

# RECURSIVE FUNCTIONALS AND QUANTIFIERS OF FINITE TYPES I

BY

S. C. KLEENE

This is the third of a series of papers in these Transactions on hierarchies obtained by quantifying variables of recursive predicates. In *Recursive predicates and quantifiers* [10] variables for natural numbers (type 0) were quantified, and in *Arithmetical predicates and function quantifiers* [14] also variables for one-place number-theoretic functions (type 1). In the present paper we shall quantify also variables for functions (i.e. functionals) of higher finite types (types 2, 3, 4, . . .). A theory of recursive functions and predicates of variables of these types has not previously been developed. For the hierarchy results in the form that new predicates are definable by increasing the highest type of variable quantified<sup>(1)</sup>, or the number of alternations of the quantifiers of the highest type, it would suffice to extend the notion of primitive recursiveness to the higher types of variables. This is how we began the investigation in 1952<sup>(2)</sup>. However there are situations where the question "What becomes of this theory for higher types?" calls for an extension of general and partial recursiveness. For example, before we can extend Post's notion of degree of unsolvability [25; 19], which proved fruitful in the study of hierarchies of number-theoretic predicates [5; 14; 16; 26], to predicates with type-1 variables, we must have a notion of relative general recursiveness for such predicates, which when uniform amounts to having general recursiveness for functions with type-2 variables. Accordingly we now give an extension of general and partial recursiveness. The treatment entails incidentally a somewhat new treatment of general and partial recursive functions of variables of types 0, 1. As will appear, many of the known results for types 0, 1 extend to the higher types, but not all. In this Part I we leave undiscussed various aspects of the subject on which work is in progress or completed which we hope to report in a Part II.

---

Presented to the Society August 30, 1957; received by the editors June 13, 1957.

<sup>(1)</sup> This result was obtained by Tarski [28], using set variables. Also cf. [29].

<sup>(2)</sup> Cf. XXXIX below. This paper incorporates (with a new method of proof) the results alluded to in the last sentence of the abstract of [14] as presented to the Association for Symbolic Logic on December 29, 1952 (J. Symb. Logic vol. 18 (1953) p. 190), and is the paper referred to in [14, p. 312] and [16, p. 212] as to be written under the title *Analytic predicates and function quantifiers of higher finite types*. The term "analytic" was applied in [14] to the predicates obtained by quantifying variables of types  $\leq 1$ , and we have meanwhile decided that it would invite confusion to extend the use of "analytic" to include quantification of higher-type variables; hence the present change of title.

**1. Primitive recursive functions. 1.1.** We shall cite our *Introduction to metamathematics* [13] simply as IM, and follow it in terminology and notation (cf. bottom p. 538) except as otherwise specified<sup>(3)</sup>.

**1.2.** The objects of *type 0* are the natural numbers  $0, 1, 2, \dots$ . As type-0 variables, i.e. variables ranging over the natural numbers, we use  $a, b, c, \dots$ ,  $a_1, a_2, \dots$  or  $\alpha^0, \beta^0, \gamma^0, \dots$ ,  $\alpha_1^0, \alpha_2^0, \dots$ , etc.

For each  $j > 0$ , the objects of *type j* are the one-place functions from type- $j-1$  objects to natural numbers<sup>(4)</sup>. As type- $j$  variables, i.e. variables ranging over the type- $j$  objects, we use  $\alpha^j, \beta^j, \gamma^j, \dots$ ,  $\alpha_1^j, \alpha_2^j, \dots$ , etc. (For  $j > 1$ , they are "functional" variables.)

In illustrations, we may use simply  $\alpha, \beta, \gamma, \dots$ ,  $\alpha_1, \alpha_2, \dots$  for type-1 variables (as in [12; 14; 16]); and **F, G, H, ...** for type-2 variables.

Sometimes we omit the type index on the letters  $\alpha^j, \beta^j, \gamma^j, \dots$  used as type- $j$  variables in contexts where the type should be clear (e.g.  $\alpha^5(\beta^4)$  can be written  $\alpha^5(\beta)$  or  $\alpha(\beta^4)$  without ambiguity). One must then not confuse  $\alpha, \beta, \gamma, \dots$  for variables of types shown by the context with  $\alpha, \beta, \gamma, \dots$  for type-1 variables.

Letters like  $\phi, \psi, \chi, \dots$  will be used (as in [12; 14] with types 0, 1) for functions of a given finite number of variables, each variable being of a specified one of our types, taking a natural number as value. (These functions are "functionals," when any of their variables are of type  $> 0$ .)

**1.3.** In the theory of general and partial recursive functions of variables of types 0, 1 a useful role was played from the beginning by the primitive recursive functions (cf. e.g. [8], IM). They constitute a subclass of the general recursive functions, each function of which subclass can be calculated for given arguments by steps which can be proved by known reasoning to terminate; and they can be enumerated effectively. Subsequent research showed that smaller classes of functions can play the same role, e.g. the elementary functions of Csillag and Kalmár (cf. IM, p. 285). Such results are of interest, e.g. for type 0 as a possible line of approach toward a negative solution of Hilbert's tenth problem (1900). However for our purposes the largeness of the class of primitive recursive functions has seemed rather convenient. Accordingly we shall begin the theory of recursive functions of variables of types 0, 1, 2,  $\dots$  with them<sup>(5)</sup>, leaving open the question what smaller classes of functions could play the same role.

<sup>(3)</sup> Several notations from papers subsequent to IM are:  $\bar{\alpha}(x)$  for  $\prod_{i < x} p_i^{\alpha(i)+1}$  (cf. [12; 14, Footnote 2]),  $\text{Seq}(w)$  for  $w \neq 0$  &  $(i)_{i < \text{lh}(w)} [(w)_i \neq 0]$  (cf. [15, p. 416]),  $\text{Ext}(w, u)$  for  $\text{Seq}(w)$  &  $(Ex)_{x \leq \text{lh}(w)} [u = \prod_{i < x} p_i^{(w)_i}]$  (cf. Spector [27, p. 588]).

<sup>(4)</sup> I.e., arbitrary such functions, not merely recursive ones. For the type-1 case, cf. [12, p. 683].

<sup>(5)</sup> Our extension of primitive recursiveness to allow higher types of variables does not alter the notion for functions of variables of types 0, 1 (by VIII below). What Péter calls a "primitive recursion on the second level (II-te Stufe)" XXIV below can be used to define number-theoretic functions that are not primitive recursive (cf. [24, pp. 247-248, 252, 256] and [23, pp. 68, 97-99]).

For each application of the following schemata S1–S8,  $\mathfrak{b}$  is any list (possibly empty) of variables, distinct from one another and from the other variables of the schema, and each of a specified one of our types. Furthermore,  $a$  is a number variable,  $\alpha$  is a function variable of type 1,  $q$  is a given natural number,  $\psi$  and  $\chi$  are given functions of the indicated variables, and  $\phi$  is the function being defined. For S6,  $\mathfrak{a}_1$  is a list of distinct variables, containing at least  $k+1$  of type  $j$ , from which  $a$  results by moving the  $(k+1)$ st type- $j$  variable to the front of the list. The expressions shown at the right will be explained in §3.

S1	$\phi(a, \mathfrak{b}) = a' = a+1$	$\langle 1, \langle n_0, \dots, n_r \rangle \rangle$ .
S2	$\phi(\mathfrak{b}) = q$	$\langle 2, \langle n_0, \dots, n_r \rangle, q \rangle$ .
S3	$\phi(a, \mathfrak{b}) = a$	$\langle 3, \langle n_0, \dots, n_r \rangle \rangle$ .
S4	$\phi(\mathfrak{b}) = \psi(\chi(\mathfrak{b}), \mathfrak{b})$	$\langle 4, \langle n_0, \dots, n_r \rangle, g, h \rangle$ .
S5	$\left\{ \begin{array}{l} \phi(0, \mathfrak{b}) = \psi(\mathfrak{b}), \\ \phi(a', \mathfrak{b}) = \chi(a, \phi(a, \mathfrak{b}), \mathfrak{b}) \end{array} \right.$	$\langle 5, \langle n_0, \dots, n_r \rangle, g, h \rangle$ .
S6	$\phi(a) = \psi(\mathfrak{a}_1)$	$\langle 6, \langle n_0, \dots, n_r \rangle, j, k, g \rangle$ .
S7	$\phi(\alpha, a, \mathfrak{b}) = \alpha(a)$	$\langle 7, \langle n_0, \dots, n_r \rangle \rangle$ .
S8	$\phi(\alpha^j, \mathfrak{b}) = \alpha^j(\lambda \alpha^{j-2} \chi(\alpha^j, \alpha^{j-2}, \mathfrak{b}))$	$\langle 8, \langle n_0, \dots, n_r \rangle, j, h \rangle$ .

S6, S7, S8 may also be designated as S6. $j$  ( $j \geq 0$ ), S7.1, S8. $j$  ( $j \geq 2$ ).

The schemata will be used under the convention that only the order of listing the variables within each type is material. For example,  $\phi(a, b, \alpha) = a+1$ ,  $\phi(a, \alpha, b) = a+1$ ,  $\phi(\alpha, a, b) = a+1$  are all admissible as applications of S1, but not  $\phi(b, a, \alpha) = a+1$  (though this function can be introduced by an application of S1 followed by one of S6).

A function  $\phi(a)$  is *primitive recursive*, if there is a *primitive recursive description* (analogous to IM, p. 220) of it in terms of S1–S8 used under the stated convention.

**1.4.** We call a primitive recursive description *irredundant*, if in it each function except the last is used as the  $\psi$  or  $\chi$  of a later schema application. By the *maximum type* of  $a$  we mean the greatest of the types of the variables  $a$  if  $a$  is nonempty (0 if  $a$  is empty).

I. In an irredundant primitive recursive description of a function  $\phi(a)$  with  $r$  the maximum type of  $a$ , each function has the same maximum type  $r$  of its variables and for  $r \geq 2$  ( $r = 1$ ) the same variables of types  $r, r-1$  (type  $r$ ), and hence S6. $j$ , S7. $j$ , S8. $j$  can be used only for  $j \leq r$ .

**Proof.** In an application of S4, S5, S6 or S8, the  $\phi$  has no variables of types  $> 0$  which are not variables of the  $\psi$  and  $\chi$ , and lacks no such variables of the  $\psi$  or  $\chi$  other than the variable  $\alpha^{j-2}$  of the  $\chi$  in S8.

II. If  $\phi(a)$  is primitive recursive (with a given description), and  $a'$  comes from  $a$  by a permutation (without repetitions or omissions), then  $\phi'(a') = \phi(a)$

is also primitive recursive (with a description obtainable from the given one by suffixing applications of S6).

**Proof.** Inversely to S6,  $k$  successive applications of S6 move a variable from the first to the  $(k+1)$ st position among the type- $j$  variables. By S6 with this inverse (for different  $k$ 's), any two type- $j$  variables can be interchanged.

III. If  $\phi(a)$  is primitive recursive (with a given description), then  $\phi(a, c) = \phi(a)$ , where  $a, c$  are distinct variables, is primitive recursive (with a description consisting of applications of the same respective schemata as the given one).

**Proof.** In the given description of  $\phi(a)$  any variables not in  $a$  but in  $c$  can be changed to new distinct variables. Then the variables  $c$  can be introduced at each application of S1-S3, S7 and can be carried through each application of S4-S6, S8.

1.5. Although our schemata as stated introduce functions  $\phi(a)$  with values of type 0, we can get functions with values of any higher type  $j$  by introducing  $\phi(a, \alpha^{j-1})$  and considering  $\lambda\alpha^{j-1}\phi(a, \alpha^{j-1})$  as a function of  $a$ . The latter we call *primitive recursive* when  $\phi(a, \alpha^{j-1})$  is<sup>(6)</sup>.

IV. For each  $n \geq 1$ , if  $\phi(a, \sigma^n, b)$  and  $\theta(a, b, \tau^{n-1})$  are primitive recursive, so is  $\phi(a, b) = \phi(a, \lambda\tau^{n-1}\theta(a, b, \tau^{n-1}), b)$ .

**Proof.** First replace in IV  $\theta(a, b, \tau^{n-1})$ ,  $\phi(a, b)$  by  $\theta(a, b, c, \tau^{n-1})$ ,  $\phi(a, b, c)$ . The resulting proposition (of which IV is the case for  $c$  empty) we prove by induction on  $n$ , and, within that, induction on the length  $l$  of a given primitive recursive description of  $\phi(a, \sigma^n, b)$ . Cases are numbered according to the schema last applied in that description. We treat in detail the two cases requiring the most care, and summarize the others.

CASE 5.  $\phi(a, \sigma, b)$  is introduced by S5 thus, writing  $a = (a, b)$ ,

$$\begin{cases} \phi(0, b, \sigma, b) = \psi(b, \sigma, b), \\ \phi(a', b, \sigma, b) = \chi(a, \phi(a, b, \sigma, b), b, \sigma, b), \end{cases}$$

where  $\psi, \chi$  are previously introduced primitive recursive functions. We need to show that  $\phi(a, b, b, c) = \phi(a, b, \lambda\tau\theta(a, b, b, c, \tau), b)$ , where  $\theta(a, b, b, c, \tau)$  is primitive recursive, is primitive recursive. Using II and III, so are

$$\begin{aligned} \theta_1(b, b, c, u, \tau) &= \theta(u, b, b, c, \tau), \\ \theta(a, b, b, b, c, u, \tau) &= \theta(u, b, b, c, \tau). \end{aligned}$$

By the hypothesis of the induction on  $l$ ,

$$\begin{aligned} \psi(b, b, c, u) &= \psi(b, \lambda\tau\theta_1(b, b, c, u, \tau), b), \\ \chi(a, b, b, b, c, u) &= \chi(a, b, b, \lambda\tau\theta(a, b, b, b, c, u, \tau), b) \end{aligned}$$

are primitive recursive. As a new application of S5, let

<sup>(6)</sup> More generally, a function of an arbitrary one of the "finite types" considered in [18, §5] shall be *primitive recursive*, if the function of "special type" correlated to it there is primitive recursive under 1.3 here.

$$\begin{cases} \phi(0, \mathfrak{d}, \mathfrak{b}, \mathfrak{c}, u) = \psi(\mathfrak{d}, \mathfrak{b}, \mathfrak{c}, u), \\ \phi(a', \mathfrak{d}, \mathfrak{b}, \mathfrak{c}, u) = \chi(a, \phi(a, \mathfrak{d}, \mathfrak{b}, \mathfrak{c}, u), \mathfrak{d}, \mathfrak{b}, \mathfrak{c}, u). \end{cases}$$

Using S6, S3, S4, let

$$\begin{aligned} \phi_1(u, a, \mathfrak{d}, \mathfrak{b}, \mathfrak{c}) &= \phi(a, \mathfrak{d}, \mathfrak{b}, \mathfrak{c}, u), \\ \eta(a, \mathfrak{d}, \mathfrak{b}, \mathfrak{c}) &= a, \\ \phi(a, \mathfrak{d}, \mathfrak{b}, \mathfrak{c}) &= \phi_1(\eta(a, \mathfrak{d}, \mathfrak{b}, \mathfrak{c}), a, \mathfrak{d}, \mathfrak{b}, \mathfrak{c}) = \phi(a, \mathfrak{d}, \mathfrak{b}, \mathfrak{c}, a). \end{aligned}$$

To see that then  $\phi(a, \mathfrak{d}, \mathfrak{b}, \mathfrak{c}) = \phi(a, \mathfrak{d}, \lambda\tau\theta(a, \mathfrak{d}, \mathfrak{b}, \mathfrak{c}, \tau), \mathfrak{b})$  as required, we prove by induction on  $a$  that, for any fixed  $\mathfrak{d}, \mathfrak{b}, \mathfrak{c}, u$ ,

$$(1) \quad \phi(a, \mathfrak{d}, \mathfrak{b}, \mathfrak{c}, u) = \phi(a, \mathfrak{d}, \lambda\tau\theta(u, \mathfrak{d}, \mathfrak{b}, \mathfrak{c}, \tau), \mathfrak{b}).$$

$$\begin{aligned} \text{BASIS. } \phi(0, \mathfrak{d}, \mathfrak{b}, \mathfrak{c}, u) &= \psi(\mathfrak{d}, \mathfrak{b}, \mathfrak{c}, u) = \psi(\mathfrak{d}, \lambda\tau\theta_1(\mathfrak{d}, \mathfrak{b}, \mathfrak{c}, u, \tau), \mathfrak{b}) \\ &= \psi(\mathfrak{d}, \lambda\tau\theta(u, \mathfrak{d}, \mathfrak{b}, \mathfrak{c}, \tau), \mathfrak{b}) = \phi(0, \mathfrak{d}, \lambda\tau\theta(u, \mathfrak{d}, \mathfrak{b}, \mathfrak{c}, \tau), \mathfrak{b}). \end{aligned}$$

$$\begin{aligned} \text{IND. STEP. } \phi(a', \mathfrak{d}, \mathfrak{b}, \mathfrak{c}, u) &= \chi(a, \phi(a, \mathfrak{d}, \mathfrak{b}, \mathfrak{c}, u), \mathfrak{d}, \mathfrak{b}, \mathfrak{c}, u) \\ &= \chi(a, \phi(a, \mathfrak{d}, \lambda\tau\theta(u, \mathfrak{d}, \mathfrak{b}, \mathfrak{c}, \tau), \mathfrak{b}), \mathfrak{d}, \mathfrak{b}, \mathfrak{c}, u) \text{ (by hyp. ind. on } a) \\ &= \chi(a, \phi(a, \mathfrak{d}, \lambda\tau\theta(u, \mathfrak{d}, \mathfrak{b}, \mathfrak{c}, \tau), \mathfrak{b}), \mathfrak{d}, \lambda\tau\theta(u, \mathfrak{d}, \mathfrak{b}, \mathfrak{c}, \tau), \mathfrak{b}) \\ &= \phi(a', \mathfrak{d}, \lambda\tau\theta(u, \mathfrak{d}, \mathfrak{b}, \mathfrak{c}, \tau), \mathfrak{b}). \end{aligned}$$

CASE 8. SUBCASE 1.  $\sigma$  is  $\alpha^i$ . Then  $n = j$ . We have

$$\phi(\sigma, \mathfrak{b}) = \sigma(\lambda\alpha^{i-2} \chi(\sigma, \alpha^{i-2}, \mathfrak{b}))$$

with  $\chi$  primitive recursive, and need to obtain primitive recursively

$$\begin{aligned} \phi(\mathfrak{b}, \mathfrak{c}) &= \{\lambda\tau\theta(\mathfrak{b}, \mathfrak{c}, \tau)\}(\lambda\alpha^{i-2} \chi(\lambda\tau\theta(\mathfrak{b}, \mathfrak{c}, \tau), \alpha^{i-2}, \mathfrak{b})) \\ &= \theta(\mathfrak{b}, \mathfrak{c}, \lambda\alpha^{i-2} \chi(\lambda\tau\theta(\mathfrak{b}, \mathfrak{c}, \tau), \alpha^{i-2}, \mathfrak{b})). \end{aligned}$$

By III and S6, we can express  $\theta(\mathfrak{b}, \mathfrak{c}, \tau)$  as  $\theta(\alpha^{i-2}, \mathfrak{b}, \mathfrak{c}, \tau)$ . By the hyp. ind. on  $l$ ,  $\chi(\alpha^{i-2}, \mathfrak{b}, \mathfrak{c}) = \chi(\lambda\tau\theta(\alpha^{i-2}, \mathfrak{b}, \mathfrak{c}, \tau), \alpha^{i-2}, \mathfrak{b}) = \chi(\lambda\tau\theta(\mathfrak{b}, \mathfrak{c}, \tau), \alpha^{i-2}, \mathfrak{b})$  is primitive recursive, and thence using II, so is  $\chi_1(\mathfrak{b}, \mathfrak{c}, \alpha^{i-2}) = \chi(\alpha^{i-2}, \mathfrak{b}, \mathfrak{c})$ . Finally, by the hyp. ind. on  $n$ ,

$$\begin{aligned} \theta(\mathfrak{b}, \mathfrak{c}, \lambda\alpha^{i-2} \chi_1(\mathfrak{b}, \mathfrak{c}, \alpha^{i-2})) &= \theta(\mathfrak{b}, \mathfrak{c}, \lambda\alpha^{i-2} \chi(\alpha^{i-2}, \mathfrak{b}, \mathfrak{c})) \\ &= \theta(\mathfrak{b}, \mathfrak{c}, \lambda\alpha^{i-2} \chi(\lambda\tau\theta(\mathfrak{b}, \mathfrak{c}, \tau), \alpha^{i-2}, \mathfrak{b})) = \phi(\mathfrak{b}, \mathfrak{c}) \end{aligned}$$

is primitive recursive. SUBCASE 2.  $\sigma$  is not  $\alpha^i$ . Writing  $\mathfrak{a} = (\alpha^i, \mathfrak{d})$ , we express  $\theta(\alpha^i, \mathfrak{d}, \mathfrak{b}, \mathfrak{c}, \tau)$  as  $\theta(\alpha^i, \alpha^{i-2}, \mathfrak{d}, \mathfrak{b}, \mathfrak{c}, \tau)$ , apply the hyp. ind. on  $l$  to introduce  $\chi(\alpha^i, \alpha^{i-2}, \mathfrak{d}, \mathfrak{b}, \mathfrak{c})$ , and make a new application of S8 to introduce  $\phi(\alpha^i, \mathfrak{d}, \mathfrak{b}, \mathfrak{c})$ .

CASE 7. SUBCASE 1.  $\sigma$  is  $\alpha$ . Writing  $\mathfrak{b} = (a, \mathfrak{d})$ , we need to get primitive recursively  $\phi(a, \mathfrak{d}, \mathfrak{c}) = \{\lambda\tau\theta(a, \mathfrak{d}, \mathfrak{c}, \tau)\}(a) = \theta(a, \mathfrak{d}, \mathfrak{c}, a)$ . But from  $\theta(a, \mathfrak{d}, \mathfrak{c}, \tau)$ , since  $\tau$  is of type 0, we can obtain  $\theta(a, \mathfrak{d}, \mathfrak{c}, a)$  by S6, S3, S4 (cf. Case 5).

CASES 1, 2, 3 and CASE 7 SUBCASE 2. Simply use a new application of the same schema omitting  $\sigma$  and adding  $\mathfrak{c}$  as variables  $\mathfrak{b}$ .

CASE 6. SUBCASE 1.  $\sigma$  is the  $(k+1)$ st type- $j$  variable for  $\psi$ . Then  $\phi(\sigma, \mathfrak{b}) = \psi(\mathfrak{b}_1, \sigma, \mathfrak{b}_2)$  where  $\mathfrak{b} = (\mathfrak{b}_1, \mathfrak{b}_2)$ . By hyp. ind. on  $l$ ,  $\psi(\mathfrak{b}, \mathfrak{c}) = \psi(\mathfrak{b}_1, \lambda\tau\theta(\mathfrak{b}, \mathfrak{c}, \tau), \mathfrak{b}_2)$

is primitive recursive, and this is  $\phi(\mathfrak{b}, \mathfrak{c})$ . SUBCASE 2. otherwise. Then  $\phi(\mathfrak{a}, \sigma, \mathfrak{b}) = \psi(\mathfrak{a}_1, \sigma, \mathfrak{b}_1)$  where  $(\mathfrak{a}_1, \mathfrak{b}_1)$  is a permutation of  $(\mathfrak{a}, \mathfrak{b})$ . By II,  $\theta(\mathfrak{a}, \mathfrak{b}, \mathfrak{c}, \tau)$  can be expressed as  $\theta_1(\mathfrak{a}_1, \mathfrak{b}_1, \mathfrak{c}, \tau)$ . Apply the hyp. ind. on  $l$  to  $\psi$ , and use S6.

CASE 4. Express  $\theta(\mathfrak{a}, \mathfrak{b}, \mathfrak{c}, \tau)$  as  $\theta(\mathfrak{b}, \mathfrak{a}, \mathfrak{b}, \mathfrak{c}, \tau)$ , apply the hyp. ind. on  $l$  to  $\psi$  and  $\chi$ , and use S4.

1.6. By a *full substitution* we mean one in which, for each one  $\beta^j$  of the  $m$  variables of a function  $\psi(\mathfrak{b})$ , there is substituted a function of the same list  $\mathfrak{a}$  of variables (formed when  $j > 0$  by use of the  $\lambda$ -operator). For example, a full substitution into  $\psi(\mathfrak{a}, \alpha, \mathbf{F})$  gives  $\phi(\mathfrak{a}) = \psi(\chi_1(\mathfrak{a}), \lambda t \chi_2(\mathfrak{a}, t), \lambda \sigma \chi_3(\mathfrak{a}, \sigma))$ .

V. *The class of primitive recursive functions is closed under full substitution.*

**Proof.** If  $\mathfrak{b}$  includes variables from the list  $\mathfrak{a}$ , first change them to new variables throughout a given primitive recursive description of  $\psi$ . Now we illustrate using the above example (supposing  $\mathfrak{a}, \mathfrak{b}$  distinct). Using III and II, we can express  $\psi(\mathfrak{a}, \alpha, \mathbf{F})$ ,  $\chi_1(\mathfrak{a})$ ,  $\chi_2(\mathfrak{a}, t)$  as  $\psi(\mathfrak{a}, \alpha, \mathbf{F}, \mathfrak{a})$ ,  $\chi_1(\alpha, \mathbf{F}, \mathfrak{a})$ ,  $\chi_2(\mathbf{F}, \mathfrak{a}, t)$ . Now, by one application of S4 and two of IV, we obtain successively

$$\begin{aligned} \psi_1(\alpha, \mathbf{F}, \mathfrak{a}) &= \psi(\chi_1(\alpha, \mathbf{F}, \mathfrak{a}), \alpha, \mathbf{F}, \mathfrak{a}) = \psi(\chi_1(\mathfrak{a}), \alpha, \mathbf{F}), \\ \psi_2(\mathbf{F}, \mathfrak{a}) &= \psi_1(\lambda t \chi_2(\mathbf{F}, \mathfrak{a}, t), \mathbf{F}, \mathfrak{a}) = \psi(\chi_1(\mathfrak{a}), \lambda t \chi_2(\mathfrak{a}, t), \mathbf{F}), \\ \phi(\mathfrak{a}) &= \psi_2(\lambda \sigma \chi_3(\mathfrak{a}, \sigma), \mathfrak{a}) = \psi(\chi_1(\mathfrak{a}), \lambda t \chi_2(\mathfrak{a}, t), \lambda \sigma \chi_3(\mathfrak{a}, \sigma)). \end{aligned}$$

VI. *For each  $j \geq 1$ , the function  $\phi(\alpha^j, \alpha^{j-1}, \mathfrak{b}) = \alpha^j(\alpha^{j-1})$  is primitive recursive.*

**Proof,** by induction on  $j$ . BASIS ( $j = 1$ ). Use S7. IND STEP. By hyp. ind., III and II,  $\phi(\alpha^{j+1}, \alpha^{j-1}, \alpha^j, \mathfrak{b}) = \alpha^j(\alpha^{j-1})$  is primitive recursive. Using S8.j+1, so is  $\phi(\alpha^{j+1}, \alpha^j, \mathfrak{b}) = \alpha^{j+1}(\lambda \alpha^{j-1} \phi(\alpha^{j+1}, \alpha^{j-1}, \alpha^j, \mathfrak{b})) = \alpha^{j+1}(\lambda \alpha^{j-1} \alpha^j(\alpha^{j-1})) = \alpha^{j+1}(\alpha^j)$ .

REMARK 1. Consider the following schemata, for  $j \geq 1$ :

$$S4.j \phi(\mathfrak{b}) = \psi(\lambda \alpha^{j-1} \chi(\mathfrak{b}, \alpha^{j-1}), \mathfrak{b}). \quad S7.j \phi(\alpha^j, \alpha^{j-1}, \mathfrak{b}) = \alpha^j(\alpha^{j-1}).$$

(S4 = S4.0, S7 = S7.1.) Using IV and VI, (S1-S7, S8.2-S8.r) = (S1-S7, S4.1-S4.r - 1, S7.2-S7.r), and S4.r is derivable from either list.

VII. *The class of primitive recursive functions is closed under explicit definition, using (besides given functions, constant natural numbers, and variables) the  $\lambda$ -operator to form terms for substitution for function variables.*

**Proof.** We have full substitution by V. Also we have all constant and identity functions, using (besides II) S2, S3 and VI. The lemma follows as in IM, p. 221.

1.7. We now reconcile the present notion of primitive recursiveness with the notions in the literature. For functions  $\phi(\mathfrak{a})$  of number variables  $\mathfrak{a}$  only, cf. e.g. IM, p. 220 (and for  $\mathfrak{a}$  empty, Remark 1, p. 223). In e.g. [14, p. 313], we further called a function  $\phi(\mathfrak{a})$  with  $\mathfrak{a}$  of types 0, 1 "primitive recursive," if as a function of its type-0 variables it is primitive recursive uniformly in its type-1 variables (IM, p. 234 and Remark 1, p. 238).

VIII. A function  $\phi(a)$  of variables  $a$  of types  $<2$  is primitive recursive in the present sense, if and only if it is primitive recursive in the former sense.

**Proof.** For "if," use VII. For "only if," use I and IM, pp. 220-221.

1.8. Primitive recursiveness can be relativized to assumed functions  $\psi_1, \dots, \psi_l$  (briefly,  $\Psi$ ) by using *primitive recursive derivations from  $\Psi$*  (analogous to IM, p. 224) instead of primitive recursive descriptions. In a primitive recursive derivation from  $\Psi$ , besides using S1-S8, we may introduce any one  $\psi_i(c)$  of  $\Psi$  by the schema for a full substitution (cf. 1.6), for only its function variables, of functions of variables  $b$  distinct from its number variables; e.g., if  $\psi_i$  is  $\psi_i(c, \gamma_2^1, \gamma_3^2)$ :

$$S0.i \quad \phi(c, b) = \psi_i(c, \lambda t \chi_2(t, b), \lambda \sigma \chi_3(\sigma, b)) \quad \langle 0, \langle n_0, \dots, n_r \rangle, i, h_2, h_3 \rangle.$$

When  $c$  is number variables only, S0.i is  $\phi(c, b) = \psi_i(c)$ . (Alternatively, if S8.j ( $j \geq 2$ ) is replaced by S4.j ( $j \geq 1$ ), S7.j ( $j \geq 2$ ) of Remark 1, each  $\psi_i$  can be introduced simply by  $\phi(c, b) = \psi_i(c)$ .)

