terminal state in the formalism F", i. e., essentially, one defines a function $\kappa_F \in R_1$ with the interpretation "$\kappa_F(x) = o$" means "x is F-terminal".

d. By the following rule (which is the same for all formalisms) it is said how an F-computation, eventually, defines a resulting terminal state $\psi_F^*(x)$ for every state $x \in N$: "Successively apply $\bar{\psi}_F$ to x until for a certain t $\kappa_F\, \bar{\psi}_F^{(t)}(x) = o$". This is exactly what could also be written

$$(1.5) \qquad \psi_F^*(x) = [\bar{\psi}_F, \kappa_F]^P(x).$$

e. The functions $\bar{\psi}_F$, κ_F are characteristic for a given formalism F. For defining Gödel numberings by means of such a formalism one, usually, proceeds as follows. One defines an (intuitively) computable, total "input" function $\gamma(i,x)$ and an (intuitively) computable, total output function ρ (i. e. using Church's thesis: $\gamma \in R_2$, $\rho \in R_1$).

By defining

$$(1.6) \qquad \mathcal{F}_F^1 := \{ f \mid (Ei)(f(x) = \rho\, [\bar{\psi}_F, \kappa_F]^P\, \gamma(i,x)) \},$$

$$\mathcal{F}_F^n := \{ f \mid (Ei)(f(x_1,\ldots,x_n) = \rho[\bar{\psi}_F, \kappa_F]^P\, \gamma(i, \sigma_n(x_1,\ldots,x_n))) \},$$

for $n \geq 2$,

$$\mathcal{F}_F := \bigcup_{n=o}^{\infty} \mathcal{F}_F^n,$$

one, finally, is in a position to have a class $\mathcal{F}_F$ of functions which can be proposed to be an exact substitute for the informal notion "the class of (intuitively) computable functions".

If $\bar{\Psi}_F$, κ_F, γ, ρ are chosen in a skillful way one will be able to prove that $\mathcal{F}_F$ coincides, for instance, with the class of Turing-computable functions, and what is more, to show that Ψ , defined by

$$\Psi(i,x) := \rho\left[\bar{\Psi}_F, \kappa_F\right]^P \gamma(i,x), \tag{1.7}$$

is a Gödel numbering.

Of course, if $\bar{\Psi}_F$, κ_F together with ρ, γ are such that (1.7) defines a Gödel numbering (i. e. ρ, $\bar{\Psi}_F$, κ_F, γ determine a P-decomposition of a Gödel numbering Ψ), then for any other Gödel numbering $\Psi' \in P_2$ one can find a $\gamma' \in R_2$ such that ρ, $\bar{\Psi}_F$, κ_F, γ' determine a P-decomposition of Ψ'. To show this we choose $\tau \in R_1$ such that

$$(i,x)(\Psi'(i,x) = \Psi(\tau(i),x)) \tag{1.8}$$

(Rogers [12]). Hence,

$$\Psi'(i,x) = \rho\left[\bar{\Psi}_F, \kappa_F\right]^P \gamma(\tau(i),x) = \rho\left[\bar{\Psi}_F, \kappa_F\right]^P \gamma'(i,x),$$

with $\gamma'(i,x) = \gamma(\tau(i),x)$.

Let us summarize the analysis of the computability formalisms given above by stating that:

A. Concrete Gödel numberings Ψ until now (and, presumably, in the next future) have been (will be) given by constructing a P-decomposition of Ψ : The P-operator seems to be the kind of recursion which is most "natural" to human brain and "therefore" appears on the meta level of the definitions for the notion

"computability", which, essentially, are definitions of the notion "recursion" on some object level.

B. A computability formalism F (programming language, computer concept etc.) is characterized by the functions $\bar{\Psi}_F$, κ_F and not by the Gödel numbering definable by $\bar{\Psi}_F$, κ_F since we saw that one and the same $\bar{\Psi}_F$, κ_F can define all Gödel numberings by means of suitable "input" and "output" functions.

A. and B. seem to be sufficient to make a systematic study of P-decompositions of Gödel numberings worthwhile.

3. There is yet another reason why P-decompositions of Gödel numberings seem to be important. It is the fact that Gödel numberings ("programming languages") Ψ, usually, are "implemented" by constructing a P-decomposition of Ψ by means of some automaton (computer), which can be given by two functions $\bar{\Psi}$, $\kappa \in R_1$ (Remark 1). For every program i and data x, one defines an initial state $x = \gamma(i,x)$ ($\gamma \in R_2$) from which the computation starts. After eventual termination a function $\rho \in R_1$ defines the "result". Thus, implementing a Gödel numbering Ψ, usually, means to realize a P-decomposition

$$\Psi(i,x) = \rho\left[\bar{\Psi}, \kappa\right]^P \gamma(i,x). \tag{1.9}$$

The assumption that γ, ρ are recursive seems to be quite sound. Requiring only $\gamma \in P_2$ and not $\gamma \in R_2$ would not exclude γ to be, for instance, a Gödel numbering itself, in which case the "real computa-

tional work" would already be done by the input function γ. Similarly, dispensing ρ from being recursive would admit computations that terminate, but don't yield any resulting value. On the other hand, for the beginning it seems to be difficult to lay further reasonable restrictions on ρ, γ without being totally arbitrary or excluding interesting cases that occur in practice.

4. It is the aim of this report to begin a study of P-decompositions of Gödel numberings under these very general assumptions about the functions ρ, $\bar{\Psi}$, κ, γ to have a theoretical framework for existing and non-existing studies on special P-decompositions or classes of P-decompositions defined by assuming ρ, $\bar{\Psi}$, κ, γ to have more restrictive properties.

2. A basic problem

We now are in a position to give an exact formulation of a basic problem in the theory of programming languages which, on an intuitive level, would read as follows:

Problem 2.1: What makes a given computability formalism (programming language, computer concept etc.) equally "strong" as Turing's formalism?

Though much work has been done on showing special formalisms to have the same expressive power as Turing's (or some other standard) formalism, very little is known about this general problem. In terms of P-decompositions this problem can be stated in the following way:

Problem 2.2: Give necessary and sufficient conditions for ρ, $\bar{\Psi}$, $\kappa \in R_1$ and $\gamma \in R_2$ to determine a P-decomposition of a Gödel numbering.

Here, to have conditions on $\bar{\Psi}$, κ would be the most interesting since, on an intuitive level, this would give an answer to the general question: What power must be contained in the instruction set of a programming language (which essentially defines the $\bar{\Psi}$) in order that the whole formalism has the power of the standard formalisms?

From Rice' theorem (Rice [11]) it is clear that such conditions can't be decidable, at least if we represent ρ, $\bar{\Psi}$, κ, γ themselves in some Gödel numbering. Instead of requiring the decidability of alleged (sufficient) conditions on ρ, $\bar{\Psi}$, κ, γ we shall somewhat vaguely require that they should

be "easy to check" at least in the sense, that it should be easier to check them than to write a "translator" from one of the standard formalisms to the numbering determined by $\rho[\bar{\Psi},\kappa]^P \gamma(i,x)$.

We hope that the considerations of section 1 are a convincing argument for showing that our Problem 2.2 is an adequate precise substitute for the somewhat vague Problem 2.1.

3. Literature

In this section, we shall briefly discuss four papers which seem to have some relevance to our Problem 2.2.

a) Markov [9].

Let $\Psi_T(i,x)$ be the Gödel numbering defined by Kleene's normal form theorem (Davis [6] p. 63). Let us, further, define

Definition 3.1: A function U is of large oscillation if

$$(3.1) \qquad (y,z)(Ex)(x > z \wedge U(x) = y).$$

In [9] Markov, essentially, proves the fact that in a representation

$$(3.2) \qquad \Psi_T(i,x) = U(\mu_y\, T(i,x,y)) \qquad (T \ldots \text{primitive recursive predicate})$$

the possible "output" functions U are exactly the functions of large oscillation. It is claer that this gives the solution to a problem which is similar to our Problem 2.2. However,

1. Markov's theorem concerns only the "butput" function U. What seems to be more interesting, is a condition on the necessary and sufficient computational power contained in the predicate T.

2. The result on U cannot be carried over to the "output" function of a P-decomposition of a Gödel numbering (see p. 35).

b. Davis [4,5].

Davis starts form the observation that the universality of a Turing machine depends on the encoding of programs and data and attempts to give a definition of "universal Turing machine" which explicitly specifies the admissable codes. He, eventually, arrives at the following definition (Davis [5]) which we shall put down in our terminilogy:

Definition 3.2: A Turing machine M is called universal if there exist functions $\rho \in R_1$ and $\gamma \in R_2$, such that

$$\Psi_T(i,x) = \rho\, [\bar{\Psi}_M,\ \kappa_M]^P\, \gamma(i,x). \tag{3.3}$$

Here, $\Psi_T(i,x)$ is, again, the Gödel numbering of Kleene's normal form theorem (Davis [6], p. 63). $\bar{\Psi}_M$, κ_M are the transition function respectively the termination criterion of M after some fixed Gödelization of the instantaneous descriptions (Davis [6], p. 6). Davis's main result is

Theorem 3.1: If M is a universal Turing machine then the set $C_M := \{ x \mid [\bar{\Psi}_M,\ \kappa_M]^P(x)$ is defined $\}$, i. e. the set of all instantaneous descriptions that start a terminating computation of M, is a complete set.

The notion "complete set" is given by the following

Definition 3.3: $C \subset N$ is a complete set if there is a function $f \in R_2$ such that for all i

(3.4) $\quad U_i := \{ x \mid \Psi_T(i,x) \text{ is defined} \} = \{ x \mid f(i,x) \in C \}.$

It is easy to see that the converse of Theorem 3.1 is not true. Davis's approach is only a partial step towards a satisfactory solution of our Problem 2.2 because

1. The functions $\bar{\Psi}_M$, κ_M admitted are of a very special kind: they must be the transition function, respectively the termination criterion of Turing machines. We would like to have propositions about general $\bar{\Psi}$, $\kappa \in R_1$.

2. The content of Theorem 3.1 is only a necessary condition. We would like to have sufficient conditions, too, and best, conditions which are necessary and sufficient.

3. There is no independent analysis of $\bar{\Psi}$ and κ . The condition only says something about the combined function $\Psi^* = [\bar{\Psi}, \kappa]^P$. That is, we get no information about the machine's necessary "power" at one step (about the power of the "instruction" list).

c. Blum [1].

In [1] , Blum introduced the following wellknown notion of "step counting functions"

Definition 3.4: Given a Gödel numbering $\{\rho_i\}$, let $\{\phi_i\}$ be a family of p.r.f. satisfying the following two axioms

(S1) $\rho_i(x)$ is defined $\leftrightarrow$ $\phi_i(x)$ is defined

(S2) the function

$$M(i,x,m) := \begin{cases} 1, & \text{if } \phi_i(x) = m, \\ 0, & \text{otherwise,} \end{cases}$$

is recursive.

Then $\{\phi_i\}$ is called a family of "step counting functions with respect to $\{\rho_i\}$".

Let us consider functions ρ, $\bar{\Psi}$, κ, γ which determine a P-decomposition of $\Psi(i,x) := \rho_i(x)$. Then it is easy to show that the functions

(3.5) $\tilde{\phi}_i(x) := \mu_t(\kappa\ \bar{\Psi}^{(t)}\ \gamma(i,x) = 0)$ $\qquad (i = 0,1\ldots)$

satisfy (S1) and (S2) (see Buchberger [3]), i. e. $\{\tilde{\phi}_i\}$ are step counting functions for $\{\rho_i\}$. Thus, every theorem about step counting functions

gives some necessary condition about those $\bar{\Psi}$, κ, γ which, together with some ρ, determine a P-decomposition of some Gödel numbering Ψ.

Again, we get no sufficient conditions [+)]. And, clearly, we get only assertions corcerning the number of times $\bar{\Psi}$ is used in various computations and no propositions about other properties of ρ, $\bar{\Psi}$, κ, γ .

[+)] Also the converse of the above proposition on the $\{\tilde{\phi}_i\}$ is true, i. e. the proposition " $\{\phi_i\}$ is a family of step counting functions for Ψ " is equivalent to the proposition " $\{\phi_i\}$ is a family of functions definable from κ, $\bar{\Psi}$, γ by (3.5), where κ, $\bar{\Psi}$, γ together with some ρ determine a P-decomposition of Ψ " (see Buchberger [3]). However, we can't see any way how to get some reasonable sufficient conditions for our Problem 2.2 form this equivalence.

d. Nepomniaschy [10].

In our terminology, a résumé of Nepomniaschy's work [10] would be like this: One considers special ρ, $\bar{\Psi}$, κ, γ, namely those corresponding to a formalized presentation of a special variant of operator algorithms (Ershov [7]). Operator algorithms define a whole class of numberings of the p.r.f. (and other function classes) depending on which functions and predicates are taken as primitives (similar to the possibility of various choices of "oracles" in A-computability (see Davis [6], p. 20)). This is reflected by the fact that $\bar{\Psi}$ depends on a function F(i,n) which enumerates the set of functions

$$(3.6) \qquad \mathcal{F} := \{ f \mid (Ei)(x)(f(x) = F(i,x)) \} ,$$

and a predicate G(i,n) which enumerates the set of predicates

$$(3.7) \qquad \mathcal{G} := \{ g \mid (Ei)(x)(g(x) \leftrightarrow G(i,x)) \} .$$

Thus, let us write $\bar{\Psi}^{F,G}$ to make this dependence explicit.

In [10], Neopomniaschy gives some necessary and some sufficient conditions for F and G in order that

$$(3.8) \qquad \Psi^{F,G}(i,x) = \rho\left[\bar{\Psi}^{F,G}, \kappa\right]^{P} \gamma(i,x)$$

describes a numbering of the p.r.f.

Nopomniaschy's work seems to be closest to our problem. However, by the choice of one computability formalism, the class of $\bar{\Psi}$, κ considered is similarly restricted like in the work of Davis. Also, it is not explicitly stated that the $\Psi^{F,G}$ of (3.8) is always a Gödel numbering when Neomniaschy's conditions are satisfied. However, to show this should not make any difficulties since the proofs are constructive.

4. First results.

a. Some propositions about the operator P.

We first give some propositions about the operator P which plays an important role in the Problem 2.2.

First, it is readily seen that the P-operator and the μ-operator have, approximately, the same "power". Namely,

(4.1) $[f,g]^P(x) = \rho(x, \mu_t(g\,\rho(x,t) = o))$, where

$$\rho(x,o) = x,$$

$$\rho(x,t+1) = f\,\rho(x,t),$$

and

(4.2) $\mu_t(f(x,t)=o) = \sigma_{22}[\rho, \gamma]^P \sigma_2(x,o)$, where

$$\gamma(x) = f(\sigma_{21}(x), \sigma_{22}(x)),$$

$$\rho(x) = \sigma_2(\sigma_{21}(x), \sigma_{22}(x)+1).$$ +)

+) For the purposes of (4.1) and (4.2) the operator of primitive recursion and the μ-operator have to be modified so as to be applicable to p.r.f., too.

Further, we have

Proposition 4.1: The P-operator is an "effective" operator, i. e. there is a $\pi \in R_2$ such that

(4.3) $$(f,g)([\phi_f, \phi_g]^P = \phi_{\pi(f,g)})$$

($\{\phi_i\}$ being some Gödel numbering).

Proof: For instance, by a slight modification of the proof appearing in Strong [14] for a similar operator on the functions of some uniformly reflexive structure.

From a general theorem about the P-operator on arbitrary function sets (Buchberger [2]) there follows by specialization

Proposition 4.2: Starting from the functions $C_o(x)$, $S(x)$, $U_i^n(x_1,\ldots,x_n)$ +) and some pair α, σ of associating functions one can obtain all p.r.f. by finitely many times applying regular substitution and once the P-Operator.

Here, a pair of associating functions is a pair of functions satisfying the following two axioms

---

+) For notation see Davis [6], p.41.

(A1) $\quad (E\varepsilon)(x)(\alpha(\varepsilon,x) = \varepsilon),$

(A2) $\quad (a,b,c,d)(\alpha(\sigma(a,b,c\ ),d) = \begin{cases} c, & \text{if } d = b \\ \alpha(a,d), & \text{otherwise} \end{cases}$

An example of recursive associating functions and the intuitive backgound for this notion can be found in Buchberger [2].

An immediate consequence of Proposition 4.2 is the following

Proposition 4.3: Every p.r.f. f can be decomposed in the following way

(4.4) $\quad f(x_1,\ldots,x_n) = r[\bar{f},g]^P\ g(x_1,\ldots,x_n),$

where $r,\bar{f},k \in R_1$ and $g \in R_n$.

b. Some propositions on the Problem 2.2.

We, now , attempt to give a partial solution for Problem 2.2 by proving our main theorems4.5 and 4.6. We introduce the following notation

$$C^{x} := \{ \bar{\Psi}^{(t)}(x) \mid t \in N \} .$$

Theorem 4.5: Let $\bar{\Psi} \in R_1$ be such that there is a $\beta \in R_1$ with

(C1) (i) ($| C^{\beta(i)} | = \infty$),

(C2) (i,k) ($i \neq k \rightarrow C^{\beta(i)} \cap C^{\beta(k)} = \emptyset$)

(C3) The set $M := \bigcup_{i=0}^{\infty} C^{\beta(i)}$ is decidable.

Then one can find $\kappa, \rho \in R_1$ and $\gamma \in R_2$ such that $\rho, \bar{\Psi}, \kappa, \gamma$ determine a P-decomposition of a Gödel numbering Ψ ,

Theorem 4.6: If $\rho, \bar{\Psi}, \kappa, \gamma$ determine a P-decomposition of a Gödel numbering then there exist infinitely many $y_0, y_1, \ldots$ such that

(C'1) (i) ($| C^{y_i} | = \infty$),

(C'2) (i,k) ($i \neq k \rightarrow C^{y_i} \cap C^{y_k} = \emptyset$).

Proof of Theorem 4.5: We take some Gödel numbering $\{\rho_i\}$. Let $\Psi \in P_2$

be the function that describes $\{\rho_i\}$. Take $\{\phi_i\}$ to be a family of step counting functions for $\{\rho_i\}$, and define

$$(4.5) \qquad \gamma(i,x) = \beta\sigma_2(i,x),$$

$$(4.6) \qquad \kappa(y) = \begin{cases} 0, & \text{if } y \in M \wedge \phi(\pi(y), \delta(y)) = \tau(y) \\ 1, & \text{otherwise,} \end{cases}$$

$$(4.7) \qquad \rho(y) = \begin{cases} \Psi(\pi(y), \delta(y)), & \text{if } \kappa(y) = 0, \\ 0, & \text{otherwise,} \end{cases}$$

$$(4.8) \qquad \alpha(y) = \begin{cases} \sigma_{22}\, \mu_z\, (\bar{\Psi}^{(\sigma_{21}(z))} \beta(\sigma_{22}(z)) = y), & \text{if } y \in M, \\ 0, \text{ otherwise,} \end{cases}$$

$$(4.9) \qquad \tau(y) = \begin{cases} \sigma_{21}\, \mu_z (\bar{\Psi}^{(\sigma_{21}(z))} \beta(\sigma_{22}(z)) = y), & \text{if } y \in M \\ 0, \text{ otherwise,} \end{cases}$$

$$(4.10) \qquad \pi(y) = \sigma_{21}\, \alpha(y),$$

$$(4.11) \qquad \delta(y) = \sigma_{22}\, \alpha(y).$$

It is easily shown that γ, κ, ρ, α, τ, π, δ are recursive (to show this for ρ apply Corollary 2.5 of Davis [6], p.64). We want to prove

$$(4.12) \qquad (i,x)\ (\ \Psi(i,x) = \rho[\bar{\Psi},\kappa]^{p}\ \gamma(i,x)\ ).$$

Case 1: $\Psi(i,x) = a \in N$. From this we know by (S1) (p.15)

(4.13) $\phi(i,x) = T \in N$, and

(4.14) $\phi(i,x) \neq t$ for $t < T$.

We show

(4.15) $\kappa\bar{\Psi}^{(t)}\,\gamma(i,x) \neq o$ for $t < T$,

(4.16) $\kappa\bar{\Psi}^{(T)}\,\gamma(i,x) = o$.

For this we note that for $y \in M$

(4.17) $y = \bar{\Psi}^{(t)}\,\beta(j)$ for certain t, $j \in N$, which by (C1) and (C2) are uniquely determined.

Thus, for $y \in M$

(4.18) $y = \bar{\Psi}^{(\sigma_{21}(z))}\,\beta\sigma_{22}(z)$ for a certain unique z,

(4.19) $\alpha(y) = j$, and

(4.20) $\tau(y) = t$.

Now, take $y := \bar{\Psi}^{(t)}\,\gamma(i,x) \in M$. Using (4.10), (4.11), (4.19),

Hence, $[\bar{\Psi}, \kappa]^P \gamma(i,x)$ and $\rho[\bar{\Psi}, \kappa]^P \gamma(i,x)$ undefined.

Proof of Theorem 4.6: Theorem 4.6 will be a consequence of the following

Lemma 4.7: Let ρ, $\bar{\Psi}$, κ, $\dot{\gamma}$ determine a P-decomposition of $\Psi \in P_2$. Let $\beta \in R_1$ be such that

(C''4) $(i,t)(\kappa\, \bar{\Psi}^{(t)}\, \gamma(\beta(i),o) \neq o)$

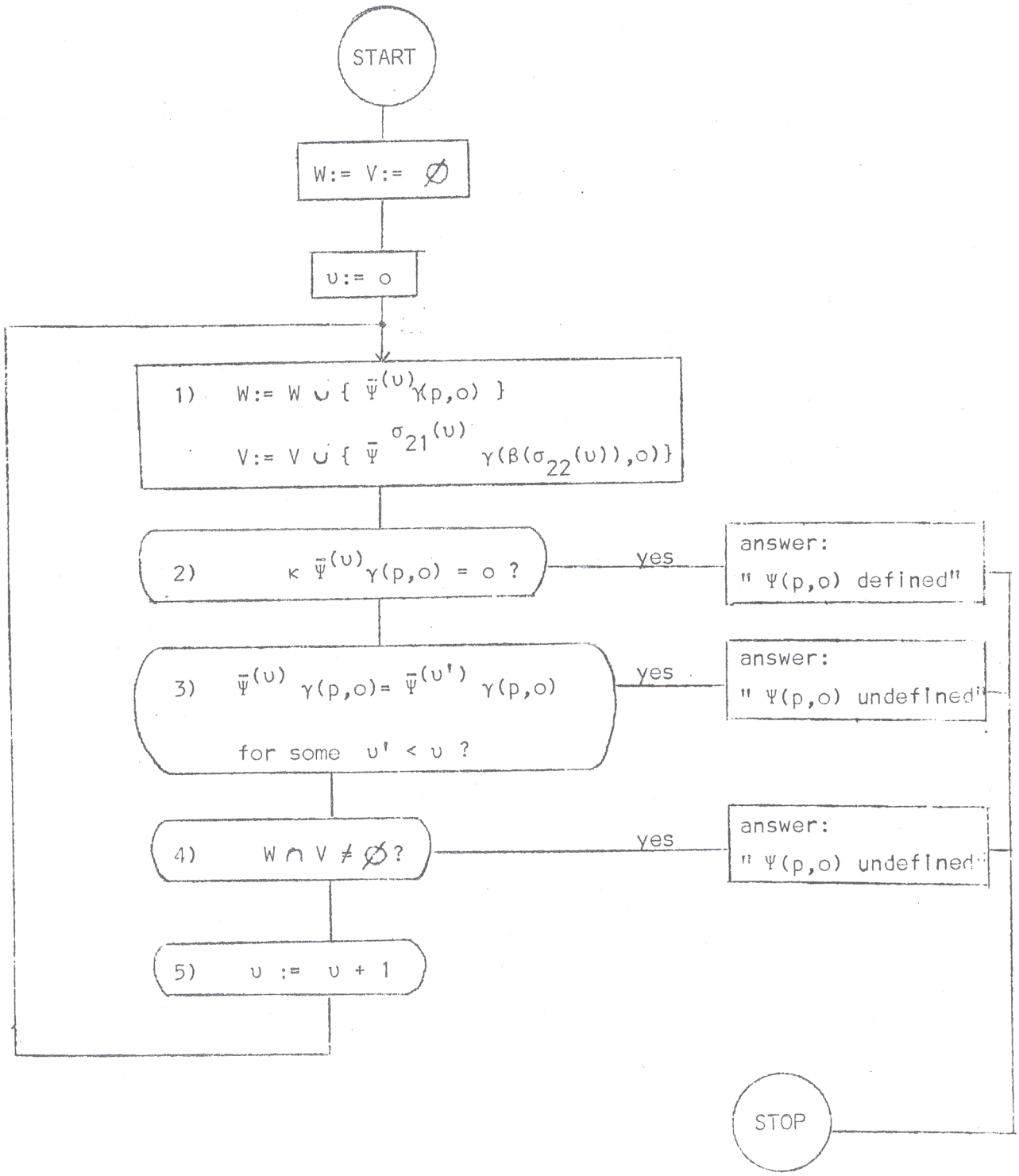
(C''5) $(p)(\mid C^{\gamma(p,o)} \mid < \infty \vee$

$(Et)(\kappa\, \bar{\Psi}^{(t)}\, \gamma(p,o) = o) \vee$

$(Ei)(C^{\gamma(p,o)} \cap C^{\gamma(\beta(i),o)} \neq \emptyset)$.

Then the property " $\Psi(p,o)$ is defined" is decidable.

Proof: We show that the following algorithm is a decision procedure for the property " $\Psi(p,o)$ is defined":



It is clear that after the υ-th step of the algorithm

(4.29) $W := \{ \bar{\Psi}^{(t)} \gamma(p,o) \mid o \leq t \leq v \}$

(4.30 $V := \{ \bar{\Psi}^{(t)} \gamma(\beta(i),o) \mid o \leq \sigma_2(\cdot,i) \leq v \}$.

Further, from (C''4)

(4.31) $(i)(\ \Psi(\beta(i),o)$ undefined).

Now, assume $\Psi(p,o)$ to be defined, i.e.

$$(ET)(\ \Psi(p,o) = \rho\bar{\Psi}^{(T)} \gamma(p,o) \ \wedge$$

$$\kappa\, \bar{\Psi}^{(T)} \gamma(p,o) = o \quad \wedge$$

$$(\tau)\ (\ \tau < T \ \rightarrow \kappa\, \bar{\Psi}^{(\tau)} \gamma(p,o) \neq o)\ ).$$

Then the algorithm will stop in any case after the T-th step at the branching point 2). It can't stop after some τ-th step ($\tau < T$) at the branching point 3), since this would mean $\bar{\Psi}^{(\tau)} \gamma(p,o) = \bar{\Psi}^{(\tau')}\gamma(p,o)$ for some $\tau' < \tau$ and $\kappa\bar{\Psi}^{(\tau')} \gamma(p,o) \neq o$ for all $\tau' \leq \tau$, i.e. $\Psi(p,o)$ undefined. Similarly, it can't stop after some τ-th step ($\tau < T$) at branching point 4) since this would mean $\kappa\, \bar{\Psi}^{(\tau')} \gamma(p,o) \neq o$ for all $\tau' \leq \tau$ and $\bar{\Psi}^{(\tau'')} \gamma(p,o) = \bar{\Psi}^{(t)} \gamma(\beta(i),o)$ for some $\tau'' \leq \tau$ and some $t,i \in N$, hence, $\Psi(p,o) = \Psi(\beta(i),o)$, i.e. $\Psi(p,o)$ undefined.

On the other hand, assume $\Psi(p,o)$ to be undefined, i.e.

(4.33) $(\upsilon)(\kappa\ \bar{\Psi}^{(\upsilon)}\ \gamma(p,o) \neq o)$, and either

(4.34 $(E\upsilon,\ \upsilon')(\bar{\Psi}^{(\upsilon)}\ \gamma(p,o) = \bar{\Psi}^{(\upsilon')}\ \gamma(p,o) \wedge \upsilon > \upsilon')$

or, by (C''5),

(4.35) $(E\upsilon,\ i,\ t)(\ \bar{\Psi}^{(\upsilon)}\ \gamma(p,o) = \bar{\Psi}^{(t)}\ \gamma(\beta(i),o)\ )$.

If (4.33) and (4.34) then at least after the υ-th step the algorithm stops at branching point 3). Because of (4.33) it can't stop earlier at 2), i.e. the algorithm gives the right answer. If (4.33) and (4.35) then at least after the $\bar{\upsilon}$-the step $(\bar{\upsilon} = \max(\upsilon, \sigma_2(t,1))\ )$ the algorithm stops at branching point 4). Because of (4.33) it can't stop earlier at 2), i. e. the algorithm gives the right answer also in this case. Thus Lemma 4.7. is proven.

Now, let $\rho, \bar{\Psi}, \kappa, \gamma$ determine a P-decomposition of a Gödel numbering and assume the conclusion of Theorem 4.6 to be false, i. e. let $X := \{y_1, \ldots, y_k\}$ be a "maximal" set such that

(4.36) (C'1) $|\ C^{y_i}\ | = \infty$ for $1 \leq i \leq k$,

(C'2) $C^{y_i} \cap C^{y_1} = \emptyset$ for $1 \leq i,\ 1 \leq k,\quad i \neq 1$.

That means for arbitrary y

(4.37) $\quad |\, C^{y} \,| < \infty$ or $\quad C^{y} \cap C^{y_i} \neq \emptyset \quad$ for some $i \leq k$.

We define

$$X_1 := \{\, y \mid y \in X \wedge (E\tau)(\kappa\, \bar{\Psi}^{(\tau)}\, y \neq 0 \quad \text{for } \tau \geq t) \,,$$

$$X_2 := X - X_1,$$

$$P_i := \begin{cases} \emptyset & , \text{ if } y_i \in X_2 \vee i > k \\ \{\, p \mid (C^{\gamma(p,0)} \cap C^{y_i} \neq \emptyset) \wedge (t)(\kappa\, \bar{\Psi}^{(t)}\, \gamma(p,0) \neq 0)\} & , \text{ otherwise} \end{cases}$$

$$\beta(i) := \begin{cases} \bar{p}, & \text{if} \quad P_i = \emptyset, \\ \mu_p(p \in P_i) \,, & \text{otherwise,} \end{cases}$$

where $\bar{p}$ is such that $\Psi(\bar{p},0)$ is undefined, i. e.
$(t)(\kappa\, \bar{\Psi}^{(t)}\, \gamma(\bar{p},0) \neq 0)$.

$\beta \in R_1$, since β is constant except for finitely many arguments. We shall show that β satisfies (C''4) and (C''5).

(C''4): Either $P_i = \emptyset$, then $\beta(i) = \bar{p}$, hence by the definition of $\bar{p}$ $(t)(\kappa\, \bar{\Psi}^{(t)}\, \gamma(\beta(i),0) \neq 0)$, or $P_i \neq \emptyset$, hence $\beta(i) \in P_i$ and therefore by the definition of P_i $(t)(\kappa\, \bar{\Psi}^{(t)}\, \gamma(\beta(i),0) \neq 0)$.

(C''5): Assume for some $p \in N$

(4.38) $\quad |C^{\gamma(p,o)}| = \infty,$

(4.39) $\quad (t)(\kappa\, \bar{\Psi}^{(t)}\, \gamma(p,o) \neq o)$, and

(4.40) $\quad (i)(C^{\gamma(p,o)} \cap C^{\gamma(\beta(i),o)} = \emptyset).$

We shall show

(4.41) $\quad C^{\gamma(p,o)} \cap C^{y_i} = \emptyset \quad$ for all $i \leq k$.

Take, first, $y_i \in X_2$, then $C^{\gamma(p,o)} \cap C^{y_i} = \emptyset$, because otherwise $(Et)(\kappa\, \bar{\Psi}^{(t)}\, \gamma(p,o) = o)$, which contradicts (4.39). If, on the other hand, $y_i \in X_1$, then $C^{\gamma(p,o)} \cap C^{y_i} \neq \emptyset$ would imply $p \in P_i$, hence $C^{\gamma(p,o)} \cap C^{\gamma(\beta(i),o)} \neq \emptyset$, which contradicts (4.40) Thus, under the assumptions (4.38)-(4.40), (4.41) is proven. (4.38) together with (4.41) is a contradiction to (4.37), hence (C''5).

However, (C''5) true would make " $\Psi(p,o)$ defined" a decidable property (by Lemma 4.7), which contradicts Rice's theorem for general Gödel numberings (Rogers [13]). Thus, we have to reject (4.37), i.e. the conclusion of Theorem 4.6 is true.

Remarks on the Theorems 4.5 and 4.6:

1. The conditions (C1) - (C3) are really "easy to check" (cf. p.11) for all the computability formalisms encountered in the literature. For instance, for the Turing formalism, consider the "states" y_i which consist of the following components

(4.42)
$$\begin{array}{l} \text{the program } p_i \\ \text{the square number } o \\ \text{the internal configuration } q_o \\ \text{the tape inscription b ("blank"),} \end{array}$$

where p_i consists of the quadruples $q_o b R q_o$, $q_1 b R q_1, \ldots, q_l b R q_i$. It is clear that the effect of the computations started with the states y_i is the same for all i: remain in the internal configuration q_o and move steadily to the right.

If we encode the states of the Turing formalism in some effective way, the initial states y_i can be given by some $\beta \leq R_1$. (C1) is true since the square number steadily encreases and the computation never stops. (C2) is satisfied because two computations starting from y_i and y_j $(i \neq j)$ always differ by their Turing tables p_i, p_j. As for (C3), it is readily seen that the states which occur in the computation starting from a y_i consist of the following components

(4.43)

the program p_i

a square number $a \in N$

the internal configuration q_o

the tape inscription "blank".

It is (intuitively) decidable whether some given state has the form (4.43), hence, after some effective encoding this property will be decidable.

Similar "programs" can be written for all the other formalisms to show that the preposition of Theorem 4.5 is ful filled, and therefore there is good reason to believe that Theorem 4.5 is a useful tool for easy equivalence proofs of further computability formalisms.

2. Though, from the examples, there is much evidence to suspect that suitable y_i in Theorem 4.6 can always be effectively enumerated and that (C3) is also a necessary condition, we were not able to prove this under the very general assumptions on P-decompositions.

3. There are many equivalent forms of the condition (C1) $\wedge$ (C2) $\wedge$ (C3) which more apparently show that the intuitive meaning of this condition is: $\bar{\Psi}$ must have the power of steadily altering one "component" of the "state" while not changing some other, i.e. there must be the possibility to execute a computation with some information while storing some other information.

Maybe, it represents some interesting insight into the essence of computation that this very general principle unter the very general assumptions for P-decompositions is sufficient and (at least in its non-effective version) necessary for a P-decomposition to determine a Gödel numbering.

We add some more propositions on ρ, $\bar{\Psi}$, κ, γ which are of minor interest. At first, generalizing Davis [5] one can easily show

Proposition 4.8: If ρ, $\bar{\Psi}$,κ,γ determine a P-decomposition of a Gödel numbering then

$$C := \{ y \mid [\bar{\Psi},\kappa]^P (y) \text{ is defined} \}$$

is a complete set (see Definition 3.3).

Proof: Let $\Psi_T(i,x)$ be the Gödel numbering of Kleene's normal form theorem (Davis [6], p.63), and $\tau \in R_1$ such that $(i,x)(\Psi_T(i,x) = \Psi(\tau(i),x))$. Then

$$\begin{aligned} U_1 &:= \{ x \mid \Psi_T(i,x) \text{ is defined} \} = \\ &= \{ x \mid \Psi(\tau(i),x) \text{ is defined} \} = \\ &= \{ x \mid \rho [\bar{\Psi},\kappa]^P \gamma(\tau(i),x) \text{ is defined} \} = \\ &= \{ x \mid \gamma(\tau(i),x) \in C \} . \end{aligned}$$

Thus, taking $f(i,x) := \gamma(\tau(i),x)$, the proposition is proven.

Continuing the investigations presented in Markov [9] one might suspect that:

If ρ, $\bar{\Psi}$, κ, γ determine a P-decomposition of a Gödel numbering then ρ must be of large oscillation, i. e.

(4.44) $\quad (y,z)(Ex)\ (\rho(x) = y \wedge x > z).$

That this is not the case is shown by the following argument. Let some ρ, $\bar{\Psi}$, κ, γ determine a P-decomposition of a Gödel numbering Ψ . We define

$$\bar{\Psi}'(y) := \begin{cases} \bar{\Psi}(y), \text{ if } \kappa(y) \neq o \\ \mu_z(\kappa(z) = o \wedge \rho(z) = \rho(y)), \text{ otherwise,} \end{cases}$$

(4.45)
$$\kappa'(y) := \begin{cases} \kappa(y), \text{ if } \kappa(y) \neq o \\ 1, \text{ if } \kappa(y) = o \wedge y \neq \mu_z(\kappa(z) = o \wedge \rho(z) = \rho(y)) \\ o, \text{ otherwise,} \end{cases}$$

$$\rho'(y) \quad \begin{cases} \rho(y), \text{ if } \kappa'(y) = o \\ o, \text{ otherwise.} \end{cases}$$

It is easy to show that ρ', $\bar{\Psi}'$, κ', γ determine a P-decomposition of the same Gödel numbering Ψ . However, for $r > o$ $\rho'(y) = r$ holds for exactly one y, which contradicts (4.44).

On the other hand, one can show

Proposition 4.9: If $\tau_{21} \in R_1$ is of large oscillation then one can find $\bar{\Psi}$, κ, γ such that τ_{21}, $\bar{\Psi}$, κ, γ determine a P-decomposition of a Gödel numbering.

Proof: If ρ is of large oscillation then one can find $\tau_{22} \in R_1$, $\tau_2 \in R_2$ such that τ_2, τ_{21}, τ_{22} are pairing functions (following Markov [9]). Take some ρ', $\bar{\Psi}'$, κ', γ' which determine a P-decomposition of a Gödel numbering then $\bar{\Psi}$, κ, γ defined by

$$
\begin{aligned}
&\gamma(i,x) := \tau_2(\rho'\, \gamma'(i,x), \gamma'(i,x)), \\
&\bar{\Psi}(y) := \tau_2(\rho'\, \bar{\Psi}'\, \tau_{22}(y), \bar{\Psi}'\, (\tau_{22}(y))), \\
&\kappa(y) := \begin{cases} 0, & \text{if } \kappa'\, \tau_{22}(y) = 0 \\ 1, & \text{otherwise,} \end{cases}
\end{aligned}
\tag{4.46}
$$

are the desired functions.

The following proposition shows that we can't get through without any ρ.

Proposition 4.10: If ρ, $\bar{\Psi}$, κ, γ determine a P-decomposition of a Gödel numbering Ψ, then ρ can't be one-one.

Proof: Assume ρ to be one-one, and take i,x such that $\Psi(i,x)$ is undefined, i.e. $(t)(\kappa\, \bar{\Psi}^{(t)}\, \gamma(i,x) \neq 0)$. Let $\rho\, \gamma(i,x) = r$. Since $y := \gamma(i,x)$ is

the only y such that $\rho(y) = r$, but $\kappa(y) \neq o$ there can't be any i', x' such that $\Psi(i', x') = r$. This is, of course, impossible.

Maybe, it is interesting to note that it is not necessary for $\bar{\Psi}$ to "allow cycles", i.e. that

$$(4.47) \quad (Ey, t_1, t_2)(t_1 \neq t_2 \wedge \bar{\Psi}^{(t_1)}(y) = \bar{\Psi}^{(t_2)}(y)).$$

This is clear by inspection of the special function

$$(4.48) \quad \bar{\Psi}(y) := \sigma_3(\sigma_{31}(y), \sigma_{32}(y), \sigma_{33}(y)+1),$$

which is such that there are ρ, κ, γ such that ρ, $\bar{\Psi}$, κ, γ determine a P-decomposition of a Gödel numbering (see Buchberger [3], p.4). Nevertheless, for this $\bar{\Psi}$ condition (4.47) is not fulfilled. This fact is not supported by the experience with programming languages where "idle loops" are quite an ordinary phenomenon.

5. Conclusions.

We hope that we could give an idea how a detailed study of P-decompositions of Gödel numberings could add to our insight in the essence of computability formalisms. A further investigation should have the following stages:

a. A thorough solution to Problem 2.2, which should, especially, include a theorem of the following kind: $\bar{\Psi}$, κ fulfill some criterion K if there exist ρ, γ such that ρ, $\bar{\Psi}$, κ, γ determine a P-decomposition of some Gödel numbering.

b. Definition and study of a notion of "program" independently of γ, ρ, only based on the structure of $\bar{\Psi}$ and κ. Is the property "programming language which alters its programs during execution" an intrinsic property of the corresponding $\bar{\Psi}$? (It is not an intrinsic property of a Gödel numbering: The ALGOL definition given in Lauer [8] works with a $\bar{\Psi}$ that alters programs during execution though this would not be necessary for ALGOL).

c. How does $\bar{\Psi}$ (and, maybe γ) determine a structure of the programs and data, i. e. how does semantics define syntax?

d. Can we give theorems about the "recognizing power" of some P-decomposition $\rho\,[\bar{\Psi}, \kappa]^P\,\gamma(i,x)$ (where ρ is a mapping $\rho : N \to \{0,1\}$), which would give a general framework for the results on the recognizing power of special classes of automata (finite automata, pushdown automata etc.).

References

[1] M. Blum, A machine-independent theory of the complexity of recursive functions, JACM 14/2, 1967.

[2] B. Buchberger, Associating functions and the operator of conditioned iteration (Russian), Comm. of the Joint Institute for Nuclear Research, Dubna, 1971, (english translation: Bericht Nr. 71-1, Institut f. num. Math., Univ. Innsbruck).

[3] B. Buchberger, A comment on Blum's signal functions , Bericht Nr. 71-3, Institut f. num. Math., Univ. Innsbruck, 1971.

[4] M. Davis, A note on universal Turing machines, in: Automata Studies (Shannon, McCarthy ed.), Princton University Press, 1956.

[5] M. Davis, The definition of universal Turing machine, Proc. of the AMS, 8, pp.1125-1126, 1957.

[6] M. Davis, Computability and unsolvability, McGraw Hill, 1958.

[7] A. P. Ershov, Operator Algorithms I, Problemy Kibernetiki 3, 1960.

[8] P. Lauer, Formal defintion of ALGOL 60, Technical Report TR. 25088, IBM-Laboratory Vienna, 1968.

[9] A. A. Markov, On the representation of recursive functions (Russian), Doklady Akademii Nauk SSSR, n.s., vol.57, p.1891-1892, 1947.

[10] V. A. Nepomniaschy, Conditions for the algorithmic completeness of the systems of operations, in: Proc. of the IFIP Congress 71, Ljubljana, 1971.

[11] H. G. Rice, Classes of recursively enumerable sets and their decision problems, Trans. of the AMS, vol. 74, pp.358-366, 1953.

[12] H. Rogers Jr., Gödel numberings of partial recursive functions, The Journal of Symbolic Logic 23/3, 1958.

[13] H. Rogers Jr., Theory of recursive functions and effective computability, McGraw Hill, 1967.

[14] H. R. Strong, Algebraically generalized recursive function theory, IBM J. Res. and Dev., 12/6, 1968.