Arch. math. Logik 16 (1974), 85–96.

ON CERTAIN DECOMPOSITIONS OF GÖDEL NUMBERINGS*

B. Buchberger, Innsbruck

Summary: We define certain decompositions of the functions that describe Gödel numberings of the partial recursive functions (Section 2). These decompositions reflect the way in which concrete Gödel numberings may be obtained from the known computability formalisms. We show that such decompositions exist for all partial recursive functions (Section 3). It turns out that there is an intimate connection between these decompositions and Blum's step counting functions which yields a suggestive interpretation of Blum's notion (Section 4). In terms of these decompositions we, finally, give an exact formulation for a basic problem in the theory of computability formalisms, which, on an intuitive level, would read as follows: Find conditions on the expressive power of one step in a given computability formalism such that all partial recursive functions can be represented within that formalism. We derive two theorems which may be regarded as a first step in a thorough study of this problem.

1. Notation

N ... set of the natural numbers including 0.
An "integer" is an element of N.
Universal and existential quantification is always over N. Also, by a "function" we mean a function with arguments and values in N.
p.r.f. ... partial recursive function(s).
A "recursive" function is a partial recursive, total function.
P_n ... set of all n-ary p.r.f.
R_n ... set of all n-ary recursive functions.
If f, g denote functions, then $fg(x)$ often stands for $f(g(x))$. Further, $f^{(t)}(x)$ is defined as follows:

$$f^{(0)}(x) = x, \quad f^{(t+1)}(x) = ff^{(t)}(x) \quad (\text{for } t \in N).$$

2. Definitions

Definition 1 (Rogers [11], Uspenskii [14]): A function $\Psi \in P_2$ is said to describe a Gödel numbering (of the unary p.r.f.) if for every $f \in P_2$ there exists a $g \in R_1$ such that

$$(\forall i, x)\,(f(i, x) = \Psi(g(i), x)).$$

* Eingegangen am 18. 12. 1971.

Definition 2: The operator P associates with any two unary functions f, g the unary function $[f, g]^P$ that is defined by the following recursion:

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

Informally speaking, $[f,g]^P(x)$ is evaluated by successive evaluation of $f^{(t)}(x)$ $(t=0,1,\ldots)$ until, for some $t, gf^{(t)}(x)=0$. If no such t exists or a certain $f^{(t)}(x)$ or $gf^{(t)}(x)$ necessary for the above evaluation process is not defined then $[f,g]^P(x)$ is not defined either.

Definition 3: $r, \bar{f}, k, g$ determine a P-decomposition of f if $r, \bar{f}, k \in R_1$, $g \in R_2$, $f \in P_2$, and

$$(\forall i, x)\,(f(i,x) = r[\bar{f}, k]^P g(i,x))\ ^1.$$

Note that the recursiveness of $r, \bar{f}, k, g$ is essential in this definition.

3. The existence of P-decompositions

Theorem 1: For every $f \in P_2$ there exist $r, \bar{f}, k, g$ such that $r, \bar{f}, k, g$ determine a P-decomposition of f.

Proof: A much stronger result and a generalization for functions over arbitrary sets A (using the notion of "Basic Recursive Function Theory", Strong [13]) can be found in [2]. For obtaining Theorem 1 from Corollary 2.6 in [2] one only has to find a recursive example of "associating functions" (see, for instance, the example at the end of Section 1 in [2]).
Theorem 1 can also be obtained from Kleene's normal form theorem by expressing the μ-operator by means of the P-operator, any pairing functions and the successor function.

4. P-decompositions and Blum's step counting functions

There is an intimate connection between P-decompositions and Blum's wellknown step counting functions (Theorem 2) which gives a suggestive interpretation to Blum's notion. We, first, recall Blum's definition:

Definition 4 (Blum [1]): Let $\Psi \in P_2$ describe a Gödel numbering. Then the functions $\Phi_i \in P_1$ $(i=0,1,\ldots)$ are called step counting functions (with respect to Ψ) if

(S 1) $\Phi_i(x)$ defined $\leftrightarrow$ $\Psi(i,x)$ defined

(S 2) the function

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

is recursive.

[1] We adopt the usual meaning of the sign "=": both sides are defined and equal or both sides are undefined.

Alternatively, we shall describe the same notion in terms of P-decompositions of Ψ.

Definition 5: Let $\Psi \in P_2$ describe a Gödel numbering. Then the functions $\Phi_i \in P_1$ $(i=0, 1, \ldots)$ are called proper step counting functions (with respect to Ψ) if

(PS) there exist $\varrho, \bar{\Psi}, \kappa, \gamma$ which determine a P-decomposition of Ψ such that
$$\Phi_i(x) = (\mu t)(\kappa \bar{\Psi}^{(t)} \gamma(i, x) = 0).$$

As is to be seen from the definition, proper step counting functions count the number of applications of the function $\bar{\Psi}$ in the computation of the value $\Psi(i, x)$, i.e. it counts the number of steps in a proper sense, if we conceive $\bar{\Psi}$ to define the behaviour of the "computability formalism" $\bar{\Psi}, \kappa$ during one elementary step (compare also the remarks after Problem 1, Section 5).
On the other hand, the functions Φ_i of Definition 4 don't possess, necessarily, such an easy interpretation which fits their name equally well. In fact, very abstract constructions or functions sets defined on the basis of quite different concepts may satisfy (S 1) and (S 2), too (for instance, the functions Φ_i defined by: $\Phi_i(x) = m$ if and only if the Turing machine with Gödel number i uses exactly m cells of the tape when working on input x, see Blum [1]).
However, we shall show now that the two notions "step counting functions" and "proper step counting functions" coincide. This means, especially, that every set of step counting functions for a given Gödel numbering Ψ can be interpreted as, in fact, counting the number of applications of $\bar{\Psi}$ (= "steps") for a suitable P-decomposition of Ψ determined by some $\varrho, \bar{\Psi}, \kappa, \gamma$.

Theorem 2 (Characterization Theorem for Blum's step counting functions): Let $\Psi \in P_2$ describe a Gödel numbering and take $\Phi_i \in P_1$ $(i=0, 1, \ldots)$. Then the Φ_i satisfy the axioms (S 1) and (S 2) if and only if they satisfy (PS).

Proof: "(PS)$\rightarrow$(S 1) $\wedge$ (S 2)":
By assumption, Ψ has the following representation
$$\Psi(i, x) = \varrho[\bar{\Psi}, \kappa]^P \gamma(i, x),$$
where $\varrho, \bar{\Psi}, \kappa, \gamma$ are recursive. For recursive $\bar{\Psi}, \kappa$ the application of P can be expressed by the μ-operator and primitive recursion:
$$[\bar{\Psi}, \kappa]^P(\xi) = \theta(\xi, (\mu t)(\kappa\theta(\xi, t) = 0)),$$
where
$$\theta(\xi, 0) = \xi, \qquad \theta(\xi, t+1) = \bar{\Psi}\theta(\xi, t).$$

Note that $\bar{\Psi}^{(t)}(\xi) = \theta(\xi, t)$. Hence,
$$\Phi_i(x) = (\mu t)(\kappa \bar{\Psi}^{(t)} \gamma(i, x) = 0) \text{ defined} \leftrightarrow \Psi(i, x) \text{ defined}.$$
Further,
$$\Phi_i(x) = m \leftrightarrow (\forall t)(t < m \rightarrow \kappa\theta(\gamma(i, x), t) \neq 0) \wedge \kappa\theta(\gamma(i, x), m) = 0.$$

Since bounded universal quantification does not destroy recursiveness "$\Phi_i(x) = m$" is decidable.
"(S 1) $\wedge$ (S 2)$\Rightarrow$(PS)": Let $\sigma_3 \in R_3, \sigma_{31}, \sigma_{32}, \sigma_{33} \in R_1$ be "pairing functions" for the triples of integers, σ_3 onto N, and $\xi_j = \sigma_{3j}(\xi)$ $(j = 1, 2, 3)$. We define $\varrho, \bar{\Psi}, \kappa, \gamma$ as follows

$$\gamma(i, x) = \sigma_3(i, x, 0),$$
$$\bar{\Psi}(\xi) = \sigma_3(\xi_1, \xi_2, \xi_3 + 1),$$
$$\kappa(\xi) = 1 - M(\xi_1, \xi_2, \xi_3),$$
$$\varrho(\xi) = \begin{cases} \Psi(\xi_1, \xi_2), & \text{if} \quad \Phi_{\xi_1}(\xi_2) = \xi_3 \\ 0, & \text{otherwise}. \end{cases}$$

First, it is clear that ϱ, $\bar{\Psi}$, κ, γ are recursive (as for the recursiveness of ϱ cf. Davis [6], p. 64).
Further, we have to show

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

and

$$\Phi_i(x) = (\mu t)(\kappa \bar{\Psi}^{(t)} \gamma(i, x) = 0).$$

Case 1: $\Psi(i, x)$ defined. Then $\Phi_i(x)$ is defined by (S 1), say $\Phi_i(x) = T$. Thus, $M(i, x, t) = 0$ for $t < T$ and $M(i, x, T) = 1$. Further, it is easy to see that

$$\bar{\Psi}^{(t)} \gamma(i, x) = \sigma_3(i, x, t) \quad \text{for all } t$$

and

$$\kappa \bar{\Psi}^{(t)} \gamma(i, x) = 1 - M(i, x, t) = \begin{cases} 1, & \text{for} \quad t < T \\ 0, & \text{for} \quad t = T. \end{cases}$$

Now,

$$\varrho[\bar{\Psi}, \kappa]^P \gamma(i, x) = \varrho \bar{\Psi}^{(T)} \gamma(i, x) = \varrho \sigma_3(i, x, T) = \Psi(i, x),$$

and

$$(\mu t)(\kappa \bar{\Psi}^{(t)} \gamma(i, x) = 0) = T = \Phi_i(x).$$

Case 2: $\Psi(i, x)$ not defined. Then $M(i, x, t) = 0$ and $\kappa \bar{\Psi}^{(t)} \gamma(i, x) = 1$ for all t. Thus, $\varrho[\bar{\Psi}, \kappa]^P \gamma(i, x)$ will be undefined, too. By (S 1) $\Phi_i(x)$ is undefined and so is $(\mu t)(\kappa \bar{\Psi}^{(t)} \gamma(i, x) = 0)$.

5. A basic problem

In this section we shall consider the following

Problem 1: Find necessary and sufficient conditions for $\varrho, \bar{\Psi}, \kappa \in R_1$ and $\gamma \in R_2$ to determine a P-decomposition of a function Ψ, which describes a Gödel numbering.

We briefly state the reasoning why we consider this problem to be basic in the theory of computability formalisms (for a detailed exposition of the underlying intuitive concept see [3]): it is a matter of tedious but straightforward notational work to show that the wellknown historical computability formalisms (Turing machines, Kleene's formalism, Markov's formalism etc., including universal programming languages) which gave rise to different concrete Gödel numberings of the p.r.f. are all given in the form $[\bar{\Psi}_F, \kappa_F]^P$. Here, $\bar{\Psi}_F$ is an(intuitively)computable total function which describes the behaviour of the formalism F during one step and κ_F is an (intuitively) computable, total function which decides whether a given state during a computation in the formalism F is terminal. Though the various formalisms start from quite distinct points of view what is to be regarded as the elementary computational step and as a terminal situation (reflected by the different structure of the $\bar{\Psi}_F$ and κ_F), the rule by which one comes from some initial situation ξ to an eventual terminal situation ξ' is the same with all formalisms: "successively apply $\bar{\Psi}_F$ to ξ until κ_F decides that the momentary situation is terminal". This is exactly what can be written

$$\xi' = [\bar{\Psi}_F, \kappa_F]^P(\xi) .$$

For programming languages, this general point of view yields exactly the Vienna method for the definition of the semantics of programming languages (Lucas et al. [8]).

The operator P seems to describe 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.

Given some computability formalism F, i.e. given $\bar{\Psi}_F$ and κ_F, one can use it to define an effective list of all (unary) p.r.f. by means of some (intuitively) computable, total input/output functions γ and ϱ. It the formalism is "strong" enough and γ and ϱ are suitably chosen then

$$\Psi(i, x) = \varrho[\bar{\Psi}_F, \kappa_F]^P \gamma(i, x)$$

will describe a Gödel numbering.

Thus, Gödel numberings until now have been given by constructing a P-decomposition of a function that describes the numbering. On the other hand, by Theorem 1 we know that every function that describes a Gödel numbering has a P-decomposition. Thus, Problem 1 may be conceived as a reasonable, precise substitute for the vague question: what are the necessary and sufficient properties of a computability formalism and input/output function to provide an effective list of the p.r.f. The most interesting part of the problem seems to be conditions on $\bar{\Psi}$ since this would give insight into the necessary and sufficient power of the "instruction list" of the formalism in order that the formalism be as "strong" as the standard formalisms.

In the literature, only very spacial versions of Problem 1 have been considered until now. Thus, in [4] and [5] Davis gave a necessary condition (namely the completeness of the set $\{\xi \,|\, [\Psi_M, \kappa_M]^P(\xi) \text{ defined}\}$) for the transition function $\bar{\Psi}_M$

and the termination criterion κ_M of a Turing machine M which is satisfied if there are recursive input/output functions γ and ϱ such that

$$\varrho[\bar{\Psi}_M, \kappa_M]^P \gamma(i, x)$$

describes a Gödel numbering.

Recently, Nepomniashy [10] gave some necessary and some sufficient conditions for the function f and the predicate p such that

$$\Psi(i, x) = \varrho[\bar{\Psi}_E^{f,p}, \kappa_E]^P \gamma(i, x)$$

describes a numbering of the p.r.f. Here, $\bar{\Psi}_E^{f,p}$, κ_E are the transition function and the termination criterion of some variant of Ershov's formalism (operator algorithms, see [7]). $\bar{\Psi}_E^{f,p}$ depends in some way on the "primitives" f, p such that Ershov's formalism, in effect, gives a whole class of computability formalisms. γ and ϱ are fixed.

It is the aim of the present note to start a systematic study of the general Problem 1 without any restrictive assumptions about the functions ϱ, $\bar{\Psi}$, κ, γ beyond assuming them to be recursive. We, first, concentrate on the function $\bar{\Psi}$.

Theorem 3: Let $\bar{\Psi} \in R_1$ be such that there is a recursively enumerable sequence $\xi_0, \xi_1, \ldots (\xi_1 \in N)$ with

(C 1) $(\forall i, t_1, t_2)(t_1 \neq t_2 \rightarrow \bar{\Psi}^{(t_1)}(\xi_i) \neq \bar{\Psi}^{(t_2)}(\xi_i))$

(C 2) $(\forall i, k, t_1, t_2)(i \neq k \rightarrow \bar{\Psi}^{(t_1)}(\xi_i) \neq \bar{\Psi}^{(t_2)}(\xi_k))$

(C 3) $M := \{\bar{\Psi}^{(t)}(\xi_i) | i, t \in N\}$ is decidable.

Then one can find $\kappa, \varrho \in R_1$ and $\gamma \in R_2$ such that ϱ, $\bar{\Psi}$, κ, γ determine a P-decomposition of a function Ψ which describes a Gödel numbering.

Proof: Take some Ψ that describes a Gödel numbering. Let $\Phi_i \in P_1$ $(i = 0, 1, \ldots)$ be step counting functions with respect to $\Psi, \sigma_2, \sigma_{21}, \sigma_{22}$ some recursive pairing functions for the pairs of integers, and $\beta \in R_1$ such that $\xi_i = \beta(i)$ $(i = 0, 1, \ldots)$. We define

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

$$\nu(\xi) = \begin{cases} (\mu z)(\bar{\Psi}^{\sigma_{21}(z)}\beta\sigma_{22}(z) = \xi), & \text{if } \xi \in M, \\ 0, & \text{otherwise}, \end{cases}$$

$$\tau(\xi) = \sigma_{21}\nu(\xi),$$

$$\alpha(\xi) = \sigma_{22}\nu(\xi),$$

(2) $$\pi(\xi) = \sigma_{21}\alpha(\xi),$$

(3) $$\delta(\xi) = \sigma_{22}\alpha(\xi),$$

(4) $$\kappa(\xi) = \begin{cases} 0, & \text{if } \xi \in M \wedge \Phi(\pi(\xi), \delta(\xi)) = \tau(\xi), \\ 1, & \text{otherwise}, \end{cases}$$

(5) $$\varrho(\xi) = \begin{cases} \Psi(\pi(\xi), \delta(\xi)), & \text{if } \kappa(\xi) = 0, \\ 0, & \text{otherwise}. \end{cases}$$

$\gamma, \nu, \tau, \alpha, \pi, \delta, \kappa, \varrho$ are recursive. For γ this is obvious. By the definition of M and (C 3) ν is p.r. and total, hence recursive. Thus, $\tau, \alpha, \pi, \delta$ are recursive, too. κ is recursive because of (C 3) and (S 2). Now, ϱ is p.r. and total because of the definition of κ, i.e. ϱ is recursive.

We want to prove

$$(\forall i, x)\,(\Psi(i, x) = \varrho[\bar{\Psi}, \kappa]^P \gamma(i, x)).$$

Case 1: $\Psi(i, x)$ defined. By (S 1) we, then, know

(6) $$\Phi_i(x) = T \quad \text{for some} \quad T \in N, \quad \text{and}$$

(7) $$\Phi_i(x) \neq t \quad \text{for} \quad t < T.$$

We show

(8) $$\kappa \bar{\Psi}^{(t)} \gamma(i, x) = 1 \quad \text{for} \quad t < T, \quad \text{and}$$

(9) $$\kappa \bar{\Psi}^{(T)} \gamma(i, x) = 0.$$

For this we notice that for $\xi \in M$

$$\xi = \bar{\Psi}^{(t)} \beta(j)$$

for certain $t, j \in N$, which, by (C 1) and (C 2), are uniquely determined, such that

(10) $$\alpha(\xi) = j,$$

(11) $$\tau(\xi) = t.$$

Now, we take $\xi = \bar{\Psi}^{(t)} \gamma(i, x) \in M$. Using (2), (3), (10), (11), we have

$$\pi(\xi) = i, \qquad \delta(\xi) = x, \qquad \tau(\xi) = t.$$

Hence, by (6), (7) and (4) we, finally, obtain (8) and (9). From (5), (8) and (9) we get

$$\varrho[\bar{\Psi}, \kappa]^P \gamma(i, x) = \varrho \bar{\Psi}^{(T)} \gamma(i, x) = \Psi(i, x).$$

Case 2: $\Psi(i, x)$ not defined, i.e.

$$(\forall t)\,(\Phi_i(x) \neq t).$$

By a computation analoguous to that in case 1 we get

$$(\forall t)\,(\kappa \bar{\Psi}^{(t)} \gamma(i, x) = 1).$$

Hence, $[\bar{\Psi}, \kappa]^P \gamma(i, x)$ and $\varrho[\bar{\Psi}, \kappa]^P \gamma(i, x)$ not defined.

Example 1: We take the Turing formalism (for notation see Davis [6]). Let ξ_i be the (Gödel number of a) state that consists of the Turing machine

$$p_i := \{q_0 B R q_0, q_1 B R q_1, \ldots, q_i B R q_i\},$$

the square number 0, the internal configuration q_0, and the tape inscription B ("blank"). The ξ_i are (intuitively) enumerable and, hence, after a Gödelization of the states will be recursively enumerable. It is clear that the effect of the computation started from the "states" ξ_i is the same for all i: remain in the internal configuration q_0 and move steadily to the right, i.e. $\bar{\Psi}_T^{(t)}(\xi_i)$ consists of the program p_i, the square number t, the internal configuration q_0, the tape inscription $BB\dots B$ ($t+1$ times "blank"). It is easily seen that (C 1) – (C 3) are satisfied. Similar "programs" may be written for the other computability formalisms. Always, (C 1) – (C 3) are very easy to check.

Theorem 4: If $\varrho, \bar{\Psi}, \kappa, \gamma$ determines a P-decomposition of a function ψ that describes a Gödel numbering then there exists an infinite sequence $\xi_0, \xi_1, \dots (\xi_i \in N)$ such that

(C 1) $(\forall i, t_1, t_2)(t_1 \neq t_2 \to \bar{\Psi}^{(t_1)}(\xi_i) \neq \bar{\Psi}^{(t_2)}(\xi_i))$,

(C 2) $(\forall i, k, t_1, t_2)(i \neq k \to \bar{\Psi}^{(t_1)}(\xi_i) \neq \bar{\Psi}^{(t_2)}(\xi_k))$.

Proof: Theorem 4 will be a consequence of the following

Lemma 5: Let $\varrho, \bar{\Psi}, \kappa, \gamma$ determine a P-decomposition of a function $\Psi \in P_2$. If there is a $\beta \in R_1$ such that

(C 4) $(\forall i, t)(\kappa \bar{\Psi}^{(t)} \gamma(\beta(i), 0) \neq 0)$

(C 5) $(\forall p)((\exists t)(\kappa \bar{\Psi}^{(t)} \gamma(p, 0) = 0)) \vee$
$(\exists t_1, t_2)(t_1 \neq t_2 \wedge \bar{\Psi}^{(t_1)} \gamma(p, 0) = \bar{\Psi}^{(t_2)} \gamma(p, 0)) \vee$
$(\exists i, t_1, t_2)(\bar{\Psi}^{(t_1)} \gamma(p, 0) = \bar{\Psi}^{(t_2)} \gamma(\beta(i), 0))$,

then the property "$\Psi(p, 0)$ defined" is decidable.

Proof: We assert that the following algorithm is a decision procedure for the property "$\Psi(p, 0)$ defined".

1: $v := 0$; $W := V := \emptyset$;
2: $W := W \cup \{\bar{\Psi}^{(v)} \gamma(p, 0)\}$;
$V := V \cup \{\bar{\Psi}^{\sigma_{21}(v)} \gamma(\beta \sigma_{22}(v), 0)\}$;
3: If $\kappa \bar{\Psi}^{(v)} \gamma(p, 0) = 0$ then answer "$\Psi(p, 0)$ defined" and stop; else go to 4;
4: If $\bar{\Psi}^{(v)} \gamma(p, 0) = \bar{\Psi}^{(v')} \gamma(p, 0)$ for some $v' < v$ then answer "$\Psi(p, 0)$ undefined" and stop; else go to 5;
5: If $W \cap V \neq \emptyset$ then answer "$\Psi(p, 0)$ undefined" and stop; else go to 6;
6: $v := v + 1$; go to 2.

Instead of a detailed formal proof of the assertion we give the idea: (C 5) guarantees that for every p the sequence $S := \{\bar{\Psi}^{(t)} \gamma(p, 0) \mid t \in N\}$ contains a terminal "state" or a "cycle" or has some element in common with one of the sequences $S_i := \{\bar{\Psi}^{(t)} \gamma(\beta(i), 0) \mid t \in N\}$. Exactly in the last two cases $\Psi(p, 0)$ will be undefined [in the third case this is because of (C 4)] provided that the first possibility is excluded. The algorithm systematically generates the sequences S and S_i. This is effectively possible because of the recursiveness of β. After finitely many steps the algorithm must detect one of the three cases because of (C 5). Some precaution is necessary in the proof to make clear that the algorithm will not detect a cycle in S or an overlapping with one of the sequences S_i and, thus, answer "$\Psi(p, 0)$ undefined" in the case where for some t $\kappa\bar{\Psi}^{(t)}\gamma(p, 0) = 0$ also.

Proof of Theorem 4 (continued): Now, let ϱ, $\bar{\Psi}$, κ, γ determine a P-decomposition of Ψ where Ψ describes a Gödel numbering and assume the conclusion of Theorem 4 to be false, i.e. let $X := \{\xi_1, \ldots, \xi_j\}$ be a ("maximal") set for which

(C 1') $(\forall i, t_1, t_2)(i \leqq j \wedge t_1 \neq t_2 \rightarrow \bar{\Psi}^{(t_1)}(\xi_i) \neq \bar{\Psi}^{(t_2)}(\xi_i))$,

(C 2') $(\forall i, k, t_1, t_2)(i, k \leqq j \wedge i \neq k \rightarrow \bar{\Psi}^{(t_1)}(\xi_i) \neq \bar{\Psi}^{(t_2)}(\xi_k))$,

and such that for arbitrary $\xi \in N$

$$(12) \qquad \begin{aligned} &(\exists t_1, t_2)(t_1 \neq t_2 \wedge \bar{\Psi}^{(t_1)}(\xi) = \bar{\Psi}^{(t_2)}(\xi)) \vee \\ &(\exists i, t_1, t_2)(i \leqq j \wedge \bar{\Psi}^{(t_1)}(\xi) = \bar{\Psi}^{(t_2)}(\xi_i)). \end{aligned}$$

We define

$$X_1 := \{\xi \mid \xi \in X \wedge (\exists t)(\forall \tau)(\tau \geqq t \rightarrow \kappa\bar{\Psi}^{(\tau)}(\xi) \neq 0),$$

$$X_2 := X - X_1,$$

$$P_i := \begin{cases} \emptyset, & \text{if } \xi_1 \in X_2 \vee i > j, \\ \{p \mid (\exists t_1, t_2)(\bar{\Psi}^{(t_1)}\gamma(p, 0) = \bar{\Psi}^{(t_2)}(\xi_i)) \wedge (\forall t)(\kappa\bar{\Psi}^{(t)}\gamma(p, 0) \neq 0)\}, & \text{otherwise}, \end{cases}$$

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

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

$\beta \in R_1$, 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 $\bar{p}$, $(\forall t)(\kappa\bar{\Psi}^{(t)}\gamma(\beta(i), 0) \neq 0)$, or $P_i \neq \emptyset$, hence $\beta(i) \in P_i$ and therefore by the definition of $P_i (\forall t)(\kappa\bar{\Psi}^{(t)}\gamma(\beta(i), 0) \neq 0)$.

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

$$(13) \qquad (\forall t)(\kappa\bar{\Psi}^{(t)}\gamma(p, 0) \neq 0),$$

$$(14) \qquad (\forall t_1, t_2)(t_1 \neq t_2 \rightarrow \bar{\Psi}^{(t_1)}\gamma(p, 0) \neq \bar{\Psi}^{(t_2)}\gamma(p, 0)),$$

$$(15) \qquad (\forall i, t_1, t_2)(\bar{\Psi}^{(t_1)}\gamma(p, 0) \neq \bar{\Psi}^{(t_2)}\gamma(\beta(i), 0)).$$

We shall show

$$(16) \qquad (\forall i, t_1, t_2)(i \leqq j \rightarrow \bar{\Psi}^{(t_1)}\gamma(p, 0) \neq \bar{\Psi}^{(t_2)}(\xi_i)).$$

Take, first, $\xi_i \in X_2$, then $(\forall t_1, t_2)(\bar{\Psi}^{(t_1)}\gamma(p, 0) \neq \bar{\Psi}^{(t_2)}(\xi_i))$, because otherwise $(\exists t)(\kappa\bar{\Psi}^{(t)}\gamma(p, 0) = 0)$, which contradicts (13). If, on the other hand, $\xi_i \in X_1$, then $(\exists t_1, t_2)(\bar{\Psi}^{(t_1)}\gamma(p, 0) = \bar{\Psi}^{(t_2)}(\xi_i))$ would imply $p \in P_i$, hence $(\exists s_1, s_2)(\bar{\Psi}^{(s_1)}\gamma(p, 0) = \bar{\Psi}^{(s_2)}\gamma(\beta(i), 0))$, which contradicts (15). Thus, under the assumptions (13–(15), (16) is proven. (14) together with (16) is a contradiction to (12), hence (C 5).

However, (C 4) and (C 5) true would make "$\Psi(p, 0)$ defined" a decidable property (by Lemma 5). This contradicts Rice's Theorem (Rogers [12]). Thus, we have to reject (12), i.e. the conclusion of Theorem 4 is true.

Remark: Though, from the examples, there is much evidence to conjecture that in Theorem 4 suitable ξ_i can always be effectively enumerated and that (C 3) is also a necessary condition, we were not able to prove this under the very general assumptions on P-decompositions.

In any case, Theorem 4 may serve as a useful tool for showing that some computability formalism does not possess the necessary power to compute all p.r.f. even if one admits very powerful input/output functions.
There are many equivalent forms of the condition (C 1) $\wedge$ (C 2) $\wedge$ (C 3) which more readily 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. Maybe, that it is some interesting insight into the essence of "computation" that this very general principle under 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 function Ψ that describes a Gödel numbering.
From the experience with programming languages one might conjecture that a $\bar{\Psi}$ which, together with some ϱ, κ, γ determines a P-decomposition of a Ψ that describes a Gödel numbering must "allow cycles", i.e.

$$(\exists \xi, t_1, t_2)\,(t_1 \neq t_2 \wedge \bar{\Psi}^{(t_1)}(\xi) = \bar{\Psi}^{(t_2)}(\xi))\,.$$

However, this is not necessarily so, as can be seen by inspection of the special $\bar{\Psi}$ constructed in the proof of Theorem 2, second part.
Finally, we give some propositions which are of minor interest. They deal with the possible output functions ϱ. The first of them shows that we can't get by without any ϱ.

Proposition 6: If $\varrho, \bar{\Psi}, \kappa, \gamma$ determine a P-decomposition of a function Ψ that describes a Gödel numbering then ϱ can't be $1-1$.

Proof: Assume ϱ to be $1-1$ and take i, x such that $\Psi(i, x)$ is undefined, i.e. $(\forall t)\,(\kappa \bar{\Psi}^{(t)} \gamma(i, x) \neq 0)$. Let $\varrho\gamma(i, x) = r$. Since $\xi = \gamma(i, x)$ is the only ξ such that $\varrho(\xi) = r$, but $\kappa(\xi) \neq 0$, there can't be any i', x' such that $\Psi(i', x') = r$. This is, of course, impossible.

From a paper by Markov [9] on the possible "output" functions in Kleene's normal form theorem (Davis [6], p. 63) and from the concrete examples one might conjecture that with P-decompositions, too, the possible output functions are exactly those of large oscillation. This is only partly true (Proposition 7 and Example 2).

Proposition 7: If σ_{21} is of large oscillation, then one can find $\bar{\Psi}, \kappa, \gamma$ such that $\sigma_{21}, \bar{\Psi}, \kappa, \gamma$ determine a P-decomposition of a function Ψ, where Ψ describes a Gödel numbering.

Proof: If $\sigma_{21} \in R_1$ is of large oscillation (i.e. $(\forall y, z)(\exists x)(x > z \wedge \sigma_{21}(x) = y)$) then one can find $\sigma_{22} \in R_1, \sigma_2 \in R_2$ such that $\sigma_2, \sigma_{21}, \sigma_{22}$ are pairing functions (following Markov [9]). Take some $\varrho', \bar{\Psi}', \kappa', \gamma'$ which determine a P-decomposition of a function Ψ that describes a Gödel numbering. Then $\bar{\Psi}, \kappa, \gamma$, defined by

$$\gamma(i, x) = \sigma_2(\varrho'\gamma'(i, x), \gamma'(i, x)),$$
$$\bar{\Psi}(\xi) = \sigma_2(\varrho'\bar{\Psi}'\sigma_{22}(\xi), \bar{\Psi}'\sigma_{22}(\xi)),$$
$$\kappa(\xi) = \begin{cases} 0, & \text{if} \quad \kappa'\sigma_{22}(\xi) = 0, \\ 1, & \text{otherwise}, \end{cases}$$

are such that $\Psi(i, x) = \sigma_{21}[\bar{\Psi}, k]^P \gamma(i, x)$.

Example 2: We give an example of functions $\varrho', \bar{\Psi}', \kappa', \gamma$ which determine a P-decomposition of a function Ψ that describes a Gödel numbering, where ϱ' is not of large oscillation. Let some $\varrho, \bar{\Psi}, \kappa, \gamma$ determine a P-decomposition of Ψ. Assume Ψ to describe a Gödel numbering. We define

$$\bar{\Psi}'(\xi) = \begin{cases} \Psi(\xi), & \text{if} \quad \kappa(\xi) \neq 0, \\ (\mu\eta)(\kappa(\eta) = 0 \wedge \varrho(\eta) = \varrho(\xi)), & \text{otherwise}, \end{cases}$$

$$\kappa'(\xi) = \begin{cases} \kappa(\xi), & \text{if} \quad \kappa(\xi) \neq 0, \\ 1, & \text{if} \quad \kappa(\xi) = 0 \wedge \xi \neq (\mu\eta)(\kappa(\eta) = 0 \wedge \varrho(\eta) = \varrho(\xi)), \\ 0, & \text{otherwise}, \end{cases}$$

$$\varrho'(\xi) = \begin{cases} \varrho(\xi), & \text{if} \quad \kappa'(\xi) = 0, \\ 0, & \text{otherwise}. \end{cases}$$

It is easy to show that $\varrho', \bar{\Psi}', \kappa', \gamma$ determine a P-decomposition of the same Ψ. However, for $r > 0, \varrho'(\xi) = r$ holds exactly for one ξ which shows that ϱ' is not of large oscillation.

6. Conclusions

We hope that we have been able to give an idea how a detailed study of P-decompositions of Gödel numberings could add to our insight into the essence of computability formalisms (programming languages, computer concepts).

A better understanding of the power contained in one step of a computability formalism should also help in further axiomatizations of recursive function theory. Our notion of "step" is approximately on the same level of abstraction as Blum's notion.

In a further study of P-decompositions it would be especially interesting to find a simple two-place relation K such that: $K(\bar{\Psi}, \kappa)$ if and only if there exist ϱ, γ such that $\varrho, \bar{\Psi}, \kappa, \gamma$ determine a P-decomposition of some Gödel numbering Ψ. The isolation of $\bar{\Psi}$ in our Theorem 3 is not quite natural since presenting a computability formalism means giving $\bar{\Psi}$ and κ. Thus, κ can not be chosen freely any more.

REFERENCES

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

[2] Buchberger, B.: Associating functions and the operator of conditioned iteration (Russian). Communications of the JINR-Dubna, P5-5788, 1971 (english translation: Bericht Nr. 71-1. Institut f. num. Math., Univ. Innsbruck).

[3] Buchberger, B.: A basic problem in theory of programming languages. Bericht Nr. 72-1. Institut für num. Math., Univ. Innsbruck, 1972.

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

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

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

[7] Ershov, A. P.: Operator algorithms I. Problemy kibernetiki 3, 1960.

[8] Lucas, P., Lauer, P., Stigleitner, H.: Method and notation for the formal definition of programming languages. IBM Laboratory Vienna. Technical Report TR 25.087, 1968.

[9] Markov, A. A.: On the representation of recursive functions (Russian). Doklady akademii nauk SSSR, n.s., vol. 58, p. 1891—1892, 1947.

[10] Nepomniashy, V. A.: Conditions for the algorithmic completeness of the systems of operations. In: Proceedings of the IFIP Congress 71. Ljubljana, 1971.

[11] Rogers, H. Jr.: Gödel numberings of partial recursive functions. The journal of symbolic logic 23/3, 1958.

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

[13] Strong, H. R.: Algebraically generalized recursive function theory. IBM journal of research and development 12/6, 1968.

[14] Uspenskii, V. A.: Lectures on computable functions (Russian). Gos. izdat. fiz.-mat. lit., Moscow, 1960 (French translation: V. A. Ouspenski, Leçons sur les fonctions calculables, Hermann, Paris, 1966).