Now relativized versions I\*-VIII\* of I-VIII can be proved for functions primitive recursive on  $\Psi$ . As I\*, in an irredundant primitive recursive derivation of a function  $\phi(a)$  with  $a$  of types  $\leq r$  from functions  $\Psi$  of variables of types  $\leq s$  (introduced by S0.i ( $1 \leq i \leq l$ )), each function has all the variables in  $a$  of types  $> 0$  and only variables of types  $\leq \max(r, s-1)$ . As VIII\*, when  $\Psi$  are number-theoretic functions, and  $\phi$  a function of variables of types  $\leq 1$ , the present notion of relative primitive recursiveness coincides with the former one (IM, p. 224, [14, p. 313]).

1.9. When  $\Psi$  vary and  $\Theta$  are fixed,  $\phi$  is *primitive recursive in  $\Psi, \Theta$ , uniformly in  $\Psi$* , if there is a primitive recursive derivation of  $\phi$  from  $\Psi, \Theta$  with a fixed analysis (analogous to IM, p. 234). Then when  $\Psi$  are one-place functions,  $\phi(a)$  is primitive recursive in  $\Psi, \Theta$  uniformly in  $\Psi$ , exactly if  $\phi$  regarded as a function  $\phi(a, \Psi)$  is primitive recursive in  $\Theta$ . For on reconstructing  $\Psi$  as variables for the function  $\phi$ , the applications of S0.i to introduce  $\psi_i$  become applications of S7.1 or S8.j (with applications of S6); and inversely. It follows that, for fixed one-place functions  $\Psi_0$ , we can consider any function  $\phi(a)$  primitive recursive in  $\Psi_0, \Theta$  as the value for the fixed  $\Psi_0$  of a function  $\phi(a, \Psi)$  primitive recursive in  $\Theta$ .

1.10. As usual, the notions of primitive recursiveness, absolute and relative, extend via representing functions (IM, p. 227) to cases when some or all of  $\phi, \Psi, \Theta$  are replaced by predicates.

2. **Alterations of quantifiers.** 2.1. Let  $\langle a_0, \dots, a_n \rangle = p_0^{a_0} \cdot \dots \cdot p_n^{a_n}$  ( $= 1$  in the case  $n = -1$ , i.e.  $\langle \rangle = 1$ ). For each  $j \geq 1$  (<sup>(7)</sup>), let  $\langle \alpha_0^j, \dots, \alpha_n^j \rangle = \lambda \tau^{j-1} \langle \alpha_0^j(\tau^{j-1}), \dots, \alpha_n^j(\tau^{j-1}) \rangle$  and  $(\alpha^j)_i = \lambda \tau^{j-1} (\alpha^j(\tau^{j-1}))_i$ . Then, for each  $j$  ( $j \geq 0$ ),

(<sup>(7)</sup>) This makes  $\langle \alpha_0^j, \dots, \alpha_n^j \rangle = \lambda \tau^{j-1} 1$  for  $n = -1$  &  $j > 0$  ( $= 1$  for  $n = -1$  &  $j = 0$ ). We can write " $\langle \alpha_0^j, \dots, \alpha_n^j \rangle$ " (with " $j$ " shown) without ambiguity even when  $-1$  is a possible value of  $n$ . But " $\langle \rangle$ " shall mean 1, unless we show the type  $j$  as by " $\alpha^j = \langle \rangle$ " or " $\langle \rangle^j$ ".

$$(2) \quad (\langle \alpha_0^j, \dots, \alpha_n^j \rangle)_i = \alpha_i^j \quad (0 \leq i \leq n).$$

Now we have, generalizing IM, p. 285 (17) and [14, p. 315, (1) and (5)] to any  $j \geq 0$ ,

$$(3) \quad (E\alpha_0^j) \cdots (E\alpha_n^j) A(\alpha_0^j, \dots, \alpha_n^j) \equiv (E\alpha^j) A(\langle \alpha^j \rangle_0, \dots, \langle \alpha^j \rangle_n),$$

$$(4) \quad (\alpha^j)(E\beta^{j+1}) A(\alpha^j, \beta^{j+1}) \equiv (E\beta^{j+1})(\alpha^j) A(\alpha^j, \lambda\sigma^j \beta^{j+1}(\langle \alpha^j, \sigma^j \rangle)),$$

and their duals ( $\bar{3}$ ), ( $\bar{4}$ ).

**Proofs.** (3) Given  $\alpha_0^j, \dots, \alpha_n^j$  for the left,  $\alpha^j = \langle \alpha_0^j, \dots, \alpha_n^j \rangle$  is an  $\alpha^j$  for the right, by (2). The converse implication is immediate. (4)  $(\alpha^j)(E\beta^{j+1}) A(\alpha^j, \beta^{j+1}) \equiv (E\beta)(\alpha^j) A(\alpha^j, \lambda\sigma^j \beta(\alpha^j, \sigma^j)) \equiv (E\beta^{j+1})(\alpha^j) A(\alpha^j, \lambda\sigma^j \beta^{j+1}(\langle \alpha^j, \sigma^j \rangle))$ , since by (2), given any two-place  $\beta$  (not an object of one of our types 1.2) for the middle expression,  $\beta^{j+1} = \lambda\tau^j \beta((\tau^j)_0, (\tau^j)_1)$  is one for the right.

**2.2.** Using induction on  $j$ , we define two primitive recursive functions  $mp_j$  and  $pm_j$ , for each  $j \geq 1$ , as follows:

$$\begin{aligned} mp_1(\alpha^1, \beta^1) &= \alpha^1(\beta^1(0)), \\ pm_1(\alpha^2, \beta^0) &= \alpha^2(\lambda x \beta^0), \\ mp_{j+1}(\alpha^{j+1}, \beta^{j+1}) &= \alpha^{j+1}(\lambda\sigma^{j-1} pm_j(\beta^{j+1}, \sigma^{j-1})), \\ pm_{j+1}(\alpha^{j+2}, \beta^j) &= \alpha^{j+2}(\lambda\sigma^j mp_j(\beta^j, \sigma^j)). \end{aligned}$$

For  $j \geq 1$ , let  $mp_j(\alpha^j) = \lambda\beta^j mp_j(\alpha^j, \beta^j)$  and  $pm_j(\alpha^{j+1}) = \lambda\beta^{j-1} pm_j(\alpha^{j+1}, \beta^{j-1})$ ; and let  $mp_0(\alpha^0) = \lambda x \alpha^0$  and  $pm_0(\alpha^1) = \alpha^1(0)$ . The objects  $\alpha^j$  of type  $j$  are mapped into objects  $\alpha^{j+1} = mp_j(\alpha^j)$  of type  $j+1$ , and  $\alpha^j = pm_j(\alpha^{j+1})$  is the inverse mapping, as the following proposition states.

**IX.** For each  $j \geq 0$ ,

$$(5) \quad pm_j(mp_j(\alpha^j)) = \alpha^j.$$

The proof, using for  $j \geq 1$  induction on  $j$ , is straightforward.

Now we have, generalizing [14, (3)] to any  $j \geq 0$ ,

$$(6) \quad (E\alpha^j) A(\alpha^j) \equiv (E\alpha^{j+1}) A(pm_j(\alpha^{j+1})),$$

and its dual ( $\bar{6}$ ). For given an  $\alpha^j$  for the left,  $mp_j(\alpha^j)$  is an  $\alpha^{j+1}$  for the right, by (5).

**2.3.** It will be useful to have formulas for raising the type  $j$  by any number  $m - j \geq 0$ . Using induction on  $m$  for  $m \geq j \geq 1$ , let

$$\begin{aligned} mp_j^j(\alpha^j, \beta^{j-1}) &= \alpha^j(\beta^{j-1}), \\ mp_j^{m+1}(\alpha^j, \beta^m) &= mp_m(\lambda\beta^{m-1} mp_j^m(\alpha^j, \beta^{m-1}), \beta^m), \\ pm_j^j(\alpha^j, \beta^{j-1}) &= \alpha^j(\beta^{j-1}), \\ pm_j^{m+1}(\alpha^{m+1}, \beta^{j-1}) &= pm_j^m(\lambda\beta^{m-1} pm_m^{m+1}(\alpha^{m+1}, \beta^{m-1}), \beta^{j-1}). \end{aligned}$$

Also let

$$\text{mp}_0^0(\alpha^0) = \alpha^0, \quad \text{mp}_0^m(\alpha^0, \beta^{m-1}) = \alpha^0 \quad (m \geq 1),$$

$$\text{pm}_0^0(\alpha^0) = \alpha^0, \quad \text{pm}_0^1(\alpha^1) = \alpha^1(0), \quad \text{pm}_0^m(\alpha^m) = \alpha^m(\lambda\beta^{m-2}0) \quad (m \geq 2).$$

Finally, for  $m \geq j \geq 1$  or  $m > j = 0$ , let  $\text{mp}_j^m(\alpha^j) = \lambda\beta^{m-1} \text{mp}_j^m(\alpha^j, \beta^{m-1})$ ; and for  $m \geq j \geq 1$ , let  $\text{pm}_j^m(\alpha^m) = \lambda\beta^{j-1} \text{pm}_j^m(\alpha^m, \beta^{j-1})$ .

X. For each  $m \geq j \geq 0$ ,

$$(7) \quad \text{pm}_j^m(\text{mp}_j^m(\alpha^j)) = \alpha^j,$$

$$(8) \quad (E\alpha^j)A(\alpha^j) \equiv (E\alpha^m)A(\text{pm}_j^m(\alpha^m)) \quad (m \geq j \geq 0).$$

2.4. When  $j_0 \leq j_1 \leq \dots \leq j_n \leq m$ , let  $\langle \alpha_0^{j_0}, \dots, \alpha_n^{j_n} \rangle^m = \langle \text{mp}_{j_0}^m(\alpha_0^{j_0}), \dots, \text{mp}_{j_n}^m(\alpha_n^{j_n}) \rangle$ ; and for  $j \leq m$ , let  $(\alpha^m)_i^j = \text{pm}_j^m((\alpha^m)_i)$ . Then using (2) and (7),

$$(9) \quad (\langle \alpha_0^{j_0}, \dots, \alpha_n^{j_n} \rangle^m)_i^{j_i} = \alpha_i^{j_i} \quad (0 \leq i \leq n).$$

Consistently with 2.1, we may write  $\langle \alpha_0^{j_0}, \dots, \alpha_n^{j_n} \rangle^m$  as simply  $\langle \alpha_0^{j_0}, \dots, \alpha_n^{j_n} \rangle$  when  $m = j_n$ , and  $(\alpha^m)_i^m$  as  $(\alpha^m)_i$ . Finally, by (a) let us mean  $\langle \alpha_0^{j_0}, \dots, \alpha_n^{j_n} \rangle$  where  $\alpha_0^{j_0}, \dots, \alpha_n^{j_n}$  is the result of arranging  $\alpha$  in order of nondecreasing type preserving the given order within each type<sup>(7)</sup>.

2.5. A consecutive set of  $n+1$  quantifiers of like kind and assorted types  $\leq m$  can be contracted into one type- $m$  quantifier of the same kind, by using first (8) and then (3) (or  $(\bar{8})$ ,  $(\bar{3})$ ). The operation can be performed in one step using (9). For example, when  $p \leq m$ ,

$$(a)(\alpha_1^p)(\alpha_2^m)A(a, \alpha_1^p, \alpha_2^m) \equiv (\alpha^m)A((\alpha^m)_0^0, (\alpha^m)_1^p, (\alpha^m)_2^m).$$

The prefix  $(\alpha^j)(E\beta^k)$  for any  $j \leq m$  and  $k \leq m+1$  can be changed to  $(E\beta^{m+1})(\alpha^m)$  (or dually) by use of (8),  $(\bar{8})$  and then (4) (or  $(\bar{4})$ ). It is more expeditious, except when (4) or  $(\bar{4})$  is immediately applicable, to proceed in the manner used in proving (4), using (9) to effect the contraction required when  $k > 0$ . For example,

$$\begin{aligned} (\alpha^2)(Ex)A(\alpha^2, x) &\equiv (E\alpha^3)(\alpha^2)A(\alpha^2, \alpha^3(\alpha^2)), \\ (\alpha^4)(E\beta^3)A(\alpha^4, \beta^3) &\equiv (E\beta)(\alpha^4)A(\alpha^4, \lambda\sigma^2\beta(\alpha^4, \sigma^2)) \\ &\equiv (E\beta^5)(\alpha^4)A(\alpha^4, \lambda\sigma^2\beta^5(\langle \alpha^4, \sigma^2 \rangle)). \end{aligned}$$

The same technique can be used in advancing and contracting several quantifiers simultaneously (cf. [14, Footnote 10]); e.g.

$$\begin{aligned} (x)(E\alpha^3)(\beta^3)(E\gamma^1)A(x, \alpha^3, \beta^3, \gamma^1) &\equiv \\ (E\alpha)(E\gamma)(x)(\beta^3)A(x, \lambda\sigma^2\alpha(x, \sigma^2), \beta^3, \lambda t\gamma(x, \beta^3, t)) &\equiv \\ (E\alpha^3)(E\gamma^4)(x)(\beta^3)A(x, \lambda\sigma^2\alpha^3(\langle x, \sigma^2 \rangle), \beta^3, \lambda t\gamma^4(\langle x, \beta^3, t \rangle)) &\equiv \\ (E\alpha^4)(\beta^3)A((\beta^3)_0^0, \lambda\sigma^2\{(\alpha^4)_0^3\}(\langle (\beta^3)_0^0, \sigma^2 \rangle), (\beta^3)_1, \lambda t\{(\alpha^4)_1\}(\langle (\beta^3)_0^0, (\beta^3)_1, t \rangle)). \end{aligned}$$

2.6. These alterations of quantifiers are effected by substitutions of primitive recursive functions in the scope. As another application of the contraction formulas:

XI. *Given any function  $\psi(a)$  of variables  $a$  of types  $\leq m$ , there is a function  $\psi'(\alpha^m)$  such that each of  $\psi(a)$  and  $\psi'(\alpha^m)$  comes from the other by substitution of primitive recursive functions.*

**Proof.** For example, given  $\psi(a, \alpha_1^p, \alpha_2^m)$  ( $p \leq m$ ), take  $\psi'(\alpha^m) = \psi((\alpha^m)_0^0, (\alpha^m)_1^p, (\alpha^m)_2^m)$ . Then  $\psi(a, \alpha_1^p, \alpha_2^m) = \psi'(\langle a, \alpha_1^p, \alpha_2^m \rangle)$ .

2.7. We can use XI in connection with 1.9 when  $\Psi$  include more-than-one-place functions. For example (with  $p \leq m$ ), if  $\phi(a)$  is primitive recursive in  $\psi(a, \alpha_1^p, \alpha_2^m)$ ,  $\Theta$ , then using  $\psi(a, \alpha_1^p, \alpha_2^m) = \psi'(\langle a, \alpha_1^p, \alpha_2^m \rangle)$  we have  $\phi(a)$  primitive recursive in  $\psi'$ ,  $\Theta$  for  $\psi' = \lambda\alpha^m \psi((\alpha^m)_0^0, (\alpha^m)_1^p, (\alpha^m)_2^m)$ . Thence by 1.9 we obtain a function  $\phi(a, \alpha^{m+1})$  primitive recursive in  $\Theta$  such that  $\phi(a) = \phi(a, \psi') = \phi(a, \lambda\alpha^m \psi((\alpha^m)_0^0, (\alpha^m)_1^p, (\alpha^m)_2^m))$ .

3. **Partial and general recursive functions.** 3.1. By Church's thesis for number-theoretic functions [4] and our extension of it to functions with variables of type 1 as well [12] and their converses, the general recursive functions of variables of types 0, 1 comprise exactly the functions of such variables which are "effectively calculable" (cf. IM, §62 and end §61). The effective calculability is relative when there are type-1 variables. Thus, for a function  $\phi(\alpha)$  of a type-1 variable  $\alpha$ , we imagine after Turing [30, pp. 172–173, 200] an "oracle" for  $\alpha$  who at any step in the computation can be called upon to supply the value of  $\alpha(y)$  for a computed  $y$  <sup>(8)</sup>. The use of the word "oracle" emphasizes that in the steps from a  $y$  to the value of  $\alpha(y)$  the otherwise mechanical character of the computation is transcended. The computation is carried out by a preassigned procedure, using steps all of them mechanical except those from  $y$  to  $\alpha(y)$ , which we "assume" provided upon demand.

For a function  $\phi(\mathbf{F})$  of a type-2 variable  $\mathbf{F}$ , it seems natural to image similarly an "oracle" for  $\mathbf{F}$  who at any step in the computation can be called upon to supply the value of  $\mathbf{F}(\beta)$  when a procedure for computing  $\beta(y)$  for any  $y$  has arisen. Quite as before, we mean by this that  $\phi(\mathbf{F})$  can be computed by a preassigned procedure, using steps all of them mechanical except for steps from a  $\beta$  to the value of  $\mathbf{F}(\beta)$ . Such steps we "assume" provided upon demand, under circumstances in which we could in the same sense, i.e. with the help of the oracle for  $\mathbf{F}$ , compute  $\beta(y)$  for any  $y$  ( $y = 0, 1, 2, \dots$ ). When the oracle for  $\mathbf{F}$  is used, if the computations of  $\beta(y)$  for each  $y = 0, 1, 2, \dots$  are considered to be parts of the computation of  $\phi(\mathbf{F})$ , that computation for the given  $\mathbf{F}$  will not be a finite object, unlike the computation of  $\phi(\alpha)$  for a given  $\alpha$ . It cannot be otherwise when  $\mathbf{F}(\beta)$  depends on all the values of  $\beta$ .

<sup>(8)</sup> Turing had in mind only particular  $\alpha$ 's (primarily, in our terms, the representing function of  $(x)\bar{T}_1(y, y, x)$ ), while here (as in [12] and IM) we are adding the idea of uniformity in  $\alpha$  (the procedure for computing not varying with  $\alpha$ , but only the answers the oracle gives) to obtain the effective calculability of a function of  $\alpha$ .

So, it seems, this is the direction the generalization of effective calculability to type 2 must take, if we are to allow arbitrary type-2 objects  $\mathbf{F}$  as arguments. The resulting notion, though it employs a very potent oracle, is still an immense restriction on the notion of an arbitrary function  $\phi(\mathbf{F})$ , and should be of mathematical interest. That the computation of  $\phi(\mathbf{F})$  is not in general a finite object is already illustrated by the primitive recursive functions  $\phi(\mathbf{F})$ . Our objective now is to characterize what, we believe, is the class of all functions  $\phi(\mathbf{F})$  which are effectively calculable (and more generally, of all such functions  $\phi(\mathfrak{a})$  for  $\mathfrak{a}$  of assorted types).

Besides developing the theory for arbitrary  $\mathbf{F}$ 's as we are proposing here, we can consider specializations and alternatives under restrictions on the  $\mathbf{F}$ 's. Our basic idea of the effective calculability of a function  $\phi$  admitting non-constructive objects as arguments is that the calculation should be performable by preassigned rules, constructively except for the use of the oracles necessary to put in those nonconstructive objects. Basically, each of those nonconstructive objects can be a function (so  $\phi$  is a functional), and the role of the oracle for it is to give its value when supplied its argument(s). Different notions of "effectively calculable" or "constructive" functional may be related on this basis <sup>(9)</sup>.

**3.2.** For functions of variables of types 0, 1, there are a number of ways of describing the class of the general recursive functions, or in other words there are a number of notions equivalent to general recursiveness as usually formulated (cf. IM, §62). In setting out to give an extension to higher types of variables that shall preserve Church's thesis, one should not assume a

---

<sup>(9)</sup> For example, in [18] we confine  $\mathbf{F}$  to functions (called *countable functionals* of type 2) each of whose values  $\mathbf{F}(\beta)$  is determined by finitely many values of  $\beta(y)$ . Then we employ an oracle to which we give a different role in relation to  $\mathbf{F}$  than above. We may ask this oracle a question of the form, "Does  $\mathbf{F}(\beta)$  have the same value for all  $\beta$ 's such that  $\beta(0) = b_0, \dots, \beta(x-1) = b_{x-1}$ , and if so what is that value?" (The question can be asked by supplying to the oracle a value of  $\beta(x)$  <sup>(\*)</sup>.) The oracle may reply either "No comment" or "Yes, and that value is —"; but, for a given  $\beta$ , there must be some  $x$  for which he gives the second kind of answer. In [18, §1] we note that this amounts to having an oracle in the basic sense for a certain kind of type-1 function  $\mathbf{F}^{(1)}$  (or  $\alpha^{(2)}$  if  $\mathbf{F} = \alpha^{(2)}$ ) called an *associate* of  $\mathbf{F}$ . Under the thesis that the effectively calculable functions on a given domain are those which are general recursive on that domain (indeed, the case of it for a type-1 variable already entertained in [12] and IM end §61), the functions  $\phi(\mathbf{F})$  "effectively calculable" with this new kind of oracle are those with  $\phi_1$  partial recursive, where  $\phi_1(\mathbf{F}^{(1)}) = \phi(\mathbf{F})$  for every associate  $\mathbf{F}^{(1)}$  of  $\mathbf{F}$ ; these  $\phi(\mathbf{F})$  we call *recursively countable* (cf. [18, Theorem 2]). All functions  $\phi(\mathbf{F})$  general recursive (as defined below) for  $\mathbf{F}$  countable are recursively countable [18, Theorem 4]. Whether the converse is true is an open problem. This does not mean that the class of the general recursive functions  $\phi(\mathbf{F})$  may be too narrow to represent effective calculability when  $\mathbf{F}$  is restricted to be countable (i.e. that our thesis may fail for such  $\mathbf{F}$ ). It only means (under the thesis) that there may be functions  $\phi(\mathbf{F})$  for which  $\phi_1(\mathbf{F}^{(1)})$  is effectively calculable, though  $\phi(\mathbf{F})$  itself is not, in the basic sense. That is, the new oracle may be more potent. (We failed in an attempt to determine whether there is an effective way to perform the job the new oracle does by using the basic oracle for  $\mathbf{F}$ .) There are purposes for which one may be more interested in using the new oracle (i.e. in having merely  $\phi_1$  rather than  $\phi$  effectively calculable in the basic sense).

priori that it is immaterial which of the existing notions for types 0, 1 he extends. We shall in fact start with a new formulation, which provides the essentials for Church's thesis quite directly, and later (in §8 and Part II) we shall examine extensions of the several existing notions.

**3.3.** One essential for Church's thesis to hold is that there be available a modicum of elementary operations of computation. This is provided by the schemata for primitive recursive functions.

**3.4.** The second essential is that there be the means for reflecting upon computation procedures already set up as objects and computing further computation procedures from them.

This second essential entails that there should be no means for deciding in general whether a computation procedure terminates. For otherwise by diagonalization we could effectively get outside our class of computation procedures, so Church's thesis could not apply.

A function  $\phi$ , taking a natural number as value, which is defined over some subset proper or improper of a set  $D$  we call a *partial function over  $D$* ; we then sometimes think of  $\phi$  as a function from  $D$  to  $\{0, 1, 2, \dots; u\}$  where  $u$  stands for "undefined" (cf. IM, pp. 325–326). In the following, we deal with partial functions  $\phi(\alpha)$  for  $\alpha$  a given list of variables of the types introduced in 1.2, so  $D$  is the set of all  $n$ -tuples of objects of the respective types of  $\alpha$ .

When  $\phi(\alpha)$  and  $\psi(\alpha)$  are partial functions, we write  $\phi(\alpha) \simeq \psi(\alpha)$  to express that both  $\phi(\alpha)$  and  $\psi(\alpha)$  are defined with the same value or both are undefined (IM, p. 327).

**3.5.** In order to allow computation procedures, or what comes to the same thing recursive (or "computable") functions, to be treated as objects in computing further computation procedures, we shall assign numbers called "indices" to the recursive functions. (The indices take over the role of the Gödel numbers in our previous theory of general and partial recursive functions; cf. IM, pp. 292, 330, 340 ff., and [17, §3].)

We begin with an assignment of indices to the primitive recursive functions. Consider such a function  $\phi(\alpha)$ . Let  $r$  be the maximum type of  $\alpha$  (1.4), and let  $n_0, \dots, n_r$  be the numbers of variables in  $\alpha$  of types 0,  $\dots, r$ , respectively (thus  $n_r > 0$ , except for  $\alpha$  empty). We assign to  $\phi(\alpha)$  an index determined by a given primitive recursive description of  $\phi(\alpha)$ . For  $\phi(\alpha)$  introduced by a given one of the schemata S1–S8, the *index* is the number shown at the right opposite the schema (cf. 1.3, 2.1), where (in the cases of S4, S5, S6 and S8)  $g$  and  $h$  are the *indices* already determined for the  $\psi$  and  $\chi$  by the descriptions of them as part of the given description of  $\phi$ .

Under this method of indexing, inversely an index of  $\phi(\alpha)$  *determines* an irredundant description of  $\phi(\alpha)$ , say that one in which at each application of S4 or S5 the entire descriptions of the  $\psi$  and the  $\chi$ , and nothing else, each ap-

pear separately and successively preceding the application of S4 or S5, and similarly at each application of S6 or S8.

3.6. Now for example we can define a function  $\phi(a)$  whose value for arguments  $a$  is  $\phi_a(a)$  where, for each  $b$ ,  $\phi_b(a)$  is the function with index  $\chi(b)$ . If  $\{z\}(a)$  is the function of  $a$  with index  $z$ , the function  $\phi(a)$  can be written  $\phi(a) = \{\chi(a)\}(a)$ . To introduce this  $\phi(a)$  it would suffice to use S4 with  $\chi(a)$  and the function  $\psi(a, b) = \{a\}(b)$  as given functions. This suggests the definition of "partial recursive function" we now adopt.

3.7. To obtain the partial recursive functions, we use the schemata S1–S8, now with  $\simeq$  instead of  $=$  and interpreting  $\alpha^i(\lambda\alpha^{i-2}\chi(\alpha^i, \alpha^{i-2}, b))$  for S8 to be undefined when  $\lambda\alpha^{i-2}\chi(\alpha^i, \alpha^{i-2}, b)$  is incompletely defined<sup>(10)</sup>, and one further schema

$$S9 \quad \phi(a, b, c) \simeq \{a\}(b) \quad \langle 9, \langle n_0, \dots, n_r \rangle, \langle m_0, \dots, m_s \rangle \rangle,$$

where  $s$  is the maximum type of  $b$ , and  $m_0, \dots, m_s$  are the numbers of variables in  $b$  of types  $0, \dots, s$ , respectively. A function  $\phi(a)$  is *partial recursive*, if there is a *partial recursive description* of it in terms of these schemata. Each partial recursive description of  $\phi(a)$  determines an *index* and inversely, in the same manner as before (3.5). Finally, for each  $z$  which is an index of a partial recursive function  $\phi(a)$  and each  $a$ ,  $\{z\}(a)$  is the value if defined of  $\phi(a)$ ; otherwise  $\{z\}(a)$  is undefined. (For number variables  $a$ ,  $\{z\}(a)$  is not the function so written in IM, p. 340, but the analog of it here.)

A partial recursive function  $\phi(a)$  which is defined for all  $a$  is *general recursive*. A predicate is *partial (general) recursive*, if its representing function is partial (general) recursive.

3.8. The definition of  $\{z\}(a)$  via an interaction of schemata and indexing can be elaborated as follows. First,  $\{z\}(a)$  is defined only when  $z$  is an index of a function  $\phi(a)$ . Second, the schemata S1–S9 can be written as equations to be satisfied by  $\{z\}(a)$  as a partial function of  $z$  and a varying list  $a$  of variables; e.g.:

$$S1 \quad \{1, \langle n_0, \dots, n_r \rangle\}(a, b) \simeq a'.$$

$$S5a \quad \{5, \langle n_0, \dots, n_r \rangle, g, h\}(0, b) \simeq \{g\}(b).$$

$$S5b \quad \{5, \langle n_0, \dots, n_r \rangle, g, h\}(a', b) \simeq \{h\}(a, \{5, \langle n_0, \dots, n_r \rangle, g, h\}(a, b), b).$$

$$S8 \quad \{8, \langle n_0, \dots, n_r \rangle, j, h\}(\alpha^i, b) \simeq \alpha^i(\lambda\alpha^{i-2}\{h\}(\alpha^i, \alpha^{i-2}, b)).$$

$$S9 \quad \{9, \langle n_0, \dots, n_r \rangle, \langle m_0, \dots, m_s \rangle\}(a, b, c) \simeq \{a\}(b).$$

Third, these equations can be construed as the direct clauses of a transfinite inductive definition (cf. [11, §7], IM §53) of the predicate  $\{z\}(a) \simeq w$  (each applicable however only for  $z$  an index of a function of  $a$ ). Thus S1 is a basic clause, and S5a, S5b, S8, S9 can be read as inductive clauses; e.g.:

<sup>(10)</sup> Likewise  $\psi(\chi(b), b)$  for S4 ( $\chi(a, \phi(a, b), b)$  for S5) shall be undefined when  $\chi(b)$  ( $\phi(a, b)$ ) is undefined, i.e. it shall have the "weak sense" IM, p. 327.

- S5b If  $\{ \langle 5, \langle n_0, \dots, n_r \rangle, g, h \rangle \} (a, \mathbf{b}) \simeq u$  and  $\{ h \} (a, u, \mathbf{b}) \simeq w$ ,  
 then  $\{ \langle 5, \langle n_0, \dots, n_r \rangle, g, h \rangle \} (a', \mathbf{b}) \simeq w$ .
- S8 If for each  $\alpha^{i-2}$ ,  $\{ h \} (\alpha^i, \alpha^{i-2}, \mathbf{b}) \simeq \gamma^{i-1}(\alpha^{i-2})$ ,  
 then  $\{ \langle 8, \langle n_0, \dots, n_r \rangle, j, h \rangle \} (\alpha^i, \mathbf{b}) \simeq \alpha^i(\gamma^{i-1})$ .
- S9 If  $\{ a \} (\mathbf{b}) \simeq w$ , then  $\{ \langle 9, \langle n_0, \dots, n_r \rangle, \langle m_0, \dots, m_s \rangle \} (a, \mathbf{b}, c) \simeq w$ .

As the extremal clause of the inductive definition,  $\{ z \} (a) \simeq w$  is to hold only as required by the direct clauses.

Now we can prove by mathematical induction, in the form corresponding to this inductive definition, that, for given values of  $z$  and  $a$ , the proposition  $\{ z \} (a) \simeq w$  holds for at most one natural number  $w$ . The proof is afforded by the fact that, for given  $z$  and  $a$ , at most one of the equations has  $\{ z \} (a)$  as left member.

Thereby  $\{ z \} (a)$  is defined as the partial function whose value, for given  $z$  and  $a$ , is the unique  $w$  (if any) such that  $\{ z \} (a) \simeq w$  holds by the inductive definition.

Mathematical induction in the aforesaid form can be used to establish properties of  $\{ z \} (a)$  when defined. We shall refer to such an induction as *induction on  $\{ z \} (-)$* .

**3.9.** We consider an example to illustrate how the inductive definition of  $\{ z \} (a) \simeq w$  provides a computation process for  $\{ z \} (a)$  (but one which in general makes the computation an infinite object, cf. 3.1).

Given  $z, \mathbf{F}$ , where  $z$  is an index of a function of  $\mathbf{F}$ , how do we compute  $\{ z \} (\mathbf{F})$ ? Say e.g.  $z = \langle 4, \langle 0, 0, 1 \rangle, g, h \rangle$ . Then the value of  $\{ z \} (\mathbf{F})$  is that of  $\{ g \} (\{ h \} (\mathbf{F}), \mathbf{F})$ , if the latter is defined. Consider first  $\{ h \} (\mathbf{F})$ . Say e.g.  $h = \langle 8, \langle 0, 0, 1 \rangle, 2, k \rangle$ . Then the value of  $\{ h \} (\mathbf{F})$  is that of  $\mathbf{F}(\beta)$  as supplied by the "oracle" for  $\mathbf{F}$ , provided that, for each  $y$ ,  $\{ k \} (\mathbf{F}, y)$  is defined with value  $\beta(y)$ . That is, if, for each  $y$ , by proceeding similarly to compute  $\{ k \} (\mathbf{F}, y)$  a value would be obtained, call it  $\beta(y)$ , a value of  $\{ h \} (\mathbf{F})$  results, call it  $u$  ( $u = \mathbf{F}(\beta)$ ). Say  $g = \langle 9, \langle 1, 0, 1 \rangle, \langle 0, 0, 1 \rangle \rangle$ . Then the value of  $\{ g \} (u, \mathbf{F})$  is that of  $\{ u \} (\mathbf{F})$  if defined. Say  $u = \langle 2, \langle 0, 0, 1 \rangle, 5 \rangle$ . Then the value of  $\{ u \} (\mathbf{F})$  is 5; this is the value of  $\{ g \} (u, \mathbf{F})$ , of  $\{ g \} (\{ h \} (\mathbf{F}), \mathbf{F})$ , and of  $\{ z \} (\mathbf{F})$ , which we sought.

**3.10.** By our definition, the class of the partial recursive functions is closed under applications of the primitive recursive schemata S1–S8 <sup>(11)</sup>. In particular, primitive recursive functions are partial recursive; and the class of partial recursive functions is closed under primitive recursion, and under

<sup>(11)</sup> In the presence of S9, primitive recursion S5 can clearly be replaced by some selection of initial functions, as was the case for types  $\leq 1$  in the presence of the author's  $\mu$ -schema (cf. [9] and Grzegorzczuk [6]). However it adds little to the work below to keep S5, and we prefer doing so to being drawn here into the investigation of what initial functions constitute a good choice for this purpose. Besides, some of the theory for partial recursive functions includes conveniently theory for primitive recursive functions by S5 being kept.

substitution so long at least as substitution for a function variable takes place only directly for the argument of a function variable (for while we have S8, we do not immediately have IV). Clearly I, II, III, VI extend to partial recursive functions.

As to IV, the former proof gives it to us for the case  $\phi(a, \sigma^n, b)$  is primitive recursive,  $\theta(a, b, \tau^{n-1})$  is general recursive, and  $n = 1$  (so that Case 8 Subcase 1 of the proof will not arise). The extension of IV more generally (and thence of V, VII) will be dealt with in §4 and in Part II.

The extension of VIII will be given by XVII and XXXI below.

**3.11. XII.**  $\{z\}(a)$  is a partial recursive function of  $z, a$  such that  $\{0\}(a), \{1\}(a), \{2\}(a), \dots$  is an enumeration with repetitions of the partial recursive functions of  $a$ . (Enumeration theorem for partial recursive functions.)

**Proof.**  $\{z\}(a)$  is partial recursive, because it is introduced by S9 with  $c$  empty. Thence, for each fixed  $z$ , using S4 to substitute for  $z$  the constant function  $\chi(a) = z$  given by S2 with  $z$  as the  $q$ ,  $\{z\}(a)$  is a partial recursive function of  $a$ . Any partial recursive function  $\phi(a)$  has an index  $z$  by 3.7, and  $\phi(a) \simeq \{z\}(a)$ .

**REMARK 2.** XII is the analog of IM, Theorem XXII, p. 341, but our use of indices gives it to us at the very beginning of the theory. By XII, we get all partial recursive functions using only S9 with  $c$  empty and substitution of constants  $z$ , but of course S1–S8 enter into our definition of the function  $\{a\}(b)$  introduced by S9.

**3.12. XIII.** For each  $m \geq 1$ : There is a primitive recursive function  $S^m(z, y_1, \dots, y_m)$  such that, if  $\phi(y_1, \dots, y_m, b)$  is a partial recursive function of the variables  $y_1, \dots, y_m, b$  with index  $z$ , then for each fixed  $y_1, \dots, y_m$ ,  $S^m(z, y_1, \dots, y_m)$  is an index of  $\phi(y_1, \dots, y_m, b)$  as a function of  $b$  (analogously to IM, Theorem XXIII, p. 342).

**Proof,** by induction on  $m$ . **BASIS** ( $m = 1$ ). We have  $\phi_y(b) \simeq \phi(y, b) \simeq \phi(\chi(b), b)$  by S4 where  $\chi(b) = y$  by S2. So take  $S^1(z, y) = \langle 4, [(z)_1/2], z, \langle 2, [(z)_1/2], y \rangle \rangle$ , noting that, if  $(z)_1 = \langle n_0, n_1, \dots, n_r \rangle$  with  $n_0 > 0$ ,  $[(z)_1/2] = \langle n_0 - 1, n_1, \dots, n_r \rangle$ . **IND. STEP.** Take  $S^{m+1}(z, y_1, \dots, y_m, y_{m+1}) = S^1(S^m(z, y_1, \dots, y_m), y_{m+1})$ .

**XIV.** Given any partial recursive function  $\psi(z, b)$ , an index  $e$  of  $\psi(e, b)$  can be found; thus we can solve for  $z$  the equation

$$\{z\}(b) \simeq \psi(z, b).$$

(The recursion theorem, analogous to IM, Theorem XXVII, pp. 352–353.)

**Proof.** By the same method as before; i.e. let  $f$  be an index of  $\psi(S^1(y, y), b)$  as a function of  $y, b$ , and take  $e = S^1(f, f)$ .

**3.13. XV.** If  $\psi_0(a), \psi_1(a)$  are partial recursive functions, and  $Q(a)$  is a partial recursive predicate, the function

$$\phi(a) \simeq \begin{cases} \psi_0(a) & \text{if } Q(a), \\ \psi_1(a) & \text{if } \bar{Q}(a) \end{cases}$$

(undefined exactly if  $Q(a)$  is undefined, or  $Q(a)$  is true and  $\psi_0(a)$  is undefined, or  $Q(a)$  is false and  $\psi_1(a)$  is undefined) is partial recursive.

**Proof.** Let  $\chi(a)$  be the representing function of  $Q(a)$ , and let  $e_i$  be an index of  $\psi_i(a)$  ( $i = 0, 1$ ). Define a primitive recursive function  $\pi$  by

$$\pi(i) = \begin{cases} e_0 & \text{if } i = 0, \\ e_1 & \text{if } i \neq 0. \end{cases}$$

Then  $\phi(a) \simeq \{\pi(\chi(a))\}(a)$ .

XVI. If  $\chi(b, y)$  is partial recursive, so is  $\mu y[\chi(b, y) = 0]$ . (Cf. IM, pp. 279, 329–330.)

**Proof.** We have  $\mu y[\chi(b, y) = 0] \simeq \phi(b, 0)$  where

$$\phi(b, y) \simeq \begin{cases} 0 & \text{if } \chi(b, y) = 0, \\ \phi(b, y')' & \text{if } \chi(b, y) \neq 0. \end{cases}$$

Note that this does make  $\phi(b, 0)$  undefined not only when, as  $y$  increases, a  $y$  for which  $\chi(b, y)$  is undefined occurs before one for which  $\chi(b, y) = 0$ , but also (as then  $\phi(b, 0) \simeq 1 + \phi(b, 1) \simeq 2 + \phi(b, 2) \simeq \dots$ ) when all  $\chi(b, y)$  are false. To get  $\phi(b, y)$  as a partial recursive function, let

$$\psi(z, b, y) \simeq \begin{cases} 0 & \text{if } \chi(b, y) = 0, \\ \{z\}(b, y)' & \text{if } \chi(b, y) \neq 0 \end{cases}$$

(using XV), and set  $\phi(b, y) \simeq \psi(e, b, y)$  where  $e$  is index of  $\psi(e, b, y)$  given by XIV.

XVII. Each function  $\phi(a)$  of variables  $a$  of type  $\leq 1$  partial (general) recursive in the former sense (e.g. IM, Chapters XI, XII) is partial (general) recursive in the present sense.

**Proof.** By the former normal form theorem (IM, Theorem XIX, p. 330) with XVI and VIII. (We are not yet considering here any but completely defined functions as values of our type-1 variables, so IM, Theorem XIX is available to us. The situation when incompletely defined functions are allowed as type-1 arguments will be considered in Part II.)

XVIII. If  $\psi_0(a), \dots, \psi_m(a), Q_0(a), \dots, Q_m(a)$  are partial recursive, so is the function

$$\phi(a) \simeq \begin{cases} \psi_0(a) & \text{if } Q_0(a), \\ \text{otherwise } \psi_1(a) & \text{if } Q_1(a), \\ \dots & \dots \\ \text{otherwise } \psi_m(a) & \text{if } Q_m(a) \end{cases}$$

(undefined exactly if there is no  $i$  ( $0 \leq i \leq m$ ) such that  $Q_0(a), \dots, Q_{i-1}(a)$  are false,  $Q_i(a)$  is true and  $\psi_i(a)$  is defined). (Definition by cases.)

**Proof.** Let  $e_0, \dots, e_m(q_0, \dots, q_m)$  be indices of  $\psi_0(a), \dots, \psi_m(a)$  (of the representing functions of  $Q_0(a), \dots, Q_m(a)$ ), and let

$$\rho(i) = \begin{cases} e_0 & \text{if } i=0, \\ \dots & \dots \\ e_{m-1} & \text{if } i=m-1. \\ e_m & \text{if } i \geq m, \end{cases}$$

$$\sigma(i) = \begin{cases} q_0 & \text{if } i=0, \\ \dots & \dots \\ q_{m-1} & \text{if } i=m-1, \\ q_m & \text{if } i \geq m. \end{cases}$$

Then using XVI, let

$$\phi(a) \simeq \{ \rho(\mu i [\{ \sigma(i) \} (a) = 0]) \} (a).$$

We shall compare XVIII with the previous result for types 0, 1 (IM, Theorem XX(c), p. 337) in Part II.

**3.14.** Relative partial and general recursiveness are obtained by allowing assumed functions  $\psi_1, \dots, \psi_l$  (briefly,  $\Psi$ ), introduced by applications of S0.i (1.8, excluding the alternative there). Indices become *indices from*  $\Psi$ ;  $\{z\}(a)$  becomes  $\{z\}^\Psi(a)$ . For uniformity, cf. 1.9; for predicates, cf. 1.10.

In an application of S0.i it may now happen that some  $\lambda\alpha^{i-2} \chi_k(\alpha^{i-2}, b)$  is an incompletely defined function and thus not an object from the range of  $\gamma_k^{j-1}$  as established in 1.2. Here we shall take the result of the substitution into  $\psi_i$  to be undefined in this case, just as in 3.7 for S8.j we took  $\alpha^i(\lambda\alpha^{i-2} \chi(\alpha^i, \alpha^{i-2}, b))$  to be undefined when  $\lambda\alpha^{i-2} \chi(\alpha^i, \alpha^{i-2}, b)$  is incompletely defined.

On the other hand, there is no reason here for not allowing  $\psi_1, \dots, \psi_l$  to be partial functions.

**3.15.** When  $\psi_1, \dots, \psi_l$  are completely defined functions, the transformations of 1.9 (extended to include S9) and 2.7 provide an alternative (for most purposes) to developing the theory in relativized form.

**4. Construction of indices. 4.1.** We now write  $Ix(z)$  ( $PRI(z)$ ), and say  $z$  is an *index* (*primitive recursive index*), when  $z$  is an index of a partial (primitive) recursive function  $\phi(a)$  as assigned in 3.7 (in 3.5)<sup>(12)</sup>. In this case we also say  $z$  is an *index for a*. Then if  $a$  is a list of variables with  $r$  and  $n_0, \dots, n_r$  as in 3.5,  $\langle n_0, \dots, n_r \rangle = (z)_1$  and  $r = tp(z)$  where  $tp$  is the primitive recursive function defined thus:

$$tp(z) = \overline{\mu i_{i < z}(j)_{i < j < z} p_j} | (z)_1.$$

**XIX.** *The predicates  $Ix(z)$  and  $PRI(z)$  are primitive recursive.*

**Proof** for  $Ix(z)$ .  $Ix(z)$  satisfies the course-of-values recursion

---

<sup>(12)</sup> It is a trifling departure from the analogy to Gödel numbers (IM, pp. 292, 330, 340) that we do not call any  $z$  such that  $(a)[\phi(a) \simeq \{z\}(a)]$  an "index" of  $\phi(a)$ . By 3.7, such a  $z$  can fail to be an index only when  $\phi(a)$  is the completely undefined function of  $a$ .

$$\begin{aligned}
 \text{Ix}(z) \equiv & \left[ ((z)_0 = 1 \vee (z)_0 = 3) \ \& \ z = \prod_{i < 2} p_i^{(z)_i} \ \& \ (z)_{1,0} > 0 \right] \\
 & \vee \left[ (z)_0 = 2 \ \& \ z = \prod_{i < 3} p_i^{(z)_i} \ \& \ (z)_1 > 0 \right] \\
 & \vee \left[ (z)_0 = 4 \ \& \ z = \prod_{i < 4} p_i^{(z)_i} \ \& \ \text{Ix}((z)_2) \ \& \ (z)_{2,1} = 2 \cdot (z)_1 \ \& \ \text{Ix}((z)_3) \ \& \ (z)_{3,1} = (z)_1 \right] \\
 & \vee \left[ (z)_0 = 5 \ \& \ z = \prod_{i < 4} p_i^{(z)_i} \ \& \ \text{Ix}((z)_2) \ \& \ 2 \cdot (z)_{2,1} = (z)_1 \ \& \ \text{Ix}((z)_3) \ \& \ (z)_{3,1} = 2 \cdot (z)_1 \right] \\
 & \vee \left[ (z)_0 = 6 \ \& \ z = \prod_{i < 5} p_i^{(z)_i} \ \& \ (z)_3 < (z)_{1,(z)_2} \ \& \ \text{Ix}((z)_4) \ \& \ (z)_{4,1} = (z)_1 \right] \\
 & \vee \left[ (z)_0 = 7 \ \& \ z = \prod_{i < 2} p_i^{(z)_i} \ \& \ (z)_{1,0} > 0 \ \& \ (z)_{1,1} > 0 \right] \\
 & \vee \left[ (z)_0 = 8 \ \& \ z = \prod_{i < 4} p_i^{(z)_i} \ \& \ (z)_2 > 1 \ \& \ (z)_{1,(z)_2} > 0 \ \& \ \text{Ix}((z)_3) \right. \\
 & \qquad \qquad \qquad \left. \ \& \ (z)_{3,1} = p_{(z)_2 \cdot 2} \cdot (z)_1 \right] \\
 & \vee \left[ (z)_0 = 9 \ \& \ z = \prod_{i < 3} p_i^{(z)_i} \ \& \ (z)_{1,0} > (z)_{2,0} \ \& \ (i)_{0 < i \leq z}((z)_{1,i} \geq (z)_{2,i}) \right].
 \end{aligned}$$

4.2. XX. *There is a primitive recursive function  $\pi(j, k, g)$  such that, for  $\alpha, \alpha_1$  as in Schema S6, if  $g$  is an index of  $\psi(\alpha)$ , then  $\pi(j, k, g)$  is one of  $\phi(\alpha_1)$ .*

**Proof.** Let  $\pi(j, k, g, 0) = g, \pi(j, k, g, l+1) = \langle 6, (g)_1, j, k, \pi(j, k, g, l) \rangle, \pi(j, k, g) = \pi(j, k, g, k)$  (cf. the proof of II).

XXI. *There is a primitive recursive function  $\iota(z, m)$  such that, if  $z$  is an index of  $\phi(\alpha)$ , and  $c$  are variables distinct from  $\alpha$  and consisting of  $m_0, \dots, m_s$  of types  $0, \dots, s$ , then  $\iota(z, \langle m_0, \dots, m_s \rangle)$  is an index of  $\phi(\alpha, c) = \phi(\alpha)$  with  $\lambda(\iota(z, \langle m_0, \dots, m_s \rangle)) = \lambda(z)$  (cf. III).*

**Proof.** We define  $\iota(z, m)$  by the course-of-values recursion

$$\iota(z, m) = \begin{cases} \langle (z)_0, m \cdot (z)_1, (z)_2 \rangle & \text{if } (z)_0 = 2 \vee (z)_0 = 9, \\ \langle (z)_0, m \cdot (z)_1, \iota((z)_2, m), \iota((z)_3, m) \rangle & \text{if } (z)_0 = 4 \vee (z)_0 = 5, \\ \langle 6, m \cdot (z)_1, (z)_2, (z)_3, \iota((z)_4, m) \rangle & \text{if } (z)_0 = 6, \\ \langle 8, m \cdot (z)_1, (z)_2, \iota((z)_3, m) \rangle & \text{if } (z)_0 = 8, \\ \langle (z)_0, m \cdot (z)_1 \rangle & \text{otherwise.} \end{cases}$$

XXII. *For each  $n \geq 1$ , there is a primitive recursive function  $\gamma_n(z, w, p)$  with the following property. Suppose  $\alpha$  contains exactly  $p$  type- $n$  variables; and let  $z, w$  be indices of  $\phi(\alpha, \sigma^n, b), \theta(\alpha, b, c, \tau^{n-1})$ , respectively. Then, for values of*

$a, b, c$  such that  $\lambda\tau^{n-1}\theta(a, b, c, \tau^{n-1})$  is completely defined and  $\phi(a, \sigma^n, b)$  is defined when  $\sigma^n = \lambda\tau^{n-1}\theta(a, b, c, \tau^{n-1})$ ,

$$(10) \quad \phi(a, \lambda\tau^{n-1}\theta(a, b, c, \tau^{n-1}), b) = \{\gamma_n(z, w, p)\}(a, b, c).$$

**Proof**, by induction on  $n$ . Take any  $n \geq 1$ . We shall take  $\gamma_n(z, w, p) = 0$ , except for  $Ix(z) \ \& \ (z)_{1,n} > p \ \& \ Ix(w) \ \& \ p_{n-1} \cdot (z)_1 \mid p_n \cdot (w)_1$ , when the appropriate following case shall apply. The first eight cases correspond to those in the proof of IV. We use induction on  $\{z\}(-)$  (cf. end 3.8) in proving (10) from the specifications for  $\gamma_n(z, w, p)$  worked out in the cases. In conclusion we shall show that a primitive recursive  $\gamma_n$  exists that meets all the specifications.

CASE 5.  $(z)_0 = 5$ . First, starting from  $z, w$  as indices of  $\phi(a, b, \sigma, b)$ ,  $\theta(a, b, b, c, \tau)$ , we construct indices (right column) for the functions (left column) used in the treatment of Case 5 for IV.

$$\theta_1(b, b, c, u, \tau): \quad A = \pi(0, (w)_{1,0} \dot{\div} (1 + (2 \dot{\div} n)), w).$$

$$\theta(a, b, b, b, c, u, \tau): \quad B = \langle 6, 4 \cdot (w)_1, 0, (w)_{1,0} + 1, \langle 6, 4 \cdot (w)_1, 0, (w)_{1,0} + 1, \iota(A, 4) \rangle \rangle.$$

$$\psi(b, b, c, u): \quad C = \gamma_n((z)_2, A, p).$$

$$\chi(a, b, b, b, c, u): \quad D = \gamma_n((z)_3, B, p).$$

$$\phi(a, b, b, c, u): \quad E = \langle 5, 2 \cdot (C)_1, C, D \rangle.$$

$$\phi_1(u, a, b, b, c): \quad F = \langle 6, (E)_1, 0, (E)_{1,0} \dot{\div} 1, E \rangle.$$

$$\eta(a, b, b, c): \quad G = \langle 3, (C)_1 \rangle.$$

$$\phi(a, b, b, c): \quad H = \langle 4, (C)_1, F, G \rangle.$$

Let  $\gamma_n(z, w, p) = H$ . Consider any particular  $(a_0, b_0, b_0, c_0)$  such that  $\lambda\tau\theta(a_0, b_0, b_0, c_0, \tau)$  is completely defined and  $\phi(a_0, b_0, \sigma_0, b_0)$  is defined for  $\sigma_0 = \lambda\tau\theta(a_0, b_0, b_0, c_0, \tau)$ . Then  $\psi(b_0, \sigma_0, b_0)$  is defined and  $\chi(a, b, b_0, \sigma_0, b_0)$  is defined for  $(a, b) = (0, \phi(0, b_0, \sigma_0, b_0)), \dots, (a_0 - 1, \phi(a_0 - 1, b_0, \sigma_0, b_0))$ . This information suffices for the applications of the hypothesis of the induction on  $\{z\}(-)$  to conclude, by the method of proof under IV (taking  $(b, b, c, u) = (b_0, b_0, c_0, a_0)$ , and considering only  $a \leq a_0$  for the induction on  $a$  at the end), that taking  $\phi(a, b, b, c) \simeq \{H\}(a, b, b, c)$  will make  $\phi(a_0, b_0, b_0, c_0) = \phi(a_0, b_0, \lambda\tau\theta(a_0, b_0, b_0, c_0, \tau), b_0)$ .

CASE 8.  $(z)_0 = 8$ . SUBCASE 1.  $n = (z)_2 \ \& \ p = 0$  (only for  $n > 1$ ).

$$\theta(\alpha^{i-2}, b, c, \tau): \quad A = \langle 6, p_{n-2} \cdot (w)_1, n - 2, (w)_{1,n-2}, \iota(w, p_{n-2}) \rangle.$$

$$\chi(\alpha^{i-2}, b, c): \quad B = \gamma_n((z)_3, A, 0).$$

$$\chi_1(b, c, \alpha^{i-2}): \quad C = \pi(n - 2, (B)_{1,n-2} \dot{\div} 1, B).$$

$$\phi(b, c): \quad D = \gamma_{n-1}(w, C, (w)_{1,n-1} \dot{\div} 1).$$

Let  $\gamma_n(z, w, p) = D$ . Consider any  $b_0, c_0$  such that  $\lambda\tau\theta(b_0, c_0, \tau)$  is completely defined and  $\phi(\sigma_0, b_0)$  is defined for  $\sigma_0 = \lambda\tau\theta(b_0, c_0, \tau)$ . Then  $\chi(\sigma_0, \alpha^{i-2}, b_0)$  is defined for each  $\alpha^{i-2}$ , and so, by hyp. ind. on  $\{z\}(-)$ , letting  $\chi(\alpha^{i-2}, b, c)$

$\simeq \{B\}(\alpha^{i-2}, b, c)$  does make  $\chi(\alpha^{i-2}, b_0, c_0) = \chi(\sigma_0, \alpha^{i-2}, b_0)$ . But putting  $\alpha_0^{j-1} = \lambda\alpha^{j-2} \chi(\sigma_0, \alpha^{j-2}, b_0)$ ,  $\phi(\sigma_0, b_0) = \theta(b_0, c_0, \alpha_0^{j-1})$ , so by hyp. ind. on  $n$  letting  $\phi(b, c) \simeq \{D\}(b, c)$  does make  $\phi(b_0, c_0) = \phi(\sigma_0, b_0)$ .

SUBCASE 2.  $n \neq (z)_2 \vee p \neq 0$ .

$$\theta(\alpha^i, \alpha^{i-2}, b, b, c, \tau): A = \langle 6, p_{(z)_2+2} \cdot (w_1), (z)_2 \div 2, (w)_{1, (z)_2+2}, \iota(w, p_{(z)_2+2}) \rangle.$$

$$\chi(\alpha^i, \alpha^{i-2}, b, b, c): B = \gamma_n((z)_3, A, p + \overline{\text{sg}} \mid n \div ((z)_2 \div 2) \mid).$$

$$\phi(\alpha^i, b, b, c): C = \langle 8, [(B)_1/p_{(z)_2+2}], (z)_2, B \rangle.$$

CASE 9.  $(z)_0 = 9$ . SUBCASE 1.  $p \geq (z)_{2,n}$  (so  $\sigma^n$  is one of the variables  $c$  for the schema S9). Let  $\gamma_n(z, w, p) = \langle 9, [(w)_1/p_{n-1}], (z)_2 \rangle$ .

SUBCASE 2.  $p < (z)_{2,n}$ . Write  $a = (a, b)$ ,  $b = (e, f)$  with  $(b, \sigma, e)$  the  $b$ , and  $f$  the  $c$ , for the schema. Now  $\phi(a, b, \sigma, b) \simeq \{a\}(b, \sigma, e)$  and  $\theta(a, b, b, c, \tau) \simeq \{S^1(w, a)\}(b, b, c, \tau)$  by XIII. So for values of  $a, b, b, c$  which make  $\sigma_0 = \lambda\tau \theta(a, b, b, c, \tau)$  completely defined and  $\{a\}(b, \sigma_0, e)$  defined, the hyp. ind. on  $\{z\}(-)$  gives that  $\phi(a, b, \lambda\tau \theta(a, b, b, c, \tau), b) = \{\gamma_n(a, S^1(w, a), p)\}(b, b, c)$ . So using XIII it will suffice to take  $\gamma_n(z, w, p) = S^3(D, g_n, w, p)$  where  $D$  is an index of  $\lambda g w p a b b c \{ \{g\}(a, S^1(w, a), p) \}(b, b, c)$  constructed as follows and  $g_n$  is an index of  $\gamma_n$ . Let  $e$  be an index of  $\lambda g w p a \{g\}(a, S^1(w, a), p)$ . We construct further indices.

$$\lambda g w p a b b c \{g\}(a, S^1(w, a), p): A = \iota(e, [(w)_1/2p_{n-1}]).$$

$$\lambda b b b c g w p a \{b\}(b, b, c): B_0 = \langle 9, 2^4 \cdot [(w)_1/p_{n-1}], [(w)_1/2p_{n-1}] \rangle.$$

$$B_{i+1} = \langle 6, (B_0)_1, 0, (B_0)_{1,0} \div 1, B_i \rangle.$$

$$\lambda b g w p a b b c \{b\}(b, b, c): C = \langle 6, (B_0)_1, 0, 4, B_4 \rangle.$$

$$\lambda g w p a b b c \{ \{g\}(a, S^1(w, a), p) \}(b, b, c): D = \langle 4, (A)_1, C, A \rangle.$$

CONCLUSION. Bringing together the definitions of  $\gamma_n(z, w, p)$  proposed in the cases, we can by the recursion theorem XIV pick an index  $g_n$  of  $\gamma_n$  (which  $g_n$  is called for in Case 9) so that the resulting recursion is satisfied. But with this  $g_n$  fixed, we can see as follows from the form of the recursion for  $\gamma_n(z, w, p)$  that  $\gamma_n$  is primitive recursive, using for  $n > 1$  the hyp. ind. on  $n$  by which  $\gamma_{n-1}$  (which occurs in Case 8, Subcase 1) is primitive recursive. The right side is built up primitive recursively from  $z, w, p$  and parts  $\gamma_n(Z, W, P)$  with  $Z < z$  (under the hypothesis of the case in which the part occurs) and  $Z, W, P$  primitive recursive functions of  $z, w, p$ . Thus the recursion is a course-of-values recursion on  $z$  with "nesting" or "substitution for the parameters"  $w, p$ , and hence by Péter [22, §§1, 2, especially Nr. 20, p. 632] or [23, §§3, 5] defines a primitive recursive function  $\gamma_n$ .

4.3. Now we recapitulate XXII, taking  $c$  empty and putting  $\phi(a, b) \simeq \{\gamma_n(z, w, p)\}(a, b)$ . This gives us a version of IV extended to partial and general recursive functions.

XXIII. For  $n \geq 1$ : If  $\phi(a, \sigma^n, b)$  and  $\theta(a, b, \tau^{n-1})$  are partial recursive, there is a partial recursive function  $\phi(a, b)$  such that

$$(11) \quad \phi(a, b) = \phi(a, \lambda\tau^{n-1}\theta(a, b, \tau^{n-1}), b)$$

for values of  $a, b$  for which  $\lambda\tau^{n-1}\theta(a, b, \tau^{n-1})$  is completely defined and  $\phi(a, \lambda\tau^{n-1}\theta(a, b, \tau^{n-1}), b)$  is defined. Thus: If  $\phi(a, \sigma^n, b)$  and  $\theta(a, b, \tau^{n-1})$  are general recursive, so is  $\phi(a, b) = \phi(a, \lambda\tau^{n-1}\theta(a, b, \tau^{n-1}), b)$ .

4.4. Using partial or general recursiveness in functions  $\Psi$ , and indices from  $\Psi$  (3.14), we obtain a relativized version XXIII\* of XXIII.

4.5. XXIV. For any  $j \geq 0$ : If  $\psi(\alpha^i, b)$  and  $\chi(a, \alpha^i, \alpha^{i+1}, b)$  are general recursive, so is the function  $\phi(a, \alpha^i, b)$  defined by

$$\begin{cases} \phi(0, \alpha^i, b) = \psi(\alpha^i, b), \\ \phi(a', \alpha^i, b) = \chi(a, \alpha^i, \lambda\alpha^i \phi(a, \alpha^i, b), b). \end{cases}$$

(Functional recursion; for  $j = 0$ , Péter's "primitive Rekursion der II-ten Stufe"<sup>(6)</sup>.)

**Proof.** Let  $\chi(z, a, \alpha^i, b)$  be the partial recursive function given by XXIII such that

$$(12) \quad \chi(z, a, \alpha^i, b) = \chi(a \div 1, \alpha^i, \lambda\alpha^i \{z\}(a \div 1, \alpha^i, b), b)$$

for values of  $z, a, b$  which make  $\lambda\alpha^i \{z\}(a \div 1, \alpha^i, b)$  completely defined. Using XV, let

$$\psi(z, a, \alpha^i, b) \simeq \begin{cases} \psi(\alpha^i, b) & \text{if } a=0, \\ \chi(z, a, \alpha^i, b) & \text{if } a \neq 0. \end{cases}$$

Using the recursion theorem XIV, pick an index  $e$  of  $\psi(e, a, \alpha^i, b)$ , and let  $\phi(a, \alpha^i, b) \simeq \{e\}(a, \alpha^i, b) \simeq \psi(e, a, \alpha^i, b)$ . Now, for any given  $b$ , we prove by induction on  $a$  that  $\phi(a, \alpha^i, b)$  is defined for every  $\alpha^i$  and satisfies the recursion in XXIV. BASIS.  $\phi(0, \alpha^i, b) \simeq \psi(e, 0, \alpha^i, b) \simeq \psi(\alpha^i, b)$ , which is defined since  $\psi$  is general recursive. IND. STEP.  $\phi(a', \alpha^i, b) \simeq \psi(e, a', \alpha^i, b) \simeq \chi(e, a', \alpha^i, b) \simeq \chi(a' \div 1, \alpha^i, \lambda\alpha^i \{e\}(a' \div 1, \alpha^i, b), b)$  (by (12), since by hyp. ind.  $\lambda\alpha^i \{e\}(a' \div 1, \alpha^i, b) = \lambda\alpha^i \phi(a, \alpha^i, b)$  is completely defined)  $\simeq \chi(a, \alpha^i, \lambda\alpha^i \phi(a, \alpha^i, b), b)$ .

4.6. XXV. There are primitive recursive functions  $\nu_0, \nu_1, \nu_2, \dots$  such that  $\nu_0(n)$  ( $\nu_j(n)$  for  $a j > 0$ ) is a primitive recursive index of  $\lambda\alpha_1 \dots \alpha_n \langle \alpha_1, \dots, \alpha_n \rangle$  (of  $\lambda\alpha_1^j \dots \alpha_n^j \tau^{j-1} \langle \alpha_1^j(\tau^{j-1}), \dots, \alpha_n^j(\tau^{j-1}) \rangle$ ).

**Proof** for  $j > 0$ , using induction on  $n$ . BASIS. Let  $\nu_j(0) = \langle 2, p_{j-1}, 1 \rangle$ . IND. STEP. Assume  $\nu_j(n)$  is an index of  $\lambda\alpha_1^j \dots \alpha_n^j \tau^{j-1} \langle \alpha_1^j(\tau^{j-1}), \dots, \alpha_n^j(\tau^{j-1}) \rangle$ . Let  $h$  be an index of  $\lambda n b \alpha^i(\tau^{j-1}) b \cdot p_n^{\alpha^i(\tau^{j-1})}$ . We construct further indices.

$$\lambda b \alpha_1^j \dots \alpha_n^j \alpha \tau^{j-1} b \cdot p_n^{\alpha^i(\tau^{j-1})} : A = \pi(j, n, \iota(S^1(h, n), p_j^n)).$$

$$\lambda \alpha_1^j \dots \alpha_n^j \alpha \tau^{j-1} \langle \alpha_1^j(\tau^{j-1}), \dots, \alpha_n^j(\tau^{j-1}) \rangle : B = \iota(\nu_j(n), p_j).$$

$$\lambda \alpha_1^j \dots \alpha_{n+1}^j \tau^{j-1} \langle \alpha_1^j(\tau^{j-1}), \dots, \alpha_{n+1}^j(\tau^{j-1}) \rangle : \nu_j(n+1) = \langle 4, (B)_1, A, B \rangle.$$

**5. Reduction of the inductive definition of  $\{z\}(a) \simeq w$  to an explicit definition. 5.1.** In computing  $\{z\}(a)$  the numbers  $n_0, \dots, n_r$  in general vary, but  $r$  does not increase (for by I with 3.10  $r$  remains fixed throughout the partial recursive description determined by  $z$ , while applications of S9 start new partial recursive descriptions determined by the  $a$  with the same or smaller  $r$ , etc.). In analyzing the inductive definition of  $\{z\}(a) \simeq w$  it will help to contract all the variables of each type  $j \leq r$  into one.

So for each  $r \geq 0$  we define (writing  $n_j = (z)_{1,j}$ ; cf. 2.1, 4.1)

$$\{z\}[a, \alpha^1, \dots, \alpha^r] \simeq \{z\}((a)_0, \dots, (a)_{n_0-1}, (\alpha^1)_0, \dots, (\alpha^1)_{n_1-1}, \dots, (\alpha^r)_0, \dots, (\alpha^r)_{n_r-1})$$

if  $Ix(z)$  and  $tp(z) \leq r$ ; otherwise  $\{z\}[a, \alpha^1, \dots, \alpha^r]$  shall be undefined. Then by (2), when  $z$  is an index for  $a_1, \dots, a_{n_0}, \alpha_1^1, \dots, \alpha_{n_1}^1, \dots, \alpha_1^r, \dots, \alpha_{n_r}^r$ ,

$$(13) \quad \{z\}(a_1, \dots, a_{n_0}, \alpha_1^1, \dots, \alpha_{n_1}^1, \dots, \alpha_1^r, \dots, \alpha_{n_r}^r) \simeq \{z\}[\langle a_1, \dots, a_{n_0} \rangle, \langle \alpha_1^1, \dots, \alpha_{n_1}^1 \rangle, \dots, \langle \alpha_1^r, \dots, \alpha_{n_r}^r \rangle].$$

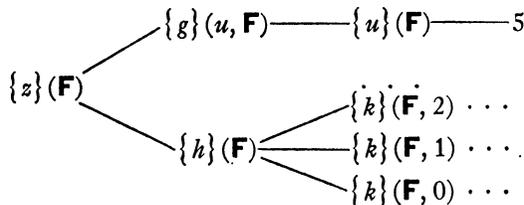
**5.2. XXVI.** For each  $r \geq 2$ , there are primitive recursive predicates  $I$  and  $J$  such that, when  $\{z\}[a, \alpha^1, \dots, \alpha^r]$  is defined,

$$(14) \quad \{z\}[a, \alpha^1, \dots, \alpha^r] = w \equiv (\beta^{r-1})(E\xi^{r-2})I(z, a, \alpha^1, \dots, \alpha^r, w, \beta^{r-1}, \xi^{r-2}),$$

$$(15) \quad \{z\}[a, \alpha^1, \dots, \alpha^r] = w \equiv (E\beta^{r-1})(\xi^{r-2})J(z, a, \alpha^1, \dots, \alpha^r, w, \beta^{r-1}, \xi^{r-2}).$$

The proof occupies 5.3–5.9. We assume  $r \geq 2$ , though afterwards (in 5.21–5.22) we shall use parts of the material for  $r = 0, 1$ . For 5.3–5.6, though of course  $z$  can take any value in the recursions, we assume for the discussion that  $Ix(z) \ \& \ tp(z) \leq r$ .

**5.3.** The “stages” in the computation of  $\{z\}(\mathbf{F})$  in the example of 3.9 can be arranged in a “tree,” thus:



Similarly (assuming  $Ix(z) \ \& \ tp(z) \leq r$ ) we can consider the stages in the computation of  $\{z\}[a, \alpha^1, \dots, \alpha^r]$  arranged in a tree. However to avoid syntactical considerations here, we consider the “positions” in the tree as being occupied, not by the expressions beginning with “ $\{z\}[a, \alpha^1, \dots, \alpha^r]$ ” at the 0- (or leftmost) position, but instead by the  $r+2$ -tuples of objects which occur successively as arguments of  $\lambda z a \alpha^1 \dots \alpha^r \{z\}[a, \alpha^1, \dots, \alpha^r]$  in carrying out the computation. Thus let  $(z_0, a_0, \alpha_0^1, \dots, \alpha_0^r)$

=  $(z, a, \alpha^1, \dots, \alpha^r)$ ; this is the  $r+2$ -tuple at the 0-position. Say a branch has been generated as far as the  $r+2$ -tuple  $(z_n, a_n, \alpha_n^1, \dots, \alpha_n^r)$  at its  $n$ -position. Then there are one, two or infinitely many choices of the  $r+2$ -tuple  $(z_{n+1}, a_{n+1}, \alpha_{n+1}^1, \dots, \alpha_{n+1}^r)$  at the  $n+1$ -position.

Here we exclude the possibility of zero choices, so each branch of the tree extends ad infinitum, by letting

$$(z_{n+1}, a_{n+1}, \alpha_{n+1}^1, \dots, \alpha_{n+1}^r) = (z_n, a_n, \alpha_n^1, \dots, \alpha_n^r)$$

in the cases (S1-S3, S7) when the computation terminates by the schema giving the value of  $\{z_n\} [a_n, \alpha_n^1, \dots, \alpha_n^r]$  outright. (We should accordingly redraw the tree shown above to replace the “- 5” by “-  $\{u\}(\mathbf{F}) - \{u\}(\mathbf{F}) \dots$ .”) Likewise, in the case for S9, which says  $\{z\}(a, b, c) \simeq \{a\}(b)$ , when the  $a$  is not an index for the  $b$ , we shall not go on from the  $r+2$ -tuple for  $\{z\}(a, b, c)$  to that for  $\{a\}(b)$ , but repeat the former ad infinitum. It is a consequence of  $\text{Ix}(z) \ \& \ \text{tp}(z) \leq r$  and these provisions, by induction on  $n$ , that along every branch for every  $n$ ,  $\text{Ix}(z_n) \ \& \ \text{tp}(z_n) \leq r$ .

Which  $r+2$ -tuple we choose in passing from an  $n$ -position already reached to an  $n+1$ -position we describe by a choice of the  $(n+1)$ st value  $\lambda\tau^{r-3} \rho(n, \tau^{r-3})$  of a function  $\lambda n \lambda\tau^{r-3} \rho(n, \tau^{r-3})$ , where in case  $r = 2$  the  $\tau^{r-3}$  is to be omitted so it is the  $(n+1)$ st value  $\rho(n)$  of  $\lambda n \rho(n)$  that is chosen. In the case of a branching at an application of S4, the choice of the lower  $r+2$ -tuple (which starts the computation of the  $\chi(b)$ ) is described by taking  $\lambda\tau \rho(n, \tau) = \lambda\tau 0$ , and of the upper  $r+2$ -tuple (for the  $\psi(\chi(b), b)$ ) by taking  $\lambda\tau \rho(n, \tau) = \lambda\tau 1$ . Similarly at an application of the second equation for S5, the choice of the lower  $r+2$ -tuple (for the  $\phi(a, b)$ ) is described by  $\lambda\tau \rho(n, \tau) = \lambda\tau 0$  and of the upper (for the  $\chi(a, \phi(a, b), b)$ ) by  $\lambda\tau \rho(n, \tau) = \lambda\tau 1$ . At an application of S8.2, the choice of the  $r+2$ -tuple which starts the computation of the  $\chi(\alpha^2, x, b)$  for a given  $x$  is described by taking  $\lambda\tau \rho(n, \tau) = \lambda\tau x = \lambda\tau \text{mp}_0^{r-2}(x, \tau)$  (cf. 2.3); and more generally, at an application of S8. $j$  ( $2 \leq j \leq r$ ), the choice of the  $r+2$ -tuple which starts the computation of the  $\chi(\alpha^j, \alpha^{j-2}, b)$  for a given  $\alpha^{j-2}$  is described by  $\lambda\tau \rho(n, \tau) = \lambda\tau \text{mp}_{j-2}^{r-2}(\alpha^{j-2}, \tau)$ . In all other cases,  $\lambda\tau \rho(n, \tau) = \lambda\tau 0$ .

An  $n$ -position in the tree is described by the first  $n$  choices, which are given by  $\lambda\tau \bar{\rho}(n; \tau)$  where  $\bar{\rho}(n; \tau) = \prod_{i < n} p_i^{\rho(i, \tau)+1}$ . A branch in its entirety is described by the function  $\lambda n \lambda\tau \rho(n, \tau)$  or simply  $\rho$ .

We would like to express  $(z_n, a_n, \alpha_n^1, \dots, \alpha_n^r)$  for the given  $z, a, \alpha^1, \dots, \alpha^r$  as a function of the position  $\gamma^{r-2} = \lambda\tau \bar{\rho}(n; \tau)$  by primitive recursion on  $n$ . However we cannot do just this. For, at a branching corresponding to an application of S4, the  $a_{n+1}$  for the upper  $n+1$ -position is not immediately forthcoming from the  $n+1$ -position  $\gamma^{r-2} = \lambda\tau \bar{\rho}(n+1; \tau)$  and the  $r+2$ -tuple  $(z_n, a_n, \alpha_n^1, \dots, \alpha_n^r)$  at the  $n$ -position, but depends (through the  $\chi(b)$  of  $\psi(\chi(b), b)$  which is contracted into that  $a_{n+1}$ ) on the outcome of the entire



To analyze this recursion, we have, for  $m > n$ ,  $\lambda\tau \rho(n, \tau) = \lambda\sigma(\bar{\rho}(m; \sigma))_n \div 1$  and  $\lambda\tau \bar{\rho}(n; \tau) * 2 = \lambda\sigma(\prod_{i < n} p_i^{(\bar{\rho}(m; \sigma))_i}) \cdot p_n$ , where  $\bar{\rho}(m; \sigma) = \{\lambda\tau \bar{\rho}(m; \tau)\}(\sigma)$ . For each  $n$ , let  $b_n = \langle z_n, a_n \rangle$ , so that

$$(17) \quad z_n = (b_n)_0, \quad a_n = (b_n)_1.$$

Using these relations, the simultaneous recursion for  $z_n$  and  $a_n$  gives

$$(18) \quad \begin{aligned} b_0 &= \langle z, a \rangle, \\ b_{n+1} &= \chi(n, b_n, \lambda\tau \bar{\rho}(m; \tau), \eta) \end{aligned} \quad (m > n)$$

with a primitive recursive  $\chi$ . Now define  $\beta$  by the primitive recursion

$$\begin{cases} \beta(0, z, a, \gamma^{r-2}, \eta) = \langle z, a \rangle, \\ \beta(n+1, z, a, \gamma^{r-2}, \eta) = \chi(n, \beta(n, z, a, \gamma^{r-2}, \eta), \gamma^{r-2}, \eta). \end{cases}$$

Then, by induction on  $n$ ,

$$(19) \quad b_n = \beta(n, z, a, \lambda\tau \bar{\rho}(m; \tau), \eta) \quad (m \geq n).$$

Taking  $m = n = \text{lh}(\{\lambda\tau \bar{\rho}(n; \tau)\}(\lambda\sigma^{r-4} 0))$  (omit  $\sigma^{r-4}$  if  $r \leq 3$ ),

$$(20) \quad b_n = \beta(z, a, \lambda\tau \bar{\rho}(n; \tau), \eta)$$

with a new primitive recursive  $\beta$ . This with (17) gives  $z_n, a_n$  by primitive recursive functions of  $z, a, \eta$  and the position  $\gamma = \lambda\tau \bar{\rho}(n; \tau)$ ; and more generally, (19) with (17) gives  $z_n, a_n$  by primitive recursive functions of  $z, a, \eta, n$  and any position  $\delta = \lambda\tau \bar{\rho}(m; \tau)$  as far or further out on any branch through the position  $\gamma$  in question.

It remains to deal similarly with  $\alpha_n^j$  for  $j = 1, \dots, r$ . First  $\alpha_0^j(v^{j-1}) = \alpha^j(v^{j-1})$ . Furthermore  $\alpha_{n+1}^j(v) = \alpha_n^j(v)$ , except in the case  $(z_n)_0 = 6$  &  $(z_n)_2 = j$ , when (writing  $k = (z_n)_3$ )

$$\alpha_{n+1}^j(v) = 2^{(\alpha_n^j(v))_k} \cdot \prod_{i < k} p_{i+1}^{(\alpha_n^j(v))_i} \cdot \prod_{k < i < \alpha_n^j(v)} p_i^{(\alpha_n^j(v))_i},$$

and in the case  $(z_n)_0 = 8$  &  $(z_n)_2 = j+2$ , when

$$\alpha_{n+1}^j(v) = 2^{\sigma^j(v)} \cdot \prod_{i < \alpha_n^j(v)} p_{i+1}^{(\alpha_n^j(v))_i}$$

where  $\sigma^j(v) = \text{pm}_j^{r-2}(\lambda\tau \rho(n, \tau), v)$  (cf. X). Using (17) and (19), thus

$$(21) \quad \begin{aligned} \alpha_0^j(v) &= \alpha^j(v), \\ \alpha_{n+1}^j(v) &= \chi_j(n, z, a, \alpha_n^j(v), \lambda\tau \bar{\rho}(m; \tau), \eta, v) \end{aligned} \quad (m > n)$$

with a primitive recursive  $\chi_j$ . Continuing from (21) as before from (18), we obtain primitive recursive  $\beta_j^i$ 's such that

$$(22) \quad \alpha_n^j(v) = \beta_j(n, z, a, \alpha^j(v), \lambda\tau \bar{\rho}(m; \tau), \eta, v) \quad (m \geq n),$$

$$(23) \quad \alpha_n^j(v) = \beta_j(z, a, \alpha^j(v), \lambda\tau \bar{\rho}(n; \tau), \eta, v).$$

5.5. A type- $r-2$  object  $\delta^{r-2}$  will describe a position in the tree under 5.3 exactly if (a)  $\delta^{r-2}$  is of the form  $\lambda\tau \bar{\rho}(m; \tau)$  for some number  $m$  and function  $\rho$  and (b) the value of  $\lambda\tau \rho(n, \tau)$  obeys certain restrictions for  $n=0, \dots, m-1$ . Now (a) is expressed by<sup>(3)</sup>

$$(24) \quad \begin{aligned} &(\tau)[\text{Seq}(\delta^{r-2}(\tau)) \ \& \ \text{lh}(\delta^{r-2}(\tau)) = \text{lh}(\delta^{r-2}(\lambda\sigma^{r-4} 0))] && (r > 2), \\ &\text{Seq}(\delta) && (r = 2). \end{aligned}$$

As to (b), suppose (a) is satisfied and the restrictions are obeyed for  $0, \dots, n-1$  where  $n < m$ . Then  $\gamma^{r-2} = \lambda\tau \bar{\rho}(n; \tau)$  is a position, at which the  $z_n$  and  $a_n$  are given by (19) with (17) from  $z, a, \eta, n$  and  $\delta^{r-2} = \lambda\tau \bar{\rho}(m; \tau)$ . The restrictions for  $n$  are expressed by

$$(25) \quad \begin{aligned} &(\tau)[\rho(n, \tau) = 0] \vee \{ [(z_n)_0 = 4 \vee ((z_n)_0 = 5 \ \& \ (a_n)_0 > 0)] \ \& \ (\tau)[\rho(n, \tau) = 1] \} \\ &\vee \{ (z_n)_0 = 8 \ \& \ (z_n)_2 = 2 \ \& \ (\tau)[\rho(n, \tau) = \text{pm}_0^{r-2}(\lambda\tau \rho(n, \tau))] \} \\ &\vee (Ej)_{3 \leq j \leq r} \{ (z_n)_0 = 8 \ \& \ (z_n)_2 = j \\ &\ \& \ (\tau)[\rho(n, \tau) = \text{mp}_{j-2}^{r-2}(\lambda v^{j-3} \ \text{pm}_{j-2}^{r-2}(\lambda\tau \rho(n, \tau), v^{j-3}), \tau)] \} && (r > 2), \\ &\rho(n) = 0 \vee \{ [(z_n)_0 = 4 \vee ((z_n)_0 = 5 \ \& \ (a_n)_0 > 0)] \ \& \ \rho(n) = 1 \} \\ &\vee \{ (z_n)_0 = 8 \ \& \ (z_n)_2 = 2 \} && (r = 2). \end{aligned}$$

A formula expressing (b) is obtained by prefixing  $(n)_{n < m}$  to (25), and replacing  $m$  by  $\text{lh}(\delta^{r-2}(\lambda\sigma^{r-4} 0))$ ,  $\rho(n, \tau)$  by  $(\delta^{r-2}(\tau))_n \div 1$ ,  $z_n$  by  $(\beta(n, z, a, \delta^{r-2}, \eta))_0$  and  $a_n$  by  $(\beta(n, z, a, \delta^{r-2}, \eta))_1$ . Forming the conjunction of this with (24), and for  $r > 2$  advancing the quantifiers  $(\tau^{r-3})$  by predicate calculus and contracting them by (3), we obtain a primitive recursive predicate  $P$  such that (omitting  $\tau^{r-3}$  for  $r = 2$ )

$$(26) \quad \{ \delta^{r-2} \text{ is a position} \} \equiv (\tau^{r-3})P(z, a, \delta^{r-2}, \eta^{r-1}, \tau^{r-3}).$$

5.6. The tree described in 5.3 for given  $(z, a, \alpha^1, \dots, \alpha^r)$  and  $\eta$  does correspond to the computation of  $\{z\} [a, \alpha^1, \dots, \alpha^r]$ , if  $\eta(\gamma)$  at each position  $\gamma$  is the value of  $\{z_n\} [a_n, \alpha_n^1, \dots, \alpha_n^r]$  there. (Only if  $\{z\} [a, \alpha^1, \dots, \alpha^r]$  is defined are there such  $\eta$ 's.) However we must study the tree for  $\eta$ 's in general.

For any  $\eta$ , let us say  $\eta$  is *locally correct* at a given position  $\gamma$ , if  $\eta(\gamma)$  has the right value in case  $\gamma$  corresponds to a schema application (of S1-S3, S7) that gives a value outright, and otherwise if  $\eta(\gamma)$  and the numbers  $\eta(\delta)$  for each of the one or more  $n+1$ -positions issuing from the  $n$ -position  $\gamma$  have the relationship required by the schema application to which  $\gamma$  corresponds.

We analyze this notion, assuming  $\gamma$  is a position. In the case of a node, only the relationship of  $\eta(\gamma)$  to the  $\eta(\delta)$  at the upper  $n+1$ -position is in question, since the relationship to the  $\eta(\delta)$  at the lower  $n+1$ -position has already

been taken care of in the tree construction by incorporating that  $\eta(\delta)$  into the  $a_{n+1}$  at the upper  $n+1$ -position. The local correctness of  $\eta$  at  $\gamma$  is expressed by

$$\begin{aligned} & \{(z_n)_0=1 \ \& \ \eta(\gamma)=(a_n)_0+1\} \vee \{(z_n)_0=2 \ \& \ \eta(\gamma)=(z_n)_2\} \vee \{(z_n)_0=3 \ \& \ \eta(\gamma)=(a_n)_0\} \\ & \vee \{[(z_n)_0=4 \vee ((z_n)_0=5 \ \& \ (a_n)_0>0)] \ \& \ \eta(\gamma)=\eta(\lambda\tau \ \gamma(\tau) \ * \ 2^2)\} \\ & \vee \{[(z_n)_0=5 \ \& \ (a_n)_0=0] \vee (z_n)_0=6 \vee (z_n)_0=9\} \ \& \ \eta(\gamma)=\eta(\lambda\tau \ \gamma(\tau) \ * \ 2)\} \\ & \vee \{(z_n)_0=7 \ \& \ \eta(\gamma)=(\alpha_n^1((a_n)_0))_0\} \vee (Ej)_{2 \leq j \leq r} \{(z_n)_0=8 \ \& \ (z_n)_2=j \ \& \\ & \ \eta(\gamma)=(\alpha_n^j(\lambda\sigma^{j-2} \ \eta(\lambda\tau \ \gamma(\tau) \ * \ [2 \ \exp \ 1 + \text{mp}_{j-2}(\sigma^{j-2}, \ \tau)])))_0\}. \end{aligned}$$

Eliminating  $z_n, a_n, \alpha_n^1, \dots, \alpha_n^r$  by (17), (20) and (23) with  $\gamma = \lambda\tau \ \bar{\rho}(n; \ \tau)$ , we obtain a primitive recursive predicate  $C$  such that, for  $\gamma$  a position,

$$(27) \quad \{\gamma^{r-1} \text{ is locally correct at } \gamma^{r-2}\} \equiv C(z, a, \alpha^1, \dots, \alpha^r, \gamma^{r-2}, \gamma^{r-1}).$$

**5.7.** Consider, as we have been doing, the tree for  $(z, a, \alpha^1, \dots, \alpha^r)$  based on a given  $\eta$ , and at any position  $\gamma$  let  $(z_n, a_n, \alpha_n^1, \dots, \alpha_n^r)$  be the  $r+2$ -tuple which occupies it. If  $\{z\} [a, \alpha^1, \dots, \alpha^r]$  is defined, and  $\eta$  is locally correct at every position  $\gamma$ , then at every position  $\gamma$   $\{z_n\} [a_n, \alpha_n^1, \dots, \alpha_n^r]$  is defined and  $= \eta(\gamma)$ , i.e.  $\eta$  gives the correct values.

**Proof.** Assume  $\{z\} [a, \alpha^1, \dots, \alpha^r]$  is defined. Then  $\text{Ix}(z) \ \& \ \text{tp}(z) \leq r$  by 5.1, and we can use induction on  $\{z\}(-)$  (end 3.8). Assume  $\eta$  is locally correct. We give two of the cases for the induction.

CASE 1.  $(z)_0 = 1$ , i.e.  $z$  is an index  $\langle 1, \langle n_0, \dots, n_r \rangle \rangle$  for an application of S1. So  $\{z\} [a, \alpha^1, \dots, \alpha^r] = (a)_0+1$ . By 5.3, the tree is unbranched with  $(z_n, a_n, \alpha_n^1, \dots, \alpha_n^r) = (z, a, \alpha^1, \dots, \alpha^r)$  for every  $n$ . So at every position  $\gamma$ ,  $\{z_n\} [a_n, \alpha_n^1, \dots, \alpha_n^r] = (a)_0+1 = \eta(\gamma)$ , since  $\eta$  is locally correct at  $\gamma$ .

CASE 4.  $(z)_0 = 4$ , i.e.  $z$  is an index  $\langle 4, \langle n_0, \dots, n_r \rangle, g, h \rangle$  for an application of S4. So  $\{z\} [a, \alpha^1, \dots, \alpha^r] = \{z\} (b) = \{g\} (\{h\} (b), b), \{h\} [a, \alpha^1, \dots, \alpha^r] = \{h\} (b)$  and  $\{g\} [a_1, \alpha^1, \dots, \alpha^r] = \{g\} (\{h\} (b), b)$ , where  $g = (z)_2, h = (z)_3, a_1 = 2^{[h](b)}$ .  $\prod_{i < a} p_{i+1}^{(a)_i}$ . Each of these three expressions is defined, since the first is<sup>(10)</sup>. In the tree for  $(z, a, \alpha^1, \dots, \alpha^r)$  (at the 0-position) and the given  $\eta$ , the next  $r+2$ -tuples are  $(h, a, \alpha^1, \dots, \alpha^r)$  at the lower 1-position, and at the upper 1-position  $(g, a_1^*, \alpha^1, \dots, \alpha^r)$  where  $a_1^* = 2^{r(\gamma)} \cdot \prod_{i < a} p_{i+1}^{(a)_i}$  for  $\gamma = \lambda\tau \ 2$ . In the construction of any tree, for any  $r+2$ -tuple  $(z_n, a_n, \alpha_n^1, \dots, \alpha_n^r)$ , the set of the next  $r+2$ -tuples (i.e. how many and what they are) is completely determined by  $(z_n, a_n, \alpha_n^1, \dots, \alpha_n^r)$ , except in the case of a node when for the upper next  $r+2$ -tuple the value of  $\eta$  at the position  $\gamma$  of the lower next  $r+2$ -tuple is also used. It follows that the trees for  $(h, a, \alpha^1, \dots, \alpha^r)$  and  $(g, a_1^*, \alpha^1, \dots, \alpha^r)$  are exactly the lower and upper subtrees which remain from the tree for  $(z, a, \alpha^1, \dots, \alpha^r)$  upon omitting the initial  $r+2$ -tuple of that, when the functions  $\eta_0$  and  $\eta_1$  used for the subtrees are respectively the functions which correlate to each position the

number correlated to the corresponding position of the whole tree, i.e.  $\eta_0(\gamma) = \eta(\lambda\tau 2^*\gamma(\tau))$  and  $\eta_1(\gamma) = \eta(\lambda\tau 2^{2^*}\gamma(\tau))$ . So the local correctness of  $\eta$  throughout the whole tree implies that of  $\eta_0$  and  $\eta_1$  throughout the respective subtrees. Applying the hypothesis of the induction to the lower subtree, at each  $\gamma \{z_n\} [a_n, \alpha_n^1, \dots, \alpha_n^r]$  is defined and  $= \eta_0(\gamma)$ , i.e.  $\eta_0$  gives the correct values throughout the lower subtree. Hence in particular,  $\eta_0(\lambda\tau 1) = \{h\}(b) = \eta(\lambda\tau 2)$ ; so  $a_1^* = a_1$ . So  $\{g\} [a_1^*, \alpha^1, \dots, \alpha^r]$  is defined, since it is  $\{g\} [a_1, \alpha^1, \dots, \alpha^r]$ ; and by applying the hyp. ind. to the upper subtree, we can conclude that at each position  $\gamma \{z_n\} [a_n, \alpha^1, \dots, \alpha^r]$  is defined and  $= \eta_1(\gamma)$ , i.e.  $\eta_1$  gives correct values throughout that. These correct values under  $\eta_0$  and  $\eta_1$  in the two subtrees become correct values under  $\eta$  at the corresponding positions of the whole tree. Finally, by the local correctness of  $\eta$  at the 0-position  $\lambda\tau 1$  of the whole tree, the value under  $\eta$  is correct also there.

**5.8.** *If  $\{z\} [a, \alpha^1, \dots, \alpha^r]$  is defined, then there is a function  $\eta$  such that, in the tree for  $(z, a, \alpha^1, \dots, \alpha^r)$  based on  $\eta$ , the function  $\eta$  is locally correct at every position  $\gamma$ .*

The proof is similar to that in 5.7, using the result of 5.7 to identify  $a_1^*$  and  $a_1$  in Case 4 (and similarly in Case 5 for  $(a)_0 > 0$ ).

**5.9.** To complete the proof of XXVI, assume that  $\{z\} [a, \alpha^1, \dots, \alpha^r]$  is defined. Then by 5.8 there is an  $\eta$  which is locally correct throughout the tree for  $(z, a, \alpha^1, \dots, \alpha^r)$  based on that  $\eta$ . By 5.7, for each such  $\eta$ , the values are defined and given correctly throughout the tree, in particular at the 0-position, so  $\eta(\lambda\tau 1) = \{z\} [a, \alpha^1, \dots, \alpha^r]$ . Thus

$$\begin{aligned} & \{z\} [a, \alpha^1, \dots, \alpha^r] = w \\ (28) \quad & \equiv (\eta) \{ (\gamma) [(\gamma \text{ is a position}) \rightarrow (\eta \text{ is locally correct at } \gamma)] \rightarrow \eta(\lambda\tau 1) = w \} \\ & \equiv (E\eta) \{ (\gamma) [(\gamma \text{ is a position}) \rightarrow (\eta \text{ is locally correct at } \gamma)] \ \& \ \eta(\lambda\tau 1) = w \}. \end{aligned}$$

Using (26) and (27) in the two expressions at the right (omitting  $\tau^{r-3}$  for  $r=2$ ), advancing the quantifiers, and for  $r > 2$  simplifying the resulting prefixes  $(\eta^{r-1})(E\gamma^{r-2})(\tau^{r-3})$  and  $(E\eta^{r-1})(\gamma^{r-2})(E\tau^{r-3})$  by use of  $(\bar{8})$ ,  $(\bar{4})$ ,  $(\bar{3})$  and (8), (4), (3), we obtain the forms in (14) and (15).

**5.10. XXVII.** *For each  $r \geq 2$  there is a primitive recursive predicate  $K$ , and for each  $r > 2$  a primitive recursive predicate  $L$ , such that*

$$(29) \quad \begin{aligned} & (\{z\} [a, \alpha^1, \dots, \alpha^r] \text{ is defined}) \\ & \equiv (\beta^{r-1})(E\xi^{r-2})K(z, a, \alpha^1, \dots, \alpha^r, \beta^{r-1}, \xi^{r-2}) \quad (r \geq 2), \end{aligned}$$

$$(30) \quad \begin{aligned} & (\{z\} [a, \alpha^1, \dots, \alpha^r] \text{ is defined}) \\ & \equiv (E\beta^{r-1})(\xi^{r-2})L(z, a, \alpha^1, \dots, \alpha^r, \beta^{r-1}, \xi^{r-2}) \quad (r > 2). \end{aligned}$$

The proof, continuing from 5.3–5.8, occupies 5.11–5.18. We assume  $r \geq 2$ , and for the discussion in 5.11–5.12  $Ix(z) \ \& \ tp(z) \leq r$ . (Cf. LI below.)

5.11. Let us say  $\gamma^{r-2}$  is *below* a position  $\delta^{r-2}$ , if  $\delta^{r-2}$  is the upper  $n+1$ -position issuing from a node (the node being an  $n$ -position), and  $\gamma^{r-2}$  is an  $m$ -position with  $m \geq n+1$  on a branch through the lower  $n+1$ -position issuing from the node.

We analyze this notion, assuming  $\delta$  a position. First, for  $\gamma$  to be below  $\delta$ , the position  $\delta$  must be an  $n+1$ -position, i.e.

$$(31) \quad \text{lh}(\delta(\lambda\sigma^{r-4} 0)) > 0.$$

In this case,  $n = \text{lh}(\delta(\lambda\sigma^{r-4} 0)) \div 1$ , and the  $z_n, a_n$  at the  $n$ -position are given from  $\delta = \lambda\tau \bar{\rho}(n+1; \tau)$  by (17) and (19) with  $m = n+1$ . Next, this  $n$ -position must be a node with  $\delta$  the upper  $n+1$ -position, i.e.

$$(32) \quad [(z_n)_0 = 4 \vee ((z_n)_0 = 5 \ \& \ (a_n)_0 > 0)] \ \& \ \text{pm}_0^{r-2}(\lambda\tau (\delta(\tau))_n \div 1) = 1.$$

In this case the lower  $n+1$ -position is  $\lambda\tau [\delta(\tau)/p_n]$ . Finally,  $\gamma$  must be a position at least as far out as  $\lambda\tau [\delta(\tau)/p_n]$  on a branch through the latter, i.e. (using (26))<sup>(3)</sup>

$$(33) \quad (\tau)P(z, a, \gamma, \eta, \tau) \ \& \ (\tau)\text{Ext}(\gamma(\tau), [\delta(\tau)/p_n]).$$

Forming the conjunction of (31)–(33), eliminating  $z_n, a_n$  and  $n$  as indicated, and for  $r > 2$  advancing and contracting the quantifiers, we obtain a primitive recursive predicate  $B$  such that, for  $\delta$  a position (and omitting  $\tau^{r-3}$  for  $r = 2$ ),

$$(34) \quad \{\gamma^{r-2} \text{ is below } \delta^{r-2}\} \equiv (\tau^{r-3})B(z, a, \gamma^{r-2}, \delta^{r-2}, \eta^{r-1}, \tau^{r-3}).$$

We say  $\gamma$  is *below* a branch of the tree, if  $\gamma$  is below some position on the branch.

5.12. The computation *terminates* at a given position  $\gamma = \lambda\tau \bar{\rho}(n; \tau)$  if S1–S3 or S7 applies there, i.e. if  $(z_n)_0 = 1 \vee (z_n)_0 = 2 \vee (z_n)_0 = 3 \vee (z_n)_0 = 7$ . Using (20) with (17), we obtain a primitive recursive  $T$  such that, for  $\rho$  a branch,

$$(35) \quad \{\text{the computation terminates at the } n\text{-position on } \rho\} \equiv T(z, a, n, \rho, \eta^{r-1}).$$

5.13. Let  $D(z, a, \alpha^1, \dots, \alpha^r) \equiv \{\text{Ix}(z) \ \& \ \text{tp}(z) \leq r, \text{ and for every } \eta, \text{ in the tree constructed for } (z, a, \alpha^1, \dots, \alpha^r) \text{ on the basis of } \eta, \text{ the computation terminates along each branch } \rho \text{ below which } \eta \text{ is locally correct}\}$ .

To analyze this, note that “ $\rho$  is a branch” is expressed using (26) by  $(n)(\tau^{r-3})P(z, a, \lambda\tau \bar{\rho}(n; \tau), \eta^{r-1}, \tau^{r-3})$ . Using also (34), (27) and (35),

$$(36) \quad \begin{aligned} D(z, a, \alpha^1, \dots, \alpha^r) &\equiv \text{Ix}(z) \ \& \ \text{tp}(z) \leq r \ \& \\ &(\eta^{r-1})(\rho) \{ (n)(\tau^{r-3})P(z, a, \lambda\tau \bar{\rho}(n; \tau), \eta^{r-1}, \tau^{r-3}) \\ &\quad \ \& \ (n)(\gamma^{r-2})[(\tau^{r-3})B(z, a, \gamma^{r-2}, \lambda\tau \bar{\rho}(n; \tau), \eta^{r-1}, \tau^{r-3}) \\ &\quad \rightarrow C(z, a, \alpha^1, \dots, \alpha^r, \gamma^{r-2}, \eta^{r-1})] \rightarrow (En)T(z, a, n, \rho, \eta^{r-1}) \}. \end{aligned}$$

For  $r > 2$ , and any two-place function  $\lambda n \tau^{r-3} \rho(n, \tau^{r-3})$ , let  $\rho^{r-2}(\sigma^{r-3}) = \rho((\sigma^{r-3})_0^0, (\sigma^{r-3})_1)$  (cf. 2.4); then by (9),  $\rho(n, \tau^{r-3}) = \rho^{r-2}(\langle n, \tau^{r-3} \rangle)$ . Using this for  $r > 2$  to replace the quantification of the two-place  $\rho$  by quantification of  $\rho^{r-2}$  (for  $r = 2$ ,  $\rho$  is a one-place function  $\rho^1$ ), then advancing and contracting quantifiers on the right (cf. 2.5), we obtain a primitive recursive predicate  $K$  such that

$$(37) \quad D(z, a, \alpha^1, \dots, \alpha^r) \equiv (\beta^{r-1})(E\xi^{r-2})K(z, a, \alpha^1, \dots, \alpha^r, \beta^{r-1}, \xi^{r-2}).$$

**5.14.** *If  $\{z\} [a, \alpha^1, \dots, \alpha^r]$  is defined, then  $D(z, a, \alpha^1, \dots, \alpha^r)$ .*

**Proof.** Assume  $\{z\} [a, \alpha^1, \dots, \alpha^r]$  is defined. Then  $Ix(z) \ \& \ tp(z) \leq r$ , and we use induction on  $\{z\} (-)$  to show that the other conjunctive member of  $D(z, a, \alpha^1, \dots, \alpha^r)$  holds.

CASE 1.  $(z)_0 = 1$ . Then  $(z_0)_0 = 1$ , which gives  $T(z, a, 0, \rho, \eta)$  and thence  $(En)T(z, a, n, \rho, \eta)$ , for any  $\eta$  and the only branch  $\rho$ .

CASE 4.  $(z)_0 = 4$ . Then the tree for  $(z, a, \alpha^1, \dots, \alpha^r)$  based on any given  $\eta$  begins with a node, with  $(h, a, \alpha^1, \dots, \alpha^r)$  at the lower 1-position and  $(g, a_1^*, \alpha^1, \dots, \alpha^r)$  at the upper (cf. Case 4 in 5.7). Consider any given branch  $\rho$  below which the  $\eta$  is locally correct.

SUBCASE 0. The 1-position on  $\rho$  is the lower 1-position issuing from the node, i.e. is occupied by  $(h, a, \alpha^1, \dots, \alpha^r)$ . Then the branch  $\rho$  minus its 0-position corresponds to a branch  $\rho_0$  in the lower subtree, i.e. in tree for  $(h, a, \alpha^1, \dots, \alpha^r)$  based on  $\eta_0$ , below which  $\eta_0$  is locally correct. So, since  $\{h\} [a, \alpha^1, \dots, \alpha^r]$  is defined, by the hyp. ind. the computation terminates along  $\rho_0$  in the lower subtree, i.e.  $(En)T(h, a, n, \rho_0, \eta_0)$ , and hence along  $\rho$  in the whole tree (with  $n$  one greater), i.e.  $(En)T(z, a, n, \rho, \eta)$ .

SUBCASE 1. The 1-position on  $\rho$  is the upper one, i.e. is occupied by  $(g, a_1^*, \alpha^1, \dots, \alpha^r)$ . Then the local correctness of  $\eta$  below  $\rho$  in the whole tree entails the local correctness of  $\eta_0$  in the lower subtree. Hence by 5.7  $\eta_0$  gives the correct values in the lower subtree, so  $a_1^* = a_1$ . So  $\{g\} [a_1^*, \alpha^1, \dots, \alpha^r]$  is defined, and we can apply the hyp. ind. to  $\{g\} [a_1^*, \alpha^1, \dots, \alpha^r]$  (in the same manner as to  $\{h\} (a, \alpha^1, \dots, \alpha^r)$  in Subcase 0) to conclude that  $(En)T(g, a_1^*, n, \rho_1, \eta_1)$  and hence  $(En)T(z, a, n, \rho, \eta)$ .

**5.15.** *If  $D(z, a, \alpha^1, \dots, \alpha^r)$ , then  $\{z\} [a, \alpha^1, \dots, \alpha^r]$  is defined.*

**Proof.** Assume  $\{z\} [a, \alpha^1, \dots, \alpha^r]$  is undefined. To conclude that then  $\bar{D}(z, a, \alpha^1, \dots, \alpha^r)$ , assume further that  $Ix(z) \ \& \ tp(z) \leq r$ . We shall "construct" a function  $\eta$ , and a branch  $\rho$  of the tree for  $(z, a, \alpha^1, \dots, \alpha^r)$  based on  $\eta$ , such that at each  $n$ -position along  $\rho$   $\{z_n\} [a_n, \alpha_n^1, \dots, \alpha_n^r]$  is undefined, and  $\eta$  is locally correct below  $\rho$ . This will contradict the second conjunctive member of  $D(z, a, \alpha^1, \dots, \alpha^r)$ ; for wherever  $\{z_n\} [a_n, \alpha_n^1, \dots, \alpha_n^r]$  is undefined, the computation is unterminated.

The construction will proceed by stages for  $n = 0, 1, 2, \dots$ . At Stage  $n$ , the first  $n+1$   $r+2$ -tuples  $(z_0, a_0, \alpha_0^1, \dots, \alpha_0^r), \dots, (z_n, a_n, \alpha_n^1, \dots, \alpha_n^r)$  along  $\rho$  will have been picked, for each of these the function value  $\{z_i\}$

$[a_i, \alpha_i^1, \dots, \alpha_i^r]$  will be undefined,  $\eta(\delta)$  will have been picked at all positions  $\delta$  below the positions of the  $n+1$   $r+2$ -tuples already picked,  $\eta$  will be locally correct at each  $\delta$  for which  $\eta(\delta)$  has thus been picked, and the further  $r+2$ -tuples along  $\rho$  and values of  $\eta$  will not yet have been picked.

At Stage 0 we necessarily have  $(z_0, a_0, \alpha_0^1, \dots, \alpha_0^r) = (z, a, \alpha^1, \dots, \alpha^r)$ , the function value  $\{z_0\} [a_0, \alpha_0^1, \dots, \alpha_0^r]$  is undefined by our assumption that  $\{z\} [a, \alpha^1, \dots, \alpha^r]$  is undefined, and no values of  $\eta$  have been picked.

Now we give some cases for the step from Stage  $n$  to Stage  $n+1$ .

CASE 1.  $(z_n)_0 = 1$ . This case can be excluded, since under it  $\{z_n\} [a_n, \alpha_n^1, \dots, \alpha_n^r]$  would be defined (with value  $(a_n)_0 + 1$ ).

CASE 4.  $(z_n)_0 = 4$ . Then  $\{z_n\} [a_n, \alpha_n^1, \dots, \alpha_n^r] \simeq \{g\} (\{h\}(b), b)$  (cf. Case 4 in 5.7).

SUBCASE 0.  $\{h\}(b)$  is undefined. Then we take its  $r+2$ -tuple  $(h, a_n, \alpha_n^1, \dots, \alpha_n^r)$  as  $(z_{n+1}, a_{n+1}, \alpha_{n+1}^1, \dots, \alpha_{n+1}^r)$  and pick no further values of  $\eta$ . Clearly the requirements are met.

SUBCASE 1.  $\{h\}(b)$  is defined. By 5.8, there is a function  $\eta_0$  such that throughout the tree for  $(h, a_n, \alpha_n^1, \dots, \alpha_n^r)$  based on  $\eta_0$  the function  $\eta_0$  is locally correct. By 5.7, the values which this  $\eta_0$  gives are correct; in particular,  $\eta_0(\lambda\tau 1) = \{h\}(b)$ . So if as our  $(z_{n+1}, a_{n+1}, \alpha_{n+1}^1, \dots, \alpha_{n+1}^r)$  we take  $(g, a_{n,1}, \alpha_{n,1}^1, \dots, \alpha_{n,1}^r)$  where  $a_{n,1} = 2^{\{h\}(b)}$ .  $\prod_{i < a_n} p_{i+1}^{(a_n)^i}$ , and if at the same time we extend the selection of values of  $\eta$  by employing at positions below the  $n+1$ -position just filled the values given by  $\eta_0$  at the corresponding positions of the tree for  $(h, a_n, \alpha_n^1, \dots, \alpha_n^r)$  based on  $\eta_0$ , then  $(z_{n+1}, a_{n+1}, \alpha_{n+1}^1, \dots, \alpha_{n+1}^r)$  will meet the condition for occupying an  $n+1$ -position based on  $\eta$ . Now  $\{z_{n+1}\} [a_{n+1}, \alpha_{n+1}^1, \dots, \alpha_{n+1}^r] \simeq \{g\} (\{h\}(b), b) \simeq \{z_n\} [a_n, \alpha_n^1, \dots, \alpha_n^r]$ , which is undefined; and  $\eta$  is locally correct at all points below the segment of  $\rho$  thus far chosen.

CASE 8.j.  $(z_n)_0 = 8$  &  $(z_n)_2 = j$ . Then  $\{z_n\} [a_n, \alpha_n^1, \dots, \alpha_n^r] \simeq \{z_n\} ((\alpha_n^j)_0, b)$ ,  $\simeq (\alpha_n^j (\lambda\sigma^{j-2} \chi((\alpha_n^j)_0, \sigma^{j-2}, b)))_0$ . Since  $\{z_n\} [a_n, \alpha_n^1, \dots, \alpha_n^r]$  is undefined,  $\chi((\alpha_n^j)_0, \sigma^{j-2}, b)$  is undefined for some  $\sigma^{j-2}$ . Choosing such a  $\sigma^{j-2}$ , we take as our  $(z_{n+1}, a_{n+1}, \alpha_{n+1}^1, \dots, \alpha_{n+1}^r)$  the  $r+2$ -tuple for  $\chi((\alpha_n^j)_0, \sigma^{j-2}, b)$ , i.e. we make the choice described by  $\lambda\tau \rho(n, \tau) = \lambda\tau \text{mp}_{j-2}^{r-2}(\sigma^{j-2}, \tau)$ , and do not choose further values of  $\eta$ .

When  $(z_n, a_n, \alpha_n^1, \dots, \alpha_n^r)$  has been thus picked for every  $n$ , the branch  $\rho$  will have been constructed, and the value of  $\eta(\delta)$  will have been picked and will be locally correct, at exactly each position  $\delta$  below  $\rho$ . We complete the construction of  $\eta$  by taking  $\eta(\delta) = 0$  for all other  $\delta$ .

5.16. Let  $E(z, a, \alpha^1, \dots, \alpha^r) \equiv \{Ix(z) \ \& \ \text{tp}(z) \leq r$ , and there is an  $\eta$  such that, in the tree constructed for  $(z, a, \alpha^1, \dots, \alpha^r)$  on the basis of  $\eta$ , the function  $\eta$  is locally correct at every position, and the computation terminates along every branch  $\}$ .

Using (26), (27) and (35) (omitting  $r^{r-3}$  for  $r = 2$ ),

$$\begin{aligned}
 & E(z, a, \alpha^1, \dots, \alpha^r) \equiv \text{Ix}(z) \ \& \ \text{tp}(z) \leq r \\
 (38) \quad & \& \ (E\eta^{r-1}) \{ (\gamma^{r-2}) [(\tau^{r-3}) P(z, a, \gamma^{r-2}, \eta^{r-1}, \tau^{r-3}) \\
 & \hspace{15em} \rightarrow C(z, a, \alpha^1, \dots, \alpha^r, \gamma^{r-2}, \eta^{r-1})] \\
 & \& \ (\rho) [(n)(\tau^{r-3}) P(z, a, \lambda\tau \bar{\rho}(n; \tau), \eta^{r-1}, \tau^{r-3}) \rightarrow (En) T(z, a, n, \rho, \eta^{r-1})] \}.
 \end{aligned}$$

For  $r > 2$ , proceeding as in 5.13 we obtain a primitive recursive  $L$  such that

$$\begin{aligned}
 (39) \quad & E(z, a, \alpha^1, \dots, \alpha^r) \\
 & \equiv (E\beta^{r-1})(\xi^{r-2})L(z, a, \alpha^1, \dots, \alpha^r, \beta^{r-1}, \xi^{r-2}) \quad (r > 2)
 \end{aligned}$$

For  $r = 2$ , the quantifier  $(\rho)$  is of type  $1 = r - 1$  instead of (after contraction)  $r - 2$ , so  $(\gamma^{r-2})(\rho)$  can only be contracted into a  $(\xi^{r-1})$ . A counterexample to (39), (30) and (41) for  $r = 2$  will be given in 8.9 below.

**5.17.** *If  $\{z\} [a, \alpha^1, \dots, \alpha^r]$  is defined, then  $E(z, a, \alpha^1, \dots, \alpha^r)$ .* For assume  $\{z\} [a, \alpha^1, \dots, \alpha^r]$  is defined. Then by 5.8 there exists an  $\eta$  which is locally correct throughout the tree for  $(z, a, \alpha^1, \dots, \alpha^r)$  based on  $\eta$ . Also by 5.14, for any  $\eta$ , the computation terminates along every branch below which  $\eta$  is locally correct; and so, with this  $\eta$ , along every branch.

**5.18.** *If  $E(z, a, \alpha^1, \dots, \alpha^r)$ , then  $\{z\} [a, \alpha^1, \dots, \alpha^r]$  is defined.* For assume  $E(z, a, \alpha^1, \dots, \alpha^r)$ , and pick such an  $\eta$  (cf. 5.16). Then, assuming  $\{z\} [a, \alpha^1, \dots, \alpha^r]$  is undefined, we reach a contradiction by constructing a branch  $\rho$  along which the computation does not terminate, as in 5.15 but using the  $\eta$  already picked here (everywhere locally correct) instead of constructing one by stages as there.

**5.19. XXVIII.** *For each  $r \geq 2$  there is a primitive recursive predicate  $M$ , and for each  $r > 2$  a primitive recursive predicate  $N$ , such that*

$$\begin{aligned}
 (40) \quad & \{z\} [a, \alpha^1, \dots, \alpha^r] \simeq w \\
 & \equiv (\beta^{r-1})(E\xi^{r-2})M(z, a, \alpha^1, \dots, \alpha^r, w, \beta^{r-1}, \xi^{r-2}) \quad (r \geq 2),
 \end{aligned}$$

$$\begin{aligned}
 (41) \quad & \{z\} [a, \alpha^1, \dots, \alpha^r] \simeq w \\
 & \equiv (E\beta^{r-1})(\xi^{r-2})N(z, a, \alpha^1, \dots, \alpha^r, w, \beta^{r-1}, \xi^{r-2}) \quad (r > 2).
 \end{aligned}$$

**Proof** of (40). Using (29) and (14),

$$\begin{aligned}
 (42) \quad & \{z\} [a, \alpha^1, \dots, \alpha^r] \simeq w \equiv (\beta^{r-1})(E\xi^{r-2})K(z, a, \alpha^1, \dots, \alpha^r, \beta^{r-1}, \xi^{r-2}) \\
 & \& \ (\beta^{r-1})(E\xi^{r-2})I(z, a, \alpha^1, \dots, \alpha^r, w, \beta^{r-1}, \xi^{r-2}).
 \end{aligned}$$

**5.20.** To get results for  $r = 0, 1$  similar to those above for  $r \geq 2$ , and analogous to known theorems in the former theory of partial and general recursive functions for  $r = 0, 1$ , various known techniques are available (e.g. IM, p. 322 (D1) or (D2), or [11, §8] with [15, §18 and p. 424]). We elect here to adapt the foregoing treatment.

**5.21. XXIX.** *There are primitive recursive predicates  $T(s, z, a)$  and  $T'(z, a, y)$  such that*

$$(43) \quad (\{z\}[a, \alpha] \text{ is defined}) \equiv (Ey)T(\bar{\alpha}(y), z, a) \quad (r = 1 \text{ case}),$$

$$(44) \quad (\{z\}[a] \text{ is defined}) \equiv (Ey)T'(z, a, y) \quad (r = 0 \text{ case}).$$

**Proof,** for  $r = 1$ . Assume  $\{z\}[a, \alpha]$  defined. Then by 5.17,  $E(z, a, \alpha)$  (cf. 5.16). Since  $\text{tp}(z) \leq r < 2$ , S8 cannot be used (cf. I, 3.10); so in the tree for an  $\eta$  given by  $E(z, a, \alpha)$  the only branching will be at applications of S4 or the second equation of S5 (i.e. at nodes) with two  $n+1$ -positions issuing from the  $n$ -position. Hence by Brouwer's  $f_{\Delta n}$  theorem ([1, Theorem 2; 2, Theorem 2; 3], König [20]), there is an  $m > 0$  such that along each branch the computation terminates at an  $n$ -position with  $n < m$ . Now we confine our attention to what remains from the tree when all its branches are pruned off beginning with their  $m$ -positions; call it the  $m$ -tree. In the  $m$ -tree, the positions are described by numbers  $g = \bar{\rho}(n) < \prod_{i < m} p_i^2$ . Let  $y = \bar{\eta}(u)$  with  $u$  chosen so that  $u = \text{lh}(y) \geq \prod_{i < m} p_i^2$  (then  $y > u > m$ ). For operations that involve only values of  $\eta$  only at positions in the  $m$ -tree,  $\eta$  is replaceable by  $\lambda t (y)_t \div 1$ .

In particular, adapting 5.5 (with 5.4), we have

$$(45) \quad \{d \text{ is a position in the } m\text{-tree}\} \equiv \text{Seq}(d) \ \& \ \text{lh}(d) < m$$

$$\ \& \ (n)_{n < \text{lh}(d)} [(d)_n = 1 \vee \{[(z_n)_0 = 4 \vee ((z_n)_0 = 5 \ \& \ (a_n)_0 > 0)] \ \& \ (d)_n = 2\}]$$

where  $z_n = (\beta(n, z, a, d, \lambda t (y)_t \div 1))_0$ ,  $a_n = (\beta(n, z, a, d, \lambda t (y)_t \div 1))_1$ , which is of the form  $P(z, a, d, m, y)$  with a primitive recursive  $P$ .

To adapt 5.6 (with 5.4), let us further assume for the choice of  $u$  that  $y = \bar{\eta}(u) > a_n$  at each position  $g$  in the  $m$ -tree ( $n = \text{lh}(g)$ ). This condition is expressed, writing  $M = \prod_{i < m} p_i^2$ , by

$$(46) \quad (g)_{g < M} [P(z, a, g, m, y) \rightarrow y > (\beta(z, a, g, \lambda t (y)_t \div 1))_1],$$

which is of the form  $G(z, a, m, y)$  with a primitive recursive  $G$ . Also  $\alpha_n(v) = \beta_1(z, a, (\bar{\alpha}(y))_v \div 1, g, \lambda t (y)_t \div 1, v)$  for  $v < y$ . Now for  $g$  a position in the  $m$ -tree,

$$(47) \quad \{\eta \text{ is locally correct at } g, \text{ or } g \text{ is an } m-1\text{-position at which the computation is unterminated}\} \equiv C(z, a, \bar{\alpha}(y), g, y)$$

with a primitive recursive  $C$ .

The part of a branch  $\rho$  of the whole tree that belongs to the  $m$ -tree can be represented by its  $m-1$ -position  $d$ . Thus branches of the  $m$ -tree are represented by the numbers  $d$  such that  $P(z, a, d, m, y) \ \& \ \text{lh}(d) = m \div 1$ . Adapting 5.12, for such  $d$ , and  $n < m$ ,

$$(48) \quad \{\text{the computation terminates at the } n\text{-position on } d\} \equiv T(z, a, n, d, y)$$

with a primitive recursive  $T$ .

Combining these remarks,

$$\begin{aligned}
 (\{z\}[a, \alpha] \text{ is defined}) &\rightarrow (Ey)\{Ix(z) \ \& \ \text{tp}(z) \leq 1 \ \& \ (Em)_{0 < m < y}\{\text{Seq}(y) \\
 (49) \quad &\& \ \text{lh}(y) \geq \prod_{i < m} p_i^2 \ \& \ G(z, a, m, y) \ \& \ (g)_{0 < y}[P(z, a, g, m, y) \\
 &\rightarrow C(z, a, \bar{\alpha}(y), g, y)] \\
 &\& \ (d)_{d < y}[P(z, a, d, m, y) \ \& \ \text{lh}(d) = m \div 1 \rightarrow (En)_{n < m}T(z, a, n, d, y)]\} \}.
 \end{aligned}$$

Upon replacing  $y$  except in  $\bar{\alpha}(y)$  by  $\text{lh}(\bar{\alpha}(y))$ , the scope of  $(Ey)$  in (49) assumes the form  $T(\bar{\alpha}(y), z, a)$  with a primitive recursive  $T$ . Thus

$$(50) \quad (\{z\}[a, \alpha] \text{ is defined}) \rightarrow (Ey)T(\bar{\alpha}(y), z, a).$$

Conversely, assume  $(Ey)T(\bar{\alpha}(y), z, a)$ , and pick such a  $y$ . Then  $y$  is of the form  $\bar{\eta}(u)$  for some function  $\eta$ , with  $u = \text{lh}(y) \geq \prod_{i < m} p_i^2$  for some  $m > 0$ , and  $G(z, a, m, y)$ . The rest of  $T(\bar{\alpha}(y), z, a)$  then gives everything stated in  $E(z, a, \alpha)$  for the tree based on this  $\eta$ , except the local correctness at  $n$ -positions for  $n \geq m$ , since the computation terminates along every branch at an  $n$ -position with  $n < m$ . We can by altering values of  $\eta$  at most at  $n$ -positions with  $n \geq m$  obtain the local correctness there also (without spoiling it for  $n < m$ , or altering the  $m$ -tree or the termination). Then 5.18 applies. Thus

$$(51) \quad (Ey)T(\bar{\alpha}(y), z, a) \rightarrow (\{z\}[a, \alpha] \text{ is defined}).$$

5.22. XXX. *There is a primitive recursive function  $U(y)$  such that*

$$(52) \quad \{z\}[a, \alpha] \simeq U(\mu y T(\bar{\alpha}(y), z, a)),$$

$$(53) \quad T(\bar{\alpha}(y), z, a) \rightarrow U(y) = \{z\}[a, \alpha] \quad (r = 1 \text{ case}),$$

$$(54) \quad \{z\}[a] \simeq U(\mu y T'(z, a, y)),$$

$$(55) \quad T'(z, a, y) \rightarrow U(y) = \{z\}[a] \quad (r = 0 \text{ case}).$$

**Proof,** for  $r = 1$ . By (43) it will suffice for (52) to pick  $U$  to satisfy (53). So assume  $T(\bar{\alpha}(y), z, a)$  (then  $\{z\}[a, \alpha]$  is defined). By the proof of (51),  $y = \bar{\eta}(u)$  for some  $\eta$ ,  $u$  and  $m$  with  $u > m > 0$ ; and, by altering values of  $\eta$  at most for some arguments  $\bar{\rho}(n)$  with  $n \geq m$ ,  $\eta$  becomes locally correct everywhere in the tree for  $(z, a, \alpha)$  based on it. So by 5.7,  $\{z\}[a, \alpha] = \eta(1) = (\bar{\eta}(u))_1 \div 1 = (y)_1 \div 1$  (since  $\eta(1)$  is unaltered). So take  $U(y) = (y)_1 \div 1$ .

5.23. XXXI. *Each function  $\phi(a)$  of variables  $a$  of types  $\leq 1$  partial (general) recursive in the present sense is partial (general) recursive in the former sense (e.g. IM, Chapters XI, XII).*

**Proof.** By (52) or (54) with (13) and IM, Theorem XVIII, p. 330.

5.24. For each  $n \geq 1$ , putting  $\alpha = \langle \alpha_1, \dots, \alpha_n \rangle$ ,

$$(56) \quad \bar{\alpha}(x) = \prod_{i < \text{lh}(\bar{\alpha}_1(x))} p_i \exp 1 + \langle (\bar{\alpha}_1(x))_i \div 1, \dots, (\bar{\alpha}_n(x))_i \div 1 \rangle.$$

This enables us by substitution into XXX to obtain the usual forms of the normal form theorem (IM, p. 292, [14, Footnote 2]) but now with indices.

For example, let

$$T_2^{1,1}(u, v, z, a, b) = T \left( \prod_{i < \text{lh}(u)} p_i \exp 1 + \langle (u)_i \div 1, (v)_i \div 1 \rangle, z, \langle a, b \rangle \right).$$

XXXII. If  $\phi(a, b, \alpha, \beta)$  is partial recursive with index  $z$ ,

$$(57) \quad \phi(a, b, \alpha, \beta) \simeq U(\mu y T_2^{1,1}(\bar{\alpha}(y), \bar{\beta}(y), z, a, b)),$$

$$(58) \quad T_2^{1,1}(\bar{\alpha}(y), \bar{\beta}(y), z, a, b) \rightarrow U(y) = \phi(a, b, \alpha, \beta).$$

5.25. To get the enumeration theorem under type-0 quantification (IM, Theorems IV, IV\*, pp. 281, 292) but now with indices, take e.g. a general recursive predicate  $R(a, b, \alpha, \beta, y)$ ; let its representing function be  $\chi$  with index  $h$ , so

$$R(a, b, \alpha, \beta, y) \equiv \chi(a, b, \alpha, \beta, y) = 0 \equiv \{h\}(a, b, \alpha, \beta, y) = 0.$$

By S9 and XVI,  $\lambda z a b \alpha \beta \mu y [\{z\}(a, b, \alpha, \beta, y) = 0]$  is a partial recursive function, say with index  $e$ . By XIII, the partial recursive function  $\lambda a b \alpha \beta \mu y [\{h\}(a, b, \alpha, \beta, y) = 0]$ , i.e.  $\lambda a b \alpha \beta \mu y R(a, b, \alpha, \beta, y)$ , has the index  $f = S^1(e, h)$ . In the following equivalence (59), the left side expresses the condition of definition of this function for particular  $a, b, \alpha, \beta$  as given directly by its definition by the  $\mu$ -operator, and the right the condition as given by (57) (or (43) via (13) and (56)) from  $f$  being an index of it.

XXXIII. To each general recursive predicate  $R(a, b, \alpha, \beta, y)$ , there are numbers  $f, g$  such that

$$(59) \quad (E y) R(a, b, \alpha, \beta, y) \equiv (E y) T_2^{1,1}(\bar{\alpha}(y), \bar{\beta}(y), f, a, b),$$

$$(60) \quad (y) R(a, b, \alpha, \beta, y) \equiv (y) \bar{T}_2^{1,1}(\bar{\alpha}(y), \bar{\beta}(y), g, a, b).$$

5.26. Other results for the present  $T$  predicates with indices follow from the foregoing in the same manner as in IM for the  $T$  predicates with Gödel numbers.

6. Reduction in type of a quantifier. 6.1. The following theorem and its dual correspond for  $r \geq 2$  to [14, (7) and (8), p. 316] for  $r = 1$ . (Cf. XLII and XLIII below.)

XXXIV. Suppose  $r \geq 2$ . Let  $a^r$  be variables of types  $\leq r$ ,  $b^{r-2}$  variables of types  $\leq r-2$ ,  $(Qb^{r-2})$  quantifiers on the variables  $b^{r-2}$ , and  $P(a^r, \sigma^r, b^{r-2})$  a general recursive predicate. Then there is a primitive recursive predicate  $R(a^r, \eta^{r-1}, \xi^{r-2})$  such that

$$(61) \quad (E \sigma^r)(Q b^{r-2}) P(a^r, \sigma^r, b^{r-2}) \equiv (E \eta^{r-1})(\xi^{r-2}) R(a^r, \eta^{r-1}, \xi^{r-2}).$$

Similarly if  $P$  is merely partial recursive, with (61) holding for those values of  $a^r$  such that  $P(a^r, \sigma^r, b^{r-2})$  is defined for all  $\sigma^r, b^{r-2}$ .

The proof occupies 6.2-6.9.

6.2. Assume for  $P$  merely partial recursive that the values of  $\mathfrak{a}^r$  under consideration make  $\lambda\sigma^r \mathfrak{b}^{r-2} P(\mathfrak{a}^r, \sigma^r, \mathfrak{b}^{r-2})$  completely defined. Say  $z_0$  is an index of the representing function of  $P$ . Then  $P(\mathfrak{a}^r, \sigma^r, \mathfrak{b}^{r-2}) \equiv \{z_0\}(\mathfrak{a}^r, \sigma^r, \mathfrak{b}^{r-2}) = 0$ . Using this in the left side of (61), applying (13) and (15), advancing the quantifier ( $E\beta^{r-1}$ ) and contracting adjacent quantifiers of like kind by (4), (3), ( $\bar{3}$ ) with (8), ( $\bar{8}$ ), writing  $\mathfrak{a}^r = (c^{r-2}, \alpha_1^{r-1}, \dots, \alpha_{m_r-1}^{r-1}, \alpha_1^r, \dots, \alpha_{m_r-1}^r)$  where  $c^{r-2}$  are of types  $\leq r-2$ ,  $\alpha_0^{r-1} = \langle \alpha_1^{r-1}, \dots, \alpha_{m_r-1}^{r-1} \rangle$  and  $\alpha_0^r = \langle \alpha_1^r, \dots, \alpha_{m_r-1}^r \rangle$ , and using (2), we obtain

$$(62) \quad (E\sigma^r)(Q\mathfrak{b}^{r-2})P(\mathfrak{a}^r, \sigma^r, \mathfrak{b}^{r-2}) \equiv (E\sigma^r)(\xi^{r-2})P(c^{r-2}, \alpha_0^{r-1}, \alpha_0^r, \sigma^r, \xi^{r-2})$$

with a primitive recursive  $P$  on the right. Thus the general case of XXXIV is reduced to the case of it for a primitive recursive predicate  $P(c^{r-2}, \alpha^{r-1}, \alpha^r, \sigma^r, \xi^{r-2})$  of variables of the types shown.

6.3. Let  $z_1$  be a primitive recursive index (cf. 4.1) of the representing function of such a  $P(c^{r-2}, \alpha^{r-1}, \alpha^r, \sigma^r, \xi^{r-2})$ . Then

$$(63) \quad P(c^{r-2}, \alpha^{r-1}, \alpha^r, \sigma^r, \xi^{r-2}) \equiv \{z_1\}(c^{r-2}, \xi^{r-2}, \alpha^{r-1}, \alpha^r, \sigma^r) = 0.$$

By I, in computing  $\{z_1\}(c^{r-2}, \xi^{r-2}, \alpha^{r-1}, \alpha^r, \sigma^r)$ , applications of S8.r with  $\sigma^r$  as the  $\alpha^i$  of S8.r will always be to introduce  $\sigma^r(\lambda\tau^{r-2}\{h\}(\tau^{r-2}, \mathfrak{d}^{r-2}, \alpha^{r-1}, \sigma^r, \alpha^r))$  with  $\mathfrak{d}^{r-2}$  of types  $\leq r-2$ ; i.e. the  $\alpha^i(\lambda\alpha^{j-2}\chi(\alpha^i, \alpha^{j-2}, \mathfrak{b}))$  will always be of this form with only  $(h, \mathfrak{d}^{r-2})$  varying from one application to the next. The idea of the proof of XXXIV is that hence we can in the computation replace the type- $r$  function  $\sigma^r$  by a type- $r-1$  function  $\eta^{r-1}$  chosen so that (cf. 2.4)  $\eta^{r-1}(\langle h, \mathfrak{e} \rangle) = \sigma^r(\lambda\tau^{r-2}\{h\}(\tau^{r-2}, \mathfrak{d}, \alpha^{r-1}, \sigma^r, \alpha^r))$  when  $\mathfrak{e} = (\epsilon^0, \dots, \epsilon^{r-2})$  and  $\mathfrak{d} = \mathfrak{e}^{(h)} = ((\epsilon^0)_0, \dots, (\epsilon^0)_{n_0-1}, \dots, (\epsilon^{r-2})_0, \dots, (\epsilon^{r-2})_{n_{r-2}-1})$  for  $n_j = (h)_{1,j} \div \text{sg}(r-2-j)$ . In particular, if  $\mathfrak{d} = (\delta_1^0, \dots, \delta_{n_0}^0, \dots, \delta_1^{r-2}, \dots, \delta_{n_{r-2}}^{r-2})$  for such  $n_j$ , then  $\mathfrak{d}^* = (\delta^0, \dots, \delta^{r-2})$  for  $\delta^i = \langle \delta_1^i, \dots, \delta_{n_i}^i \rangle$  is such an  $\mathfrak{e}$ , i.e.  $\mathfrak{d}^{*(h)} = \mathfrak{d}$ .

6.4. To carry out this idea, we introduce two primitive recursive functions  $\zeta_0(z)$  and  $\zeta_1(z)$  which will give indices of  $\{z\}(\mathfrak{d}, \alpha^{r-1}, \alpha^r, \sigma^r)$  and  $\{z\}(\mathfrak{d}, \alpha^{r-1}, \sigma^r, \alpha^r)$ , respectively, as functions of  $\mathfrak{d}, \alpha^{r-1}, \eta^{r-1}, \alpha^r$  (cf. 6.5).

The definitions of  $\zeta_0$  and  $\zeta_1$  are by simultaneous course-of-values recursion. Let  $\zeta_i(z) = 0$  ( $i = 0, 1$ ), except for  $\text{PRI}(z) \ \& \ \text{tp}(z) = r \ \& \ (z)_{1,r-1} = 1 \ \& \ (z)_{1,r} = 2$ , when the appropriate following case shall apply.

CASE 1.  $(z)_0 = 1$ . Let  $\zeta_i(z) = \langle 1, p_{r-1} \cdot [(z)_1/p_r] \rangle$ , so e.g., when  $z$  is an index by S1 of  $\lambda \mathfrak{a} \mathfrak{b} \lambda \alpha^{r-1} \alpha^r \sigma^r \mathfrak{a} + 1$ , then  $\zeta_0(z)$  is one of  $\lambda \mathfrak{a} \mathfrak{b} \lambda \alpha^{r-1} \eta^{r-1} \alpha^r \mathfrak{a} + 1$ .

CASE 4.  $(z)_0 = 4$ . Let  $\zeta_i(z) = \langle 4, p_{r-1} \cdot [(z)_1/p_r], \zeta_i((z)_2), \zeta_i((z)_3) \rangle$ .

CASE 6.r.  $(z)_0 = 6 \ \& \ (z)_2 = r$ . Let  $\zeta_0(z) = \zeta_1((z)_4)$  and  $\zeta_1(z) = \zeta_0((z)_4)$ , if  $(z)_3 = 1$ ; otherwise,  $\zeta_i(z) = \zeta_i((z)_4)$ .

CASE 8.r.  $(z)_0 = 8 \ \& \ (z)_2 = r$ . Let  $\zeta_0(z) = \langle 8, p_{r-1} \cdot [(z)_1/p_r], r, \zeta_0((z)_3) \rangle$ . We want  $\zeta_1(z)$  to be a primitive recursive index  $\theta(z)$  of  $\lambda \mathfrak{b} \lambda \alpha^{r-1} \eta^{r-1} \alpha^r \eta^{r-1}(\langle (z)_3, \mathfrak{d}^* \rangle)$ , where  $\mathfrak{d}$  is a list of variables such that  $z$  is an index for  $\mathfrak{d}, \alpha^{r-1}, \sigma^r, \alpha^r$ , and  $\theta$  is primitive recursive. To construct this index, let  $e$  be a primitive recursive

index of  $\lambda\epsilon^0 \cdots \epsilon^{r-2} \alpha^{r-1} \eta^{r-1} \alpha^r \eta^{r-1} (\langle c, \epsilon^0, \dots, \epsilon^{r-2} \rangle)$ . Suppose e.g.  $r > 2$ , and write  $n_j = (z)_{1,j}$ . We construct further indices as follows, using XIII, XXI, XXV, XXII.

$$\lambda \epsilon^0 \cdots \epsilon^{r-2} \delta_1^{r-2} \cdots \delta_{n_{r-2}}^{r-2} \alpha^{r-1} \eta^{r-1} \alpha^r \eta^{r-1} (\langle (z)_3, \epsilon^0, \dots, \epsilon^{r-2} \rangle):$$

$$A = \iota(S^1(e, (z)_3), p_{r-2}^{n_{r-2}}).$$

$$\lambda \epsilon^0 \cdots \epsilon^{r-3} \delta_1^{r-2} \cdots \delta_{n_{r-2}}^{r-2} \alpha^{r-1} \eta^{r-1} \alpha^r \tau \langle \delta_1 (\tau^{r-3}), \dots, \delta_{n_{r-2}} (\tau^{r-3}) \rangle:$$

$$C = \langle 6, (B)_1, r-3, 1, B \rangle \text{ where } B = \iota(\nu_{r-2}(n_{r-2}), p_0 \cdots p_{r-3} \cdot p_{r-1}^2 \cdot p_r).$$

$$\lambda \epsilon^0 \cdots \epsilon^{r-3} \delta_1^{r-2} \cdots \delta_{n_{r-2}}^{r-2} \alpha^{r-1} \eta^{r-1} \alpha^r \eta^{r-1} (\langle (z)_3, \epsilon^0, \dots, \epsilon^{r-3}, \delta^{r-2} \rangle):$$

$$D = \gamma_{r-2}(A, C, 0).$$

Here  $\delta^{r-2} = \langle \delta_1^{r-2}, \dots, \delta_{n_{r-2}}^{r-2} \rangle$  (end 6.3). Continuing in this manner to substitute successively  $\delta^{r-3}, \dots, \delta^1$  for  $\epsilon^{r-3}, \dots, \epsilon^1$ , and, using S4,  $\delta^0$  for  $\epsilon^0$ , we obtain an index  $\theta(z)$  of  $\lambda \delta \alpha^{r-1} \eta^{r-1} \alpha^r \eta^{r-1} (\langle (z)_3, \delta^* \rangle)$  with  $\theta$  primitive recursive. But  $\theta(z)$  is a primitive recursive index, since  $e, \nu_{r-2}(n_{r-2})$  are primitive recursive indices, and the operations used produce primitive recursive indices when applied to primitive recursive indices.

6.5. Choose any  $\alpha^{r-1}, \sigma^r, \alpha^r$ , and let  $\eta^{r-1}$  be defined thence by

$$(A) \quad \eta^{r-1}(\gamma^{r-2}) = \begin{cases} \sigma^r(\lambda\tau \{h\}(\tau, e^{(h)}, \alpha^{r-1}, \sigma^r, \alpha^r)) & \text{if } \gamma^{r-2} = \langle h, e \rangle \text{ with} \\ h \text{ a primitive recursive index for } (\tau, e^{(h)}, \alpha^{r-1}, \sigma^r, \alpha^r), & \\ 0 & \text{otherwise.} \end{cases}$$

If  $\text{PRI}(z) \ \& \ \text{tp}(z) = r \ \& \ (z)_{1,r-1} = 1 \ \& \ (z)_{1,r} = 2$ , then

$$(64) \quad \{\zeta_0(z)\}(\mathfrak{b}, \alpha^{r-1}, \eta^{r-1}, \alpha^r) = \{z\}(\mathfrak{b}, \alpha^{r-1}, \alpha^r, \sigma^r),$$

$$(65) \quad \{\zeta_1(z)\}(\mathfrak{b}, \alpha^{r-1}, \eta^{r-1}, \alpha^r) = \{z\}(\mathfrak{b}, \alpha^{r-1}, \sigma^r, \alpha^r)$$

when  $\mathfrak{b}$  are variables such that  $z$  is an index for  $(\mathfrak{b}, \alpha^{r-1}, \sigma^r, \alpha^r)$ .

**Proof**, by induction on  $z$ .

CASE 6.r for  $\zeta_0(z)$  with  $(z)_3 = 1$ . By the theorem and case hypotheses,  $\{z\}(\mathfrak{b}, \alpha^{r-1}, \alpha^r, \sigma^r) = \{(z)_4\}(\mathfrak{b}, \alpha^{r-1}, \sigma^r, \alpha^r)$  where  $\text{PRI}((z)_4) \ \& \ \text{tp}((z)_4) = r \ \& \ ((z)_4)_{1,r-1} = 1 \ \& \ ((z)_4)_{1,r} = 2$ . By hyp. ind.,  $\{\zeta_1((z)_4)\}(\mathfrak{b}, \alpha^{r-1}, \eta^{r-1}, \alpha^r) = \{(z)_4\}(\mathfrak{b}, \alpha^{r-1}, \sigma^r, \alpha^r)$ . Thence (64) follows by the definition  $\zeta_0(z) = \zeta_1((z)_4)$ .

CASE 8.r for  $\zeta_1(z)$ . We have  $\{z\}(\mathfrak{b}, \alpha^{r-1}, \sigma^r, \alpha^r) = \sigma^r(\lambda\tau \{(z)_3\}(\tau, \mathfrak{b}, \alpha^{r-1}, \sigma^r, \alpha^r))$  with  $(z)_3$  a primitive recursive index for  $(\tau, \mathfrak{b}, \alpha^{r-1}, \sigma^r, \alpha^r)$ . So by (A),  $\eta^{r-1}(\langle (z)_3, \delta^* \rangle) = \{z\}(\mathfrak{b}, \alpha^{r-1}, \sigma^r, \alpha^r)$ ; and by definition,  $\{\zeta_1(z)\}(\mathfrak{b}, \alpha^{r-1}, \eta^{r-1}, \alpha^r) = \eta^{r-1}(\langle (z)_3, \delta^* \rangle)$ .

6.6. Now we formulate a property  $F(\alpha^{r-1}, \eta^{r-1}, \alpha^r)$  of a type- $r-1$  variable  $\eta^{r-1}$  which, when (A) holds (cf. 6.7), expresses that  $\eta^{r-1}$  takes the same value for any two arguments  $\gamma_0^{r-2} = \langle h_0, e_0 \rangle$  and  $\gamma_1^{r-2} = \langle h_1, e_1 \rangle$  which represent the same function of  $\tau$ , i.e. such that  $\lambda\tau \{h_0\}(\tau, e_0^{(h_0)}, \alpha^{r-1}, \sigma^r, \alpha^r) = \lambda\tau \{h_1\}$

$(\tau, e_1^{(h_1)}, \alpha^{r-1}, \sigma^r, \alpha^r)$ . However we use (65) to state it without using  $\sigma^r$ . Thus let

$$\begin{aligned} F(\alpha^{r-1}, \eta^{r-1}, \alpha^r) \equiv & (h_0)(e_0)(h_1)(e_1) \{ (i)_{i < 2} [\text{PRI}(h_i) \ \& \ \text{tp}(h_i) = r \\ & \& \ (h_i)_{1, r-2} > 0 \ \& \ (h_i)_{1, r-1} = 1 \ \& \ (h_i)_{1, r} = 2] \\ & \& \ (\tau^{r-2})(Ew)(i)_{i < 2} [\{ \zeta_1(h_i) \} (\tau^{r-2}, e_i^{(h_i)}, \alpha^{r-1}, \eta^{r-1}, \alpha^r) = w] \\ \rightarrow & \eta^{r-1}(\langle h_0, e_0 \rangle) = \eta^{r-1}(\langle h_1, e_1 \rangle) \}. \end{aligned}$$

But (omitting  $\sigma^{r-3}$  if  $r=2$ ) for  $h_i$  an index for  $(\tau, e^{(h_i)}, \alpha^{r-1}, \sigma^r, \alpha^r)$ ,

$$(66) \quad \begin{aligned} \{ \zeta_1(h_i) \} (\tau^{r-2}, e_i^{(h_i)}, \alpha^{r-1}, \eta^{r-1}, \alpha^r) &= \{ \zeta_1(h_i) \} [\epsilon_i^0, \dots, \epsilon_i^{r-3}, \\ & \lambda \sigma^{r-3} 2^{r-2(\sigma^{r-3})} \cdot \prod_{s < n_{r-2, i}} p_{s+1} \exp(\epsilon_i^{r-2}(\sigma^{r-3}))_s, \langle \alpha^{r-1}, \eta^{r-1} \rangle, \langle \alpha^r \rangle]. \end{aligned}$$

Using (66) and (14) in the expression for  $F(\alpha^{r-1}, \eta^{r-1}, \alpha^r)$ , and advancing and contracting quantifiers, we obtain a primitive recursive  $G$  such that

$$(67) \quad F(\alpha^{r-1}, \eta^{r-1}, \alpha^r) \equiv (E\beta^{r-1})(\xi^{r-2})G(\alpha^{r-1}, \eta^{r-1}, \alpha^r, \beta^{r-1}, \xi^{r-2}).$$

**6.7.** For any given  $\alpha^{r-1}, \sigma^r, \alpha^r$ , if  $\eta^{r-1}$  is defined by (A), then  $F(\alpha^{r-1}, \eta^{r-1}, \alpha^r)$ .

For suppose the antecedent of the implication in  $F(\alpha^{r-1}, \eta^{r-1}, \alpha^r)$  is satisfied by given  $h_0, e_0, h_1, e_1$ . By the first part of this antecedent with (65), then  $\{ \zeta_1(h_i) \} (\tau, e_i^{(h_i)}, \alpha^{r-1}, \eta^{r-1}, \alpha^r) = \{ h_i \} (\tau, e_i^{(h_i)}, \alpha^{r-1}, \sigma^r, \alpha^r)$ . This with the second part gives  $\lambda \tau \{ h_0 \} (\tau, e_0^{(h_0)}, \alpha^{r-1}, \sigma^r, \alpha^r) = \lambda \tau \{ h_1 \} (\tau, e_1^{(h_1)}, \alpha^{r-1}, \sigma^r, \alpha^r)$ . Hence by (A),  $\eta^{r-1}(\langle h_0, e_0 \rangle) = \eta^{r-1}(\langle h_1, e_1 \rangle)$ .

**6.8.** Choose any  $\alpha^{r-1}, \eta^{r-1}, \alpha^r$  such that  $F(\alpha^{r-1}, \eta^{r-1}, \alpha^r)$ , and let  $\sigma^r$  be defined thence by

$$(B) \quad \sigma^r(\gamma^{r-1}) = \begin{cases} \eta^{r-1}(\langle h, \mathfrak{d}^* \rangle) \text{ if } \gamma^{r-1} = \lambda \tau \{ \zeta_1(h) \} (\tau, \mathfrak{d}, \alpha^{r-1}, \eta^{r-1}, \alpha^r) \text{ with} \\ h \text{ a primitive recursive index for } (\tau, \mathfrak{d}, \alpha^{r-1}, \sigma^r, \alpha^r), \\ 0 \text{ otherwise.} \end{cases}$$

If  $\text{PRI}(z) \ \& \ \text{tp}(z) = r \ \& \ (z)_{1, r-1} = 1 \ \& \ (z)_{1, r} = 2$ , then (64) and (65) hold for variables  $\mathfrak{d}$  such that  $z$  is an index for  $(\mathfrak{d}, \alpha^{r-1}, \sigma^r, \alpha^r)$ . (By  $F(\alpha^{r-1}, \eta^{r-1}, \alpha^r)$ , the value of  $\sigma^r(\gamma^{r-1})$  in the first case under (B) is independent of the choice of the  $h, \mathfrak{d}$ .)

**Proof.** As before (6.5), except for one case.

CASE 8.r for  $\zeta_1(z)$ . We have  $\{ z \} (\mathfrak{d}, \alpha^{r-1}, \sigma^r, \alpha^r) = \sigma^r(\lambda \tau \{ (z)_3 \} (\tau, \mathfrak{d}, \alpha^{r-1}, \sigma^r, \alpha^r))$  with  $(z)_3$  a primitive recursive index for  $(\tau, \mathfrak{d}, \alpha^{r-1}, \sigma^r, \alpha^r)$ . By hyp. ind.,  $(\tau) [\{ \zeta_1((z)_3) \} (\tau, \mathfrak{d}, \alpha^{r-1}, \eta^{r-1}, \alpha^r) = \{ (z)_3 \} (\tau, \mathfrak{d}, \alpha^{r-1}, \sigma^r, \alpha^r)]$ . So by (B),  $\sigma^r(\lambda \tau \{ (z)_3 \} (\tau, \mathfrak{d}, \alpha^{r-1}, \sigma^r, \alpha^r)) = \eta^{r-1}(\langle (z)_3, \mathfrak{d}^* \rangle)$ ; and by definition (6.4),  $\{ \zeta_1(z) \} (\mathfrak{d}, \alpha^{r-1}, \eta^{r-1}, \alpha^r) = \eta^{r-1}(\langle (z)_3, \mathfrak{d}^* \rangle)$ .

**6.9.** Combining 6.7 and 6.5,

$$(68) \quad \begin{aligned} (\alpha^{r-1})(\sigma^r)(\alpha^r)(E\eta^{r-1}) \{ F(\alpha^{r-1}, \eta^{r-1}, \alpha^r) \ \& \ (z) \{ \text{PRI}(z) \ \& \ \text{tp}(z) = r \ \& \\ (z)_{1, r-1} = 1 \ \& \ (z)_{1, r} = 2 \rightarrow (\mathfrak{d}) [\{ \zeta_0(z) \} (\mathfrak{d}, \alpha^{r-1}, \eta^{r-1}, \alpha^r) = \{ z \} (\mathfrak{d}, \alpha^{r-1}, \alpha^r, \sigma^r)] \} \} \end{aligned}$$

where, for each such  $z$ , the  $\mathfrak{b}$  are variables such that  $z$  is an index for  $(\mathfrak{b}, \alpha^{r-1}, \sigma^r, \alpha^r)$ . By 6.8,

$$(69) \quad (\alpha^{r-1})(\eta^{r-1})(\alpha^r) \{ F(\alpha^{r-1}, \eta^{r-1}, \alpha^r) \rightarrow (E\sigma^r)(z) \{ \text{PRI}(z) \ \& \ \text{tp}(z) = r \ \& \ (z)_{1,r-1} = 1 \ \& \ (z)_{1,r} = 2 \rightarrow (\mathfrak{b}) [ \{ \zeta_0(z) \} (\mathfrak{b}, \alpha^{r-1}, \eta^{r-1}, \alpha^r) = \{ z \} (\mathfrak{b}, \alpha^{r-1}, \alpha^r, \sigma^r) ] \} \}.$$

Applying (68) and (69) (cf. 6.3),

$$(70) \quad (E\sigma^r)(\xi^{r-2}) [ \{ z_1 \} (c^{r-2}, \xi^{r-2}, \alpha^{r-1}, \alpha^r, \sigma^r) = 0 ] \\ \equiv (E\eta^{r-1}) \{ F(\alpha^{r-1}, \eta^{r-1}, \alpha^r) \ \& \ (\xi^{r-2}) [ \{ \zeta_0(z_1) \} (c^{r-2}, \xi^{r-2}, \alpha^{r-1}, \eta^{r-1}, \alpha^r) = 0 ] \}.$$

Thence, using (63) and (67), and advancing and contracting quantifiers,

$$(71) \quad (E\sigma^r)(\xi^{r-2}) P(c^{r-2}, \alpha^{r-1}, \alpha^r, \sigma^r, \xi^{r-2}) \\ \equiv (E\eta^{r-1})(\xi^{r-2}) R(c^{r-2}, \alpha^{r-1}, \alpha^r, \eta^{r-1}, \xi^{r-2})$$

with a primitive recursive  $R$ .

**7. Predicates of order  $r$ .** 7.1. Consider a predicate  $P(\mathfrak{a})$ , where  $\mathfrak{a}$  is a list of variables of our types 0, 1, 2,  $\dots$  (1.2). We call  $P(\mathfrak{a})$  *r-expressible* in certain predicates and functions (the *primitives*), if there is a syntactically-constructed expression (an *r-expression*) for  $P(\mathfrak{a})$  in terms of variables of our types, the primitives (only applied to arguments), and the symbols of the predicate calculus with quantification only of variables of types  $< r$ . If  $P$  is *r-expressible* in predicates general recursive in (completely defined) functions  $\Psi$ , we say  $P$  is of *order  $r$*  in  $\Psi$ . The notion  $\Psi$  extends to the case  $P$  is replaced by a function  $\phi$  via the representing predicate of  $\phi$  (IM, p. 199), and to the case any of  $\Psi$  are replaced by predicates via their representing functions (IM, p. 227), and for  $\Psi$  varying is *uniform* if the same *r-expression* can serve for all values of  $\Psi$  and the recursiveness in  $\Psi$  of its primitives is uniform.

7.2. As in [14, 2.2], the class of the primitives for predicates of order  $r$  in  $\Psi$  can be enlarged to include the functions general recursive in  $\Psi$ .

7.3. Clearly,  $P$  is general recursive in  $\Psi$  exactly if  $P$  is of order 0 in  $\Psi$ . We say  $P$  is *arithmetical (analytic)* in  $\Psi$ , if  $P$  is of order 1 (2) in  $\Psi$ . By XVII and XXXI (and, for  $\Psi$  nonempty, 3.15), this agrees with previous usage in the case then considered that the variables of  $P$  are of types  $\leq 1$  and of  $\Psi$  of type 0 (cf. IM, pp. 239, 284–285, 291–292, [14, pp. 313–314]).

In that case, a smaller class of primitives than the predicates and functions general recursive in  $\Psi$  suffices. Indeed, for predicates arithmetical in  $\Psi$ , by IM, Theorem VII\* (b), pp. 285, 292 the primitives  $=, +, \cdot, \Psi$  suffice<sup>(13)</sup>. For predicates analytic in  $\Psi$ , by [14, 2.3 and Footnote 6] the primitives  $=, +1, \Psi$  suffice; this result is included in the  $r = 1$  case of XXXVI below.

<sup>(13)</sup> The constant natural numbers are expressible in 0, 1, + (or 0, +1); then 0, 1 are replaceable using number quantification by  $w=0, w=1$  (e.g. IM, p. 411 with  $n=0$ ); and the latter are 1-expressible in  $=, +$  (or  $+1$ ),  $\cdot$ , thus:  $w=0 \equiv (x)[x+w=x] \equiv (x)[x+1 \neq w], w=1 \equiv (x)[x \cdot w = x]$ .

**7.4. XXXV.** For each  $r \geq 1$ : If a function  $\phi(a)$  of variables  $a$  of types  $\leq r+1$  is general recursive in functions  $\Psi$  of variables of types  $\leq r$ , then  $\phi(a) = w$  is  $r+1$ -expressible in  $=, +, \cdot, \Psi$ , with a prenex  $r+1$ -expression in which all the type- $r$  quantifiers are universal, and also one in which all are existential.

**Proof.** PART (a).  $\phi(a)$  is primitive recursive in  $\Psi$ . We use induction on the length of a primitive recursive derivation of  $\phi(a)$  from  $\Psi$  by our schemata, as for IM, Theorem I, I\*, pp. 241, 292<sup>(13)</sup>. Cases 1-7 are essentially as before, using in Case 5 ideas of Dedekind and Gödel, and introducing only number quantifiers (type 0). Using in the applications of the hyp. ind. prenex forms with their type- $r$  quantifiers only universal (existential), the resulting prenex forms have the same property.

CASE 8.  $\phi(\alpha^j, b) = \alpha^j(\lambda\alpha^{j-2}\chi(\alpha^j, \alpha^{j-2}, b))$  where  $\chi$  comes earlier in the derivation. Now

$$(72) \quad \begin{aligned} \phi(\alpha^j, b) = w &\equiv (\gamma^{j-1})\{(\alpha^{j-2})[\chi(\alpha^j, \alpha^{j-2}, b) = \gamma^{j-1}(\alpha^{j-2})] \rightarrow \alpha^j(\gamma^{j-1}) = w\} \\ &\equiv (E\gamma^{j-1})\{(\alpha^{j-2})[\chi(\alpha^j, \alpha^{j-2}, b) = \gamma^{j-1}(\alpha^{j-2})] \& \alpha^j(\gamma^{j-1}) = w\}. \end{aligned}$$

The quantifier  $(\gamma^{j-1})$  or  $(E\gamma^{j-1})$  is of type  $\leq r$ , since  $j \leq r+1$ . By hyp. ind.,  $\chi(\alpha^j, \alpha^{j-2}, b) = w$ , and thence by substitution  $\chi(\alpha^j, \alpha^{j-2}, b) = \gamma^{j-1}(\alpha^{j-2})$ , is  $r+1$ -expressible in  $=, +, \cdot, \Psi$ . For  $j-1 = r$ , use of a prenex form for  $\chi(\alpha^j, \alpha^{j-2}, b) = w$  with its type- $r$  quantifiers only existential leads from the two forms of (72) to prenex forms of  $\phi(\alpha^j, b) = w$  with their type- $r$  quantifiers all universal and all existential, respectively. CASE 0 (cf. 1.8) is similar, since each variable  $\gamma^{j-1}$  of each of  $\Psi$  is of type  $\leq r$ .

PART (b). Otherwise. For  $\Psi$  nonempty,  $\phi(a) = w$  can be transformed by 3.15, 2.7, 1.9 into  $\phi(a, \Psi') = w$ , where  $\phi(a, b)$  is partial recursive absolutely, and  $\Psi'$  are functions primitive recursive in  $\Psi$ . Using (13) to express  $\phi(a, b) \simeq w$  in the form  $\{z\}[a, \alpha^1, \dots, \alpha^{r+1}] \simeq w$ , applying XXVI to obtain expressions equivalent to the latter when  $\phi(a, b)$  is defined, and substituting  $\Psi'$  for  $b$  by IV\* (1.8, 1.5), we obtain  $\phi(a) = w \equiv (\beta^r)(E\xi^{r-1})R(a, w, \beta^r, \xi^{r-1}) \equiv (E\beta^r)(\xi^{r-1})S(a, w, \beta^r, \xi^{r-1})$  with  $R, S$  primitive recursive in  $\Psi$ . The conclusion follows by applying the result of Part (a) to the representing functions of  $R, S$  as its  $\phi$ .

**7.5. XXXVI.** For each  $r \geq 1$ : If a predicate  $P$  of variables of types  $\leq r+1$  is of order  $r+1$  in functions  $\Psi$  of variables of types  $\leq r$ , then  $P$  is  $r+1$ -expressible in  $=, +1, \Psi$ .

**Proof.** In an  $r+1$ -expression for  $P(a)$  under 7.1, consider each prime part  $Q(b)$  where  $Q$  is general recursive in  $\Psi$ . We can write  $Q(b) \equiv \phi(b) = 0$  where  $\phi$  is the representing function of  $Q$ , and apply XXXV. Finally,  $+$  and  $\cdot$  can be replaced by their representing predicates (e.g. by IM, p. 411), which by Dedekind's method (cf. IM, p. 242) can be 2-expressed in  $=, 0, +1$ <sup>(13)</sup>.

**7.6. REMARK 3.** In XXXV, XXXVI, by  $=$  we mean of course the predicate  $\alpha^0 = \beta^0$ . For  $r > 0$ ,  $\alpha^r = \beta^r \equiv (\tau^{r-1})[\alpha^r(\tau^{r-1}) = \beta^r(\tau^{r-1})]$ . Thus the predi-

cate  $\alpha^r = \beta^r$  is of order  $r$  (but not less, as will be shown in Remark 7 end 7.11).

REMARK 4. Although in 7.1 we did not list the  $\lambda$ -operator in the vocabulary for  $r$ -expressions, for expressing predicates of order  $r$  under 7.1 and 7.2 we can use it via IV, IV\*, XXIII or XXIII\* (4.4).

REMARK 5. The restriction on the types of the variables in XXXV and XXXVI is necessary in general. For example,  $\alpha^{r+2}(\lambda\beta^r \chi(\beta^r)) = w$  with a primitive recursive  $\chi$  is not  $r+1$ -expressible in  $=, +, \cdot$  as there would be no way the  $\alpha^{r+2}$  could be used; and likewise  $\psi(\lambda\beta^r \chi(a, \beta^r)) = w$  is not  $r+1$ -expressible in  $=, +, \cdot, \psi$  (unless  $\psi(\lambda\beta^r \chi(a, \beta^r))$  is constant).

7.7. In an  $r+1$ -expression, (a) any function  $\psi(b)$  can be replaced as primitive by its representing predicate  $\psi(b) = w$  (e.g. as in IM, p. 411), and (b) any function  $\psi(b)$  which is the representing function of a predicate can be replaced as primitive by that predicate  $Q(b)$  (from (a) similarly to [14, 2.5]). Only type-0 quantifiers are introduced by these replacements.

7.8. Applying XXXVI with 7.7, similarly to [14, 2.6], if a predicate  $P$  of variables of types  $\leq r+1$  is of order  $r+1$  in predicates and functions  $\Psi$  of variables of types  $\leq r$  of order  $r+1$  in  $\Theta$ , then  $P$  is of order  $r+1$  in  $\Theta$ .

7.9. We turn now to reductions which minimize the use of quantifiers rather than of primitives. For brevity we state the theorems for  $\Psi$  empty, but via 3.15 as in the proof of XXXV Part (b) they have relativized forms. The  $r = 1$  case of XXXVII for  $\alpha$  of types  $\leq 1$  is [14, Theorem 1].

XXXVIIa. *For each  $r \geq 1$ : Each predicate  $P(a)$  of order  $r+1$  is expressible in one of the following forms where  $B(a)$  is of order  $r$  and each  $R$  is general recursive:*

$$(c_1) \quad B(a) \quad (\alpha^r)(E\xi^{r-1})R(a, \alpha^r, \xi^{r-1}) \quad (E\alpha^r)(\beta^r)(E\xi^{r-1})R(a, \alpha^r, \beta^r, \xi^{r-1}) \dots \\ (E\alpha^r)(\xi^{r-1})R(a, \alpha^r, \xi^{r-1}) \quad (\alpha^r)(E\beta^r)(\xi^{r-1})R(a, \alpha^r, \beta^r, \xi^{r-1}) \dots$$

**Proof, for variables  $a$  of types  $\leq r$ .** Essentially as before ([14, 3.6]). In detail: Consider some  $r+1$ -expression for  $P(a)$  in general recursive predicates (7.1). If this  $r+1$ -expression contains no quantifiers of type  $r$ , it is of the first form  $B(a)$ . Otherwise bring it to prenex form. Now apply the following steps in order.

STEP 1. Contract each sequence of several adjacent quantifiers of like kind to one quantifier of the same kind and the maximum  $m$  of their types, by 2.5. (The scope remains general recursive, using for  $m > 0$  XXIII.)

STEP 2. If the rightmost quantifier of type  $r$  has no type- $r-1$  quantifier to the right of it, reduce it to type  $r-1$  by XXXIV (61) or its dual ( $\bar{61}$ ) if  $r \geq 2$  (by [14, (7) or (8)] if  $r = 1$ ). If then no type- $r$  quantifier remains, we have the first form  $B(a)$ .

STEP 3. If more than one quantifier of types  $< r$  stand to the right of the rightmost type- $r$  quantifier, remove all but one of them, by advancing the rightmost of them and performing contractions (or using the technique illustrated end 2.5).

STEP 4. Remove each group of quantifiers of types  $< r$  included between two type- $r$  quantifiers, or to the left of the leftmost type- $r$  quantifier, by advancing the type- $r$  quantifier immediately to their right and performing contractions, and if type- $r$  quantifiers of the same kind are thereby brought together contract them.

REMARK 6. As before [14, Remark p. 317], the described procedure is best in the sense formulated in XLI below.

**Proof, without restriction on  $a$ .** In this case we cannot use Step 2 in general. In lieu of it we may be obliged to introduce a redundant  $(E\xi^{r-1})$  or  $(\xi^{r-1})$  at the right (whereupon XLI will no longer apply).

XXXVIIb. *Equivalently the forms can be written as follows where the  $B$  for each is of order  $r$  (i.e. the same predicates are expressible in a given form of  $(c_1)$  as in the respective form of  $(c_2)$ ):*

$$(c_2) \quad B(a) \quad \begin{array}{lll} (\alpha^r)B(a, \alpha^r) & (E\alpha^r)(\beta^r)B(a, \alpha^r, \beta^r) & \dots \\ (E\alpha^r)B(a, \alpha^r) & (\alpha^r)(E\beta^r)B(a, \alpha^r, \beta^r) & \dots \end{array}$$

A third part of this theorem, using set variables (cf. [14, p. 317]), is intended for Part II of the paper.

7.10. XXXVIII. *For each  $r \geq 1$ : To each of the forms (c) of XXXVII after the first, when  $a$  are variables of types  $\leq r+1$ , there is an enumerating predicate of that form with primitive recursive scope for the predicates of that form. For example, to each fixed list  $a$  of such variables, there is a primitive recursive predicate  $T(z, a, \alpha^r, \xi^{r-1})$  such that, to any general recursive predicate  $R(a, \alpha^r, \xi^{r-1})$ ,*

$$(73) \quad (\alpha^r)(E\xi^{r-1})R(a, \alpha^r, \xi^{r-1}) \equiv (\alpha^r)(E\xi^{r-1})T(f, a, \alpha^r, \xi^{r-1}),$$

$$(74) \quad (E\alpha^r)(\xi^{r-1})R(a, \alpha^r, \xi^{r-1}) \equiv (E\alpha^r)(\xi^{r-1})\bar{T}(g, a, \alpha^r, \xi^{r-1})$$

when  $f, g$  are indices of  $R(a, \alpha^r, \xi^{r-1}), \bar{R}(a, \alpha^r, \xi^{r-1})$ , respectively<sup>(14)</sup>.

**Proof.** Say e.g.  $a$  consists of exactly one variable  $\gamma^i$  of each type  $\leq r+1$ . By (14), when  $z$  is an index of a general recursive function of  $(a, \alpha^r, \xi^{r-1})$ ,

$$(75) \quad \begin{aligned} & (\alpha^r)(E\xi^{r-1})\{z\}[\langle \gamma^0 \rangle, \dots, \langle \gamma^{r-2} \rangle, \langle \gamma^{r-1}, \xi^{r-1} \rangle, \langle \gamma^r, \alpha^r \rangle, \langle \gamma^{r+1} \rangle] = 0 \\ & \equiv (\alpha^r)(E\xi^{r-1})(\beta^r)(E\eta^{r-1})I(z, \langle \gamma^0 \rangle, \dots, \\ & \quad \langle \gamma^{r-2} \rangle, \langle \gamma^{r-1}, \xi^{r-1} \rangle, \langle \gamma^r, \alpha^r \rangle, \langle \gamma^{r+1} \rangle, 0, \beta^r, \eta^{r-1}), \end{aligned}$$

$I$  being primitive recursive. The expression on the right comes to the form

<sup>(14)</sup> The  $T$ 's here are different from those of 5.21-5.26. For  $r = 1$  with  $a$  of types  $\leq 1$ , there are also enumerating predicates using those; e.g. for  $a = (a, b, \alpha)$ , quantifying  $\beta$  in XXXIII and writing  $T_2^{\alpha, \beta}(z, a, b, y) \equiv T_2^{1,1}(\bar{\alpha}(y), \bar{\beta}(y), z, a, b)$ : (73a)  $(\beta)(Ey)R(a, b, \alpha, \beta, y) \equiv (\beta)(Ey)T_2^{\alpha, \beta}(f, a, b, y)$ , (74a)  $(E\beta)(y)R(a, b, \alpha, \beta, y) \equiv (E\beta)(y)\bar{T}_2^{\alpha, \beta}(g, a, b, y)$  when  $f, g$  are indices of  $\lambda a b \alpha \beta \mu y R(a, b, \alpha, \beta, y)$ ,  $\lambda a b \alpha \beta \mu y \bar{R}(a, b, \alpha, \beta, y)$ , respectively.—Cf. XXXVIII with L below.

$(\alpha^r)(E\xi^{r-1})T(z, a, \beta^r, \xi^{r-1})$  with primitive recursive  $T$  by Step 4. For  $f, g$  as stated, (73) and (74) follow by using (13).

XXXIX. For each  $r \geq 1$ : The class of the predicates expressible in a given one of the forms  $(c_1)$  after the first, when  $a$  are variables of types  $\leq r+1$ , is the same whether a general recursive or only a primitive recursive  $R$  be allowed.

7.11. XL. For each  $r \geq 1$ : To each of the forms  $(c)$  after the first, when  $a$  is a nonempty list of variables of types  $\leq r+1$ , there is a predicate expressible in that form but not in the dual form (*a fortiori*, not in any of the forms with fewer type- $r$  quantifiers).

**Proof.** Say  $a = (\gamma^m, b)$ . Consider e.g. the first upper form, and the predicate  $(\alpha^r)(E\xi^{r-1})T((\gamma^m)_0^0, \gamma^m, b, \alpha^r, \xi^{r-1})$  of the form (cf. 2.4). For any general recursive  $R$ , using (74) and (9),  $(E\alpha^r)(\xi^{r-1})R(\gamma^m, b, \alpha^r, \xi^{r-1}) \equiv (E\alpha^r)(\xi^{r-1})\bar{T}((g^m)_0^0, \gamma^m, b, \alpha^r, \xi^{r-1}) \not\equiv (\alpha^r)(E\xi^{r-1})T((g^m)_0^0, \gamma^m, b, \alpha^r, \xi^{r-1})$  for a certain number  $g$ . This shows that  $\langle g \rangle^m$  is a value of  $\gamma^m$  for which  $(E\alpha^r)(\xi^{r-1})R(\gamma^m, b, \alpha^r, \xi^{r-1})$  is inequivalent to our predicate.

**Second proof for  $m = 0$ .** Similarly using  $(\alpha^r)(E\xi^{r-1})T(a, a, b, \alpha^r, \xi^{r-1})$ .

XLI. To any prenex form with quantifiers of types  $\leq r$  and recursive scope and a nonempty list  $a$  of free variables of types  $\leq r$ , there is a predicate of that form which is expressible in no others of the forms  $(c_1)$  than the one to which Steps 1-4 reduce it except forms with more quantifiers.

**Proofs.** As before [14, Corollary Theorem 3, p. 319].

XLII. When  $a$  includes variables of type  $> 1$ , there is a predicate  $(\alpha)R(a, \alpha)$  with  $R$  recursive which is not expressible in the form  $(x)R(a, x)$  with  $R$  recursive, and dually (in contrast to [14, (7) and (8), p. 316], which can be stated for any list  $a$  of free variables of types  $\leq 1$ ).

**Proof.** For example, with  $(a, \alpha) = (a, \alpha^2, \xi^1)$ , were  $(\xi^1)\bar{T}(a, a, \alpha^2, \xi^1) \equiv (x)R(a, \alpha^2, x)$  with a recursive  $R$ , then XXXIV would bring  $(E\alpha^2)(\xi^1)\bar{T}(a, a, \alpha^2, \xi^1)$  to the form  $(E\eta^1)(x)R(a, \eta^1, x)$  with a recursive  $R$ , contradicting XL.

XLIII. The conclusion of XXXIV does not hold in general when the free variables  $a$  include ones of type  $> r$ .

**Proof.** Were  $(E\xi^r)T(a, a, \alpha^{r+1}, \xi^r) \equiv (E\eta^{r-1})(\xi^{r-2})R(a, \alpha^{r+1}, \eta^{r-1}, \xi^{r-2})$  with a recursive  $R$ , then by a (correct) application of XXXIV we would have  $(\alpha^{r+1})(E\xi^r)T(a, a, \alpha^{r+1}, \xi^r) \equiv (\alpha^r)(E\xi^{r-1})R(a, \alpha^r, \xi^{r-1})$  with a recursive  $R$ , contradicting XL.

REMARK 7. For  $r \geq 1$ , were  $\alpha^{r+1} = \beta^{r+1}$  (cf. Remark 3) of order  $r$ , we would have

$$(a) \quad \alpha^{r+1} = \beta^{r+1} \equiv (Qb^{r-1})R(\alpha^{r+1}, \beta^{r+1}, b^{r-1})$$

with  $R$  general recursive and  $(Qb^{r-1})$  quantifiers on variables of types  $\leq r-1$ . Letting  $\psi(a, \alpha^r)$  be the representing function of  $(E\xi^{r-1})T(a, a, \alpha^r, \xi^{r-1})$ , we would then have

$$(b) \quad (\alpha^r)(E\xi^{r-1})T(a, a, \alpha^r, \xi^{r-1}) \equiv (Qb^{r-1})R(\lambda\alpha^r \psi(a, \alpha^r), \lambda\alpha^r 0, b^{r-1}).$$

Since by XXIII\*  $R(\lambda\alpha^r \psi(a, \alpha^r), \lambda\alpha^r 0, \mathfrak{b}^{r-1})$  is general recursive in  $\psi$ , by XXXV (applied to its representing function) it is  $r+1$ -expressible in  $=, +, \cdot, \psi$ , with a prenex  $r+1$ -expression in which all the type- $r$  quantifiers are existential; in this expression,  $\psi(a, \alpha^r)$  can be replaced (IM, p. 411) by the represented predicate  $(E\xi^{r-1})T(a, a, \alpha^r, \xi^{r-1})$ . Using the result in the right member of (b), and advancing and contracting quantifiers, we would obtain for  $(\alpha^r)(E\xi^{r-1})T(a, a, \alpha^r, \xi^{r-1})$  an expression of the form  $(E\alpha^r)(\xi^{r-1})R(a, \alpha^r, \xi^{r-1})$  with  $R$  recursive, contradicting XL. Similarly, were  $\alpha^1 = \beta^1$  of order 0, we would have  $(x)\bar{T}_1(a, a, x)$  recursive, contradicting IM, Theorem V Part I, p. 283.

**7.12. XLIV.** For each  $r \geq 1$  and  $m \leq r+1$ : To each of the forms (c) after the first, there is a predicate  $C(\gamma^m)$  of the form such that any predicate of the form with its free variables  $a$  of types  $\leq m$  is expressible by substitution of a primitive recursive function of  $a$  for  $\gamma^m$  in  $C(\gamma^m)$ .

**Proof.** The case of a general  $a$  is reduced to  $a = \gamma^m$  by XI. We use the same predicate as for XL. By XXIII and (73), for any general recursive  $R$ ,  $(\alpha^r)(E\xi^{r-1})R((\gamma^m)_1, \alpha^r, \xi^{r-1}) \equiv (\alpha^r)(E\xi^{r-1})T(f, \gamma^m, \alpha^r, \xi^{r-1})$  for a suitable number  $f$ . Now  $(\alpha^r)(E\xi^{r-1})R(\gamma^m, \alpha^r, \xi^{r-1}) \equiv (\alpha^r)(E\xi^{r-1})R((\langle f, \gamma^m \rangle)_1, \alpha^r, \xi^{r-1}) \equiv (\alpha^r)(E\xi^{r-1})T((\langle f, \gamma^m \rangle)_0^0, \langle f, \gamma^m \rangle, \alpha^r, \xi^{r-1})$ .

**Second proof for  $m=0$ .** Again we use the same predicate. Let  $e$  be an index of  $\lambda b\alpha^r \xi^{r-1} R(a, \alpha^r, \xi^{r-1})$ . Then by XIII,  $S^1(e, a)$  is one of  $\lambda b\alpha^r \xi^{r-1} R(a, \alpha^r, \xi^{r-1})$ . So by (73) (substituting  $S^1(e, a)$  for  $b$ ),  $(\alpha^r)(E\xi^{r-1})R(a, \alpha^r, \xi^{r-1}) \equiv (\alpha^r)(E\xi^{r-1})T(S^1(e, a), S^1(e, a), \alpha^r, \xi^{r-1})$ .

**7.13. XLV.** For each  $r \geq 1$  and  $k \geq 0$ : Let  $P$  be a predicate of variables of types  $\leq r+1$ , and  $\Psi$  predicates of variables of types  $\leq r$ . If  $P$  is of order  $r$  in  $\Psi$ , and  $\Psi$  are expressible in both the  $k+1$ -(type- $r$ )-quantifier forms of (c), then  $P$  is expressible in both the  $k+1$ -(type- $r$ )-quantifier forms of (c).

**Proof.** Consider, in a given  $r$ -expression for  $P$  under 7.1, each prime occurrence  $R(a)$  ( $\equiv \phi(a)=0$ ) of a predicate general recursive in  $\Psi$ . According as this occurrence is positive or negative (cf. [14, p. 321]), apply XXXV with 7.7 (b) to replace it by a prenex  $r+1$ -expression with the type- $r$  quantifiers only universal or only existential and with  $=, +, \cdot, \Psi$  as the primitives. Now consider in this prenex form of each  $R(a)$  each prime part  $Q(b)$  with one of  $\Psi$  as its predicate symbol  $Q$ . According as this part is a positive or negative occurrence in the expression for  $P$ , replace it by the  $k+1$ -quantifier form for  $Q$  with  $(\alpha^r)$  first or  $(E\alpha^r)$  first. Now the quantifiers can be advanced and contracted so that a  $k+1$ -quantifier expression for  $P$  with  $(\alpha^r)$  first is obtained (cf. [14, pp. 321-322]). Reversing the above choices, a  $k+1$ -quantifier expression for  $P$  with  $(E\alpha^r)$  first is obtained similarly.

**8.  $\mu$ -RECURSIVENESS VERSUS GENERAL RECURSIVENESS. 8.1.** We say a function  $\phi$  is *partial  $\mu$ -recursive*, if it is describable by a succession of applications of the primitive recursive schemata S1-S8 (written with  $\simeq$  instead of  $=$ , and

taking  $\alpha^i(\lambda\alpha^{i-2} \chi(\alpha^i, \alpha^{i-2}, \mathbf{b}))$  to be undefined when  $\lambda\alpha^{i-2} \chi(\alpha^i, \alpha^{i-2}, \mathbf{b})$  is incompletely defined<sup>(10)</sup> and one further schema

$$S10 \quad \phi(\mathbf{b}) \simeq \mu y[\chi(\mathbf{b}, y) = 0].$$

A function is  $\mu$ -recursive, if it is partial  $\mu$ -recursive and completely defined. Also cf. 3.14, 1.8–1.10, 2.7, 3.15.

By XVI the  $\mu$ -recursive (partial  $\mu$ -recursive) functions constitute a subset of the general (partial) recursive functions, which by the normal form theorem XXX, XXXII (or [8], IM) is the whole set in the case of functions of variables of types  $\leq 1$ .

8.2. In comparing  $\mu$ - and general recursiveness for functions of variables of types  $> 1$ , we employ the particular type-2 object  $\mathbf{E}$  defined thus:

$$\mathbf{E}(\alpha) = \begin{cases} 0 & \text{if } (Et)[\alpha(t) = 0], \\ 1 & \text{otherwise.} \end{cases}$$

This is a simple example of a functional depending on infinitely many values of its function arguments. (Another is the representing function  $\psi(\alpha^1, \beta^1)$  of  $\alpha^1 = \beta^1$ ; cf. Remark 3 in 7.6. Note that  $\psi(\alpha, \beta) = \overline{\text{sg}}(\mathbf{E}(\lambda t \overline{\text{sg}}|\alpha(t) - \beta(t)|))$  and  $\mathbf{E}(\alpha) = \overline{\text{sg}}(\psi(\lambda t \text{sg}(\alpha(t)), \lambda t 1))$ .)

8.3. In the description of a partial  $\mu$ -recursive function  $\phi(\mathbf{F}, a_1, \dots, a_n, \beta)$  with a single type-1 variable  $\beta$  and a single type-2 variable  $\mathbf{F}$ , only S1–S5, S6.0, S7, S8.2, S10 can be used (if we exclude identical uses of S6.1 and S6.2).

By the  $\mathbf{F}$ -height of such a description we shall mean the greatest number of applications of S8.2 in any branch when the description is written in tree form (cf. IM, pp. 106–107, taking into account the analogy between descriptions and proofs, derivations and deductions, pp. 220, 224). Thus the  $\mathbf{F}$ -height after an application of S4 is the maximum of the  $\mathbf{F}$ -heights for the  $\psi$  and the  $\chi$ ; after an application of S8.2, the  $\mathbf{F}$ -height for the  $\chi$  increased by one.

A partial function  $\phi_1(\mathbf{a})$  is an extension of a partial function  $\phi(\mathbf{a})$ , if  $\phi_1(\mathbf{a})$  is defined and  $= \phi(\mathbf{a})$  for each  $\mathbf{a}$  for which  $\phi(\mathbf{a})$  is defined.

XLVI. If  $\phi(\mathbf{F}, a_1, \dots, a_n, \beta)$  is partial  $\mu$ -recursive with a description of  $\mathbf{F}$ -height  $h$ , then there is a function  $\phi_1^\beta(a_1, \dots, a_n)$  partial recursive in  $L_h^\beta$  uniformly in  $\beta$  such that, for each  $\beta$  (and the fixed  $\mathbf{E}$  of 8.2),  $\phi_1^\beta(a_1, \dots, a_n)$  is an extension of  $\phi(\mathbf{E}, a_1, \dots, a_n, \beta)$ . Similarly without the  $\beta$ , i.e. for a  $\phi(\mathbf{F}, a_1, \dots, a_n)$  using  $L_h$  simply<sup>(15)</sup>.

**Proof** with the  $\beta$ , by induction on the length of a description of  $\phi$  by S1–S8, S10.

<sup>(15)</sup> For  $L_h^\beta$ , cf. [19, p. 400] taking  $A(a) \equiv \beta((a)_0) = (a)_1$ , or [16, p. 198] with  $Q(a) \equiv \beta((a)_0) = (a)_1$ . For this §8, it is immaterial whether we rework the definitions and theory of  $L_h^\beta$ ,  $L_h$ , and  $H_\nu$  ([14, §6 ff.] or [16, p. 200]) to use the  $T_1^P$  bases on indices as in 5.21–5.26 above (which reworking does not alter the degrees), or agree that in this section the  $T_1^P$  is to be the one based on Gödel numbers (as in IM, [19; 14; 16]).

CASES 1, 2, 3, 7.  $\phi(\mathbf{F}, a_1, \dots, a_n, \beta)$  is introduced by one of S1-S3, S7. Then the  $\mathbf{F}$ -height is 0, and  $\phi_1^\beta(a_1, \dots, a_n) = \phi(\mathbf{E}, a_1, \dots, a_n, \beta)$  is primitive recursive uniformly in  $\beta$ , and hence general recursive in  $L_0^\beta$  uniformly in  $\beta$ .

CASE 4.  $\phi(\mathbf{F}, a_1, \dots, a_n, \beta) = \psi(\mathbf{F}, \chi(\mathbf{F}, a_1, \dots, a_n, \beta), a_1, \dots, a_n, \beta)$  by S4. Let the  $\mathbf{F}$ -heights of  $\psi, \chi$  be  $h_1, h_2$  (so  $h = \max(h_1, h_2)$ ). By hyp. ind. there are  $\psi_1^\beta, \chi_1^\beta$ , partial recursive in  $L_{h_1}^\beta, L_{h_2}^\beta$  uniformly in  $\beta$ , hence (since  $L_k^\beta(a) \equiv L_{k+1}^\beta(\theta(a))$  with a primitive recursive  $\theta$ , by IM, p. 343 or [14, Lemma 1, p. 325]) in  $L_h^\beta$  uniformly in  $\beta$ , such that  $\psi_1^\beta(b, a_1, \dots, a_n), \chi_1^\beta(a_1, \dots, a_n)$  are extensions of  $\psi(\mathbf{E}, b, a_1, \dots, a_n, \beta), \chi(\mathbf{E}, a_1, \dots, a_n, \beta)$ , respectively. Then  $\phi_1^\beta(a_1, \dots, a_n) = \psi_1^\beta(\chi_1^\beta(a_1, \dots, a_n), a_1, \dots, a_n)$  is an extension of  $\phi(\mathbf{E}, a_1, \dots, a_n, \beta)$ .

CASE 8.  $\phi(\mathbf{F}, a_1, \dots, a_n, \beta) = \mathbf{F}(\lambda x \chi(\mathbf{F}, x, a_1, \dots, a_n, \beta))$  by S8.2. The  $\mathbf{F}$ -height of  $\chi$  is  $h-1$ , and by hyp. ind. there is a function  $\chi_1^\beta$  partial recursive in  $L_{h-1}^\beta$  uniformly in  $\beta$  such that  $\chi_1^\beta(x, a_1, \dots, a_n)$  is an extension of  $\chi(\mathbf{E}, x, a_1, \dots, a_n, \beta)$ . Then the function  $\phi_0^\beta$  defined by

$$(76) \quad \begin{aligned} \phi_0^\beta(a_1, \dots, a_n) &= \mathbf{E}(\lambda x \chi_1^\beta(x, a_1, \dots, a_n)) \\ &= \begin{cases} 0 & \text{if } (Ex)[\chi_1^\beta(x, a_1, \dots, a_n) = 0], \\ 1 & \text{otherwise,} \end{cases} \end{aligned}$$

for the fixed  $\mathbf{E}$  and any  $\beta$ , is an extension of  $\phi(\mathbf{E}, a_1, \dots, a_n, \beta)$ ; here, to correspond to our interpretation of S8 for  $\chi$  a partial function,  $\phi_0^\beta(a_1, \dots, a_n)$  is to be undefined for given  $\beta, a_1, \dots, a_n$  when  $\lambda x \chi_1^\beta(x, a_1, \dots, a_n)$  is incompletely defined. But since  $\chi_1^\beta$  is partial recursive in  $L_{h-1}^\beta$  uniformly in  $\beta$ , there is by [14, Lemma 1] a primitive recursive  $\theta$  such that

$$(77) \quad \begin{aligned} (Ex)[\chi_1^\beta(x, a_1, \dots, a_n) = 0] \\ \equiv (Ex)T_1^{L_{h-1}^\beta}(\theta(a_1, \dots, a_n), \theta(a_1, \dots, a_n), x) \equiv L_h^\beta(\theta(a_1, \dots, a_n)) \end{aligned}$$

for values of  $\beta, a_1, \dots, a_n$  which make  $\lambda x \chi_1^\beta(x, a_1, \dots, a_n)$  completely defined, i.e. which make  $\phi_0^\beta(a_1, \dots, a_n)$  defined. So by replacing  $(Ex)[\chi_1^\beta(x, a_1, \dots, a_n) = 0]$  by  $L_h^\beta(\theta(a_1, \dots, a_n))$  in (76), we obtain a function

$$(78) \quad \phi_1^\beta(a_1, \dots, a_n) = \begin{cases} 0 & \text{if } L_h^\beta(\theta(a_1, \dots, a_n)), \\ 1 & \text{otherwise,} \end{cases}$$

which is an extension of  $\phi_0^\beta$  and thence of  $\phi(\mathbf{E}, a_1, \dots, a_n, \beta)$ , and is clearly partial (indeed, primitive) recursive in  $L_h^\beta$  uniformly in  $\beta$ .

8.4. Now we give an argument by which any person who accepts the primitive recursive functions as effectively calculable, and who allows such a type-2 object as  $\mathbf{E}$ , must admit that the  $\mu$ -recursive functions are not all the effectively calculable functions.

Since  $T_1^\alpha(a, a, t)$  is primitive recursive as a predicate of  $\alpha, a, t$  (IM, p. 292, or §5 above), the function

$$\tau(t, \alpha, a) = \begin{cases} 0 & \text{if } T_1^\alpha(a, a, t), \\ 1 & \text{otherwise} \end{cases}$$

is primitive recursive. Hence using III and S8, so is

$$\tau(\mathbf{F}, \alpha, a) = \mathbf{F}(\lambda t \tau(t, \alpha, a)).$$

Using induction on  $k$  (and in the induction step, III and IV), we define a succession of primitive recursive functions  $\lambda_k(\mathbf{F}, a, \beta)$  ( $k = 0, 1, 2, \dots$ ):

$$\begin{aligned} \lambda_0(\mathbf{F}, a, \beta) &= \text{sg} | \beta((a)_0) - (a)_1 |, \\ \lambda_{k+1}(\mathbf{F}, a, \beta) &= \tau(\mathbf{F}, \lambda t \lambda_k(\mathbf{F}, t, \beta), a). \end{aligned}$$

Upon giving  $\mathbf{F}$  the fixed value  $\mathbf{E}$ , we have

$$(79) \quad \tau(\mathbf{E}, \alpha, a) = \begin{cases} 0 & \text{if } (Et)[\tau(t, \alpha, a) = 0] \\ 1 & \text{otherwise} \end{cases} = \begin{cases} 0 & \text{if } (Et)T_1^\alpha(a, a, t), \\ 1 & \text{otherwise,} \end{cases}$$

and thence by induction on  $k$ , for each  $\beta$ ,

$$(80) \quad \lambda_k(\mathbf{E}, a, \beta) = \{ \text{the representing function of } L_k^\beta(a) \}.$$

Similarly, omitting the  $\beta$  and taking  $\lambda_0(\mathbf{F}, a) = 0$ , we obtain for  $k = 0, 1, 2, \dots$  a primitive recursive function  $\lambda_k(\mathbf{F}, a)$  such that

$$(81) \quad \lambda_k(\mathbf{E}, a) = \{ \text{the representing function of } L_k(a) \}.$$

Now consider the function

$$\lambda(k, \mathbf{F}, a) = \lambda_k(\mathbf{F}, a).$$

This must be accepted as being effectively calculable (accepting that the primitive recursive functions of variables of types 0, 1, 2 are). For, given  $k, \mathbf{F}, a$ , we can via the induction on  $k$  find effectively a primitive recursive description of  $\lambda_k$ , and then “compute”  $\lambda_k(\mathbf{F}, a)$ .

But  $\lambda(k, \mathbf{F}, a)$  is not  $\mu$ -recursive. For by (81) and the definition of  $\lambda(k, \mathbf{F}, a)$ ,  $\lambda(k, \mathbf{E}, a)$  is the representing function of  $L(k, a) \equiv L_k(a)$ , which is of degree of recursive unsolvability  $\mathbf{0}^{(\omega)}$  [19, p. 401; 16, p. 198]). But by XLVI, were  $\lambda(k, \mathbf{F}, a)$   $\mu$ -recursive, then, for some  $h$ ,  $\lambda(k, \mathbf{E}, a)$  would be recursive in  $L_h$ , and thus would be of degree  $\leq \mathbf{0}^{(h)} < \mathbf{0}^{(\omega)(16)}$ .

<sup>(16)</sup> As  $\mu$ -recursiveness is one of the simplest of the equivalent notions of “effective calculability” or “computability” for functions of variables of types 0, 1 (cf. 3.2), we considered it as a possible definition for the higher types at the beginning of our study of the subject in 1952, but rejected it for the reason given now. In a 1955 paper Grzegorzcyk [7, p. 170] seems to be proposing this definition; his formulation is equivalent to  $\mu$ -recursiveness as defined here, when our schema S8.j for  $j \geq 2$  (or S4.j-1 and S7.j of 1.6, Remark 1) is added to the schemata he explicitly mentions (and indeed without some such addition the higher-type arguments of the functions could not be utilized) and the (maybe inessential) restriction is imposed on our  $\mu$ -schema S10 that at each application the  $\phi(b)$  be completely defined.

**8.5.** The function  $\lambda(k, \mathbf{F}, a)$  is general recursive, as we see by writing the equations for  $\lambda_k(\mathbf{F}, a)$  with  $k$  as argument, and applying XXIV (or by showing with the help of XXI and XXII that  $\lambda_k(\mathbf{F}, a)$  has a primitive recursive index  $\xi(k)$  with  $\xi$  primitive recursive, and putting  $\lambda(k, \mathbf{F}, a) = \{\xi(k)\}(\mathbf{F}, a)$ ).

We can go much further. Let  $u_b \simeq U(\mu v T_1(u, b_0, v))$  where  $0_0 = 1, (b+1)_0 = 2^{b_0}$  (cf. [14, p. 325] or [16, pp. 199–200]). Consider the recursion

$$(82) \quad \{z\}(y, \mathbf{F}, a) \simeq \begin{cases} 0 & \text{if } y=1, \\ \chi(z, y, \mathbf{F}, a) & \text{if } y=2^{(y)_0} \neq 1, \\ \{z\}([(y)_2]_{(a)_1}, \mathbf{F}, (a)_0) & \text{if } y=3 \cdot 5^{(y)_2}, \end{cases}$$

where  $\chi$  is a partial recursive function given by XXIII such that

$$(83) \quad \chi(z, y, \mathbf{F}, a) = \tau(\mathbf{F}, \lambda t \{z\}((y)_0, \mathbf{F}, t), a)$$

whenever  $\lambda t \{z\}((y)_0, \mathbf{F}, t)$  is completely defined. By XVIII, the right side of (82) is of the form  $\psi(z, y, \mathbf{F}, a)$  with  $\psi$  partial recursive; so by the recursion theorem XIV, we can find a solution  $e$  of (82) for  $z$ . Let  $\kappa(y, \mathbf{F}, a) \simeq \{e\}(y, \mathbf{F}, a)$ . Now by induction on  $y$  over the class  $O$  of ordinal notations (loc. cit.),

$$(84) \quad y \in O \rightarrow \{\kappa(y, \mathbf{E}, a) \text{ is the representing function of } H_y(a)\}.$$

**8.6.** We collect these results in a pair of contrasting theorems. Instead of speaking of completely defined functions  $\phi(\mathbf{E}, a_1, \dots, a_n)$  ( $\phi(\mathbf{E}, a_1, \dots, a_n, \beta)$ ) for  $\phi(\mathbf{F}, a_1, \dots, a_n)$  ( $\phi(\mathbf{F}, a_1, \dots, a_n, \beta)$ ) partial  $\mu$ -, or partial, recursive, we can equivalently (by 1.9 extended to include S10 or S9) speak of functions  $\phi(a_1, \dots, a_n)$  ( $\phi(a_1, \dots, a_n, \beta)$ )  $\mu$ -, or general, recursive in  $\mathbf{E}$ . For the notion “arithmetical,” cf. 7.3, IM, pp. 239, 284–285, 291–292, [14, 2.1]. For “hyperarithmetical,” cf. [16, p. 210]; a function is *hyperarithmetical* if its representing predicate is such.

**XLVII.** *The functions  $\phi(a_1, \dots, a_n)$  ( $\phi(a_1, \dots, a_n, \beta)$ )  $\mu$ -recursive in  $\mathbf{E}$  are exactly the arithmetical functions.*

**Proof.** with  $\beta$  present. By XLVI, for each  $\phi(a_1, \dots, a_n, \beta)$   $\mu$ -recursive in  $\mathbf{E}$ ,  $\lambda a_1 \dots a_n \phi(a_1, \dots, a_n, \beta)$  is general recursive, a fortiori arithmetical, in  $L_h^\beta$  for some  $h$  uniformly in  $\beta$ , and hence is arithmetical uniformly in  $\beta$  (using 7.8 or [14, 2.6 with uniformity], and induction on  $h$ ), i.e.  $\phi(a_1, \dots, a_n, \beta)$  is arithmetical. Conversely, for any arithmetical function  $\phi(a_1, \dots, a_n, \beta)$ ,  $\lambda a_1 \dots a_n \phi(a_1, \dots, a_n, \beta)$  is  $\mu$ -recursive in  $L_k^\beta$  for some  $k$  uniformly in  $\beta$  (using  $\phi(a_1, \dots, a_n, \beta) = \mu w [\phi(a_1, \dots, a_n, \beta) = w]$ , [16, IV\*, VII\*, p. 197 with uniformity], and [14, 9.8 (31) and (32), and Lemma 16\* 9.7]). But by (80),  $L_k^\beta(a) \equiv \lambda_k(\mathbf{E}, a, \beta) = 0$ ; and  $\lambda_k(\mathbf{F}, a, \beta)$  is primitive, a fortiori  $\mu$ -, recursive.

**XLVIII.** *The functions  $\phi(a_1, \dots, a_n)$  general recursive in  $\mathbf{E}$  are exactly the hyperarithmetical functions.*

**Proof.** Suppose  $\phi(a_1, \dots, a_n)$  is general recursive in  $\mathbf{E}$ . Equivalently,  $\phi(a_1, \dots, a_n)$  is completely defined and  $= \phi(\mathbf{E}, a_1, \dots, a_n)$  where

$\phi(\mathbf{F}, a_1, \dots, a_n)$  is partial recursive, say with index  $z$ . Using XXVI with (13),  $\phi(\mathbf{E}, a_1, \dots, a_n) = w$  is expressible in both the forms  $(\beta)(Ex)R(\mathbf{E}, a_1, \dots, a_n, w, \beta, x)$  and  $(E\beta)(x)S(\mathbf{E}, a_1, \dots, a_n, w, \beta, x)$  with  $R$  and  $S$  primitive recursive. But by XLVII,  $R(\mathbf{E}, a_1, \dots, a_n, w, \beta, x)$  and  $S(\mathbf{E}, a_1, \dots, a_n, w, \beta, x)$  are arithmetical, hence (XLV or [14, Corollary p. 322 with a free function variable  $\beta$ ]) expressible in both 1-function-quantifier forms; and hence so is  $\phi(\mathbf{E}, a_1, \dots, a_n) = w$ , i.e.  $\phi(\mathbf{E}, a_1, \dots, a_n) = w$  is hyperarithmetical (by the second definition [16, p. 210]).

Conversely, suppose  $\phi(a_1, \dots, a_n)$  is hyperarithmetical, i.e.  $\phi(a_1, \dots, a_n) = w$  is hyperarithmetical. Then by [16, XXIV, p. 204, or the first definition p. 210], for some  $y \in O$ ,  $\phi(a_1, \dots, a_n) = w$  and hence  $\phi(a_1, \dots, a_n) (= \mu w[\phi(a_1, \dots, a_n) = w])$ , is general recursive in  $H_y(a)$ . But by (84),  $H_y(a) \equiv \kappa(y, \mathbf{E}, a) = 0$ ; and  $\kappa(y, \mathbf{F}, a)$  is partial recursive.

**8.7. XLIX.** (a) *A predicate  $P(a)$  expressible in both the forms  $(Ex)R(a, x)$  and  $(x)S(a, x)$  with  $R, S$  general recursive is general recursive.* (Converse holds trivially.) (b) *When  $a$  includes variables of types  $> 1$ , a general recursive predicate  $P(a)$  may fail to be expressible by any (finite) number of number-quantifiers prefixed to a  $\mu$ - (or primitive) recursive scope* (in contrast to IM, Theorem VI\* (a), pp. 284, 292, and Corollary Theorem IV\*, pp. 282, 292, for types 0, 1).

**Proof.** (a) By the former proof, IM, Theorem VI (b), p. 284. (b) By 8.5,  $\lambda(k, \mathbf{F}, a) = 0$  is general recursive. But it is not expressible e.g. in the form  $(Ex)(y)R(k, \mathbf{F}, a, x, y)$  with a  $\mu$ -recursive  $R$ . For by 8.4,  $\lambda(k, \mathbf{E}, a) = 0 \equiv L(k, a)$ , which is not arithmetical; but using XLVII, for a given  $\mu$ -recursive  $R$ ,  $(Ex)(y)R(k, \mathbf{E}, a, x, y)$  is arithmetical.

**8.8. L.** *With variables of types  $> 1$ , there is no enumeration theorem for the predicates definable by a given succession of number quantifiers applied to a general ( $\mu$ - or primitive) recursive scope. For example, there is no general recursive predicate  $S(z, \mathbf{F}, a, x)$  with the property that, to each general recursive  $R(\mathbf{F}, a, x)$ , there is a number  $e$  such that  $(Ex)R(\mathbf{F}, a, x) \equiv (Ex)S(e, \mathbf{F}, a, x)$  (in contrast to XXXIII or IM, Theorem IV\*, pp. 281, 292 for types 0, 1).*

**Proof.** (We can use XLVIII, but the following is basically simpler.) Take any fixed general recursive  $S(z, \mathbf{F}, a, x)$ . We construct as follows a general recursive  $R(\mathbf{F}, a, x)$  such that  $(Ex)R(\mathbf{F}, a, x) \equiv (Ex)S(e, \mathbf{F}, a, x)$  for no  $e$ . Let  $\sigma$  be the representing function of  $S$ . Using 8.4 and IV via 3.10, define  $\rho(\mathbf{F}, a, x) = \tau(x, \lambda y \tau(\mathbf{F}, \lambda t \sigma(t)_0, \mathbf{F}, (t)_1, (t)_2, y), a)$ , and take  $R(\mathbf{F}, a, x) \equiv \rho(\mathbf{F}, a, x) = 0$ . Let  $\mathbf{s}$  be the degree of  $S(z, \mathbf{E}, a, x)$  ( $=$  the degree of  $S(t)_0, \mathbf{E}, (t)_1, (t)_2$ ). Then  $(Ex)S(z, \mathbf{E}, a, x)$  is of degree  $\leq \mathbf{s}'$ , while  $(Ex)R(\mathbf{E}, a, x)$  is of degree  $\mathbf{s}'' > \mathbf{s}'$  (cf. [19, 1.4]).

**8.9. LI.** *The condition of definition of a partial recursive function  $\phi(a)$  of variables of types  $\leq 2$  is not in general expressible in the form  $(E\beta)(x)R(a, \beta, x)$  with recursive  $R$  (dual to XXVII (29)), a fortiori not in the form  $(Ex)R(a, x)$  with recursive  $R$  (in contrast to XXIX or IM, Theorem XIX, p. 330 for types 0, 1). Thus (30), (39) and (41) do not hold for  $r = 2$ .*

**Proof.** Suppose (cf. 8.5)

$$(a) \quad \{\kappa(y, \mathbf{F}, a) \text{ is defined}\} \equiv (E\beta)(x)R(y, \mathbf{F}, a, \beta, x)$$

with a primitive recursive  $R$  (cf. XXXIX). Using (29) and (13), for  $z$  an index of  $\kappa$ ,

$$(85) \quad \{\kappa(y, \mathbf{F}, a) \text{ is defined}\} \equiv (\beta)(Ex)K(z, \langle y, a \rangle, \langle \rangle^1, \langle \mathbf{F} \rangle, \beta, x).$$

But then, using XLVII etc. as in the first part of the proof of XLVIII,  $\{\kappa(y, \mathbf{E}, a) \text{ is defined}\}$  would be hyperarithmetical; so by the converse part of XLVIII, there would be a partial recursive  $R(y, \mathbf{F}, a)$  such that

$$(b) \quad \{\kappa(y, \mathbf{E}, a) \text{ is defined}\} \equiv R(y, \mathbf{E}, a).$$

Using XV, we could define a partial recursive  $\kappa'(y, \mathbf{F}, a)$  by

$$\kappa'(y, \mathbf{F}, a) = \begin{cases} \kappa(y, \mathbf{F}, a) & \text{if } R(y, \mathbf{F}, a), \\ 0 & \text{otherwise.} \end{cases}$$

Then  $\kappa'(y, \mathbf{E}, a)$  would be a completion of  $\kappa(y, \mathbf{E}, a)$ . This is absurd. For by XLVIII and [16, XXIV or p. 210],  $\kappa'(y, \mathbf{E}, a)$  would be recursive in  $H_u$  for some  $u \in O$ , i.e. of degree  $\leq 0^{(1+u)}$ ; but any completion of  $\kappa(y, \mathbf{E}, a)$  must be of degree  $\geq 0^{(1+u+1)} > 0^{(1+u)}$ , since by (84)  $H_{2^u}(a) \equiv \kappa(2^u, \mathbf{E}, a) = 0$ .

**8.10. LII.** *There is a primitive recursive  $R(\alpha^2, \beta^1)$  such that  $(\alpha^2)(E\beta^1)R(\alpha^2, \beta^1)$  is true, but  $(\alpha^2)R(\alpha^2, \lambda x \chi(\alpha^2, x))$  is false for every general recursive  $\chi(\alpha^2, x)$  (in contrast to XVI or IM, Theorem III, p. 279 for  $R(a, \beta^0)$ )<sup>(17)</sup>.*

**Proof.** By [16, XXVI, p. 208] there is a primitive recursive  $R(\beta^1, x)$  such that

$$(86a) \quad (E\beta^1)(x)R(\beta^1, x), \quad (86b) \quad \overline{(E\beta^1)}_{\beta^1 \text{ hyp}}(x)R(\beta^1, x)$$

where  $(E\beta^1)_{\beta^1 \text{ hyp}}$  is an existential quantifier over the 1-place hyperarithmetical functions. Advancing the  $x$  in (86a) (cf. 2.5) and bringing (86b) to prenex form,

$$(87a) \quad (\alpha^2)(E\beta^1)R(\beta^1, \alpha^2(\beta^1)), \quad (87b) \quad (\beta^1)_{\beta^1 \text{ hyp}}(Ex)\overline{R}(\beta^1, x).$$

Taking  $R(\alpha^2, \beta^1) \equiv R(\beta^1, \alpha^2(\beta^1))$ , (87a) gives one part of LII. Let

$$\alpha_0^2(\beta^1) = \begin{cases} \mu x \overline{R}(\beta^1, x) & \text{if } (Ex)\overline{R}(\beta^1, x), \\ 0 & \text{otherwise,} \end{cases}$$

so by (87b)

<sup>(17)</sup> In March 1957, Kreisel raised the questions (a) whether there is any such  $R(\alpha^2, \beta^1)$  and (b) whether the  $R(\alpha^2, \beta^1)$  obtained by advancing quantifiers in  $(x)\{(Ez)T_1(x, x, z) \vee (z)T_1(x, x, z)\}$  (cf. [10, p. 71]) is such, and we answered (a) as here. Subsequently Kreisel found two simple examples of the like with  $R(\alpha^2, \beta^2)$  instead of  $R(\alpha^2, \beta^1)$  (cf. [21], and also [18, Theorem 3]). Finally in June 1957, Kreisel gave an argument which, supported by a lemma we provided, answers (b) affirmatively; this will appear in Part II.

$$(88) \quad (\beta^1)_{\beta^1 \text{ hyp}} \bar{R}(\beta^1, \alpha_0^2(\beta^1)).$$

But  $(Ex)\bar{R}(\beta^1, x) \equiv L_1\beta^1(g)$  for some number  $g$  (by [16, VII\*, p. 197 with uniformity and  $a = 0$ ] or [14, Lemma 1, p. 325 with  $n = 0$ ]). Using XV, XVI and 8.4,

$$\alpha_0(\mathbf{F}, \beta^1) = \begin{cases} \mu x \bar{R}(\beta^1, x) & \text{if } \lambda_1(\mathbf{F}, g, \beta^1) = 0, \\ 0 & \text{otherwise} \end{cases}$$

is a general recursive function such that, by (80),

$$(89) \quad \alpha_0^2(\beta^1) = \alpha_0(\mathbf{E}, \beta^1).$$

Now suppose there were a general recursive  $\chi(\alpha^2, x)$  such that  $(\alpha^2)R(\alpha^2, \lambda x \chi(\alpha^2, x))$ , i.e.  $(\alpha^2)R(\lambda x \chi(\alpha^2, x), \alpha^2(\lambda x \chi(\alpha^2, x)))$ , whence  $R(\lambda x \chi(\alpha_0^2, x), \alpha_0^2(\lambda x \chi(\alpha_0^2, x)))$ . This would contradict (88), since by (89), XXIII and XLVIII,  $\lambda x \chi(\alpha_0^2, x) = \lambda x \chi(\lambda\beta^1 \alpha_0(\mathbf{E}, \beta^1), x)$  would be hyperarithmetical.

#### BIBLIOGRAPHY

1. L. E. J. Brouwer, *Beweis, dass jede volle Funktion gleichmässig stetig ist*, Nederl. Akad. Wetensch. vol. 27 (1924) pp. 189–193.
2. ———, *Über Definitionsbereiche von Funktionen*, Math. Ann. vol. 97 (1927) pp. 60–75.
3. ———, *Points and spaces*, Canad. J. Math. vol. 6 (1953) pp. 1–17.
4. A. Church, *An unsolvable problem of elementary number theory*, Amer. J. Math. vol. 58 (1936) pp. 345–363.
5. M. Davis, *On the theory of recursive unsolvability*, Ph.D. thesis (typewritten), Princeton University, 1950.
6. A. Grzegorzcyk, *Some classes of recursive functions*, Rozprawy Mat. no. 4, Warsaw, 1953, 46 pp.
7. ———, *Computable functionals*, Fund. Math. vol. 42 (1955) pp. 168–202.
8. S. C. Kleene, *General recursive functions of natural numbers*, Math. Ann. vol. 112 (1936) pp. 727–742.
9. ———, *A note on recursive functions*, Bull. Amer. Math. Soc. vol. 42 (1936) pp. 544–546.
10. ———, *Recursive predicates and quantifiers*, Trans. Amer. Math. Soc. vol. 53 (1943) pp. 41–73. For an error in §15, cf. [15, §19].
11. ———, *On the forms of the predicates in the theory of constructive ordinals*, Amer. J. Math. vol. 66 (1944) pp. 41–58. An error is corrected in [15].
12. ———, *Recursive functions and intuitionistic mathematics*, Proceedings of the International Congress of Mathematicians (Cambridge, Mass. Aug. 30–Sept. 6, 1950) vol. 1, 1952, pp. 679–685.
13. ———, *Introduction to metamathematics*, New York and Toronto, Van Nostrand, Amsterdam, North-Holland, and Groningen, Noordhoff, 1952, X+550 pp.
14. ———, *Arithmetical predicates and function quantifiers*, Trans. Amer. Math. Soc. vol. 79 (1955) pp. 312–340. For errata, cf. ibid. p. 386 and vol. 81 (1956) p. 524, and Proc. Amer. Math. Soc. vol. 8 (1957) p. 1006.
15. ———, *On the forms of the predicates in the theory of constructive ordinals (second paper)*, Amer. J. Math. vol. 77 (1955) pp. 405–428. Errata: p. 408 l. 7 for “y” read “y”, p. 412 l. 1 for “Im(k)” read “ $\overline{\text{Im}}(k)$ ”, p. 425 for second “ $\beta$ ” l. 15 and “ $\beta$ ” l. 16 read “ $\beta$ ”; also “ $\cong$ ” should be “ $\simeq$ ” except on p. 411 l. 9 (namely on pp. 408, 411 l. 11, 412–414, 418, 427).

16. ———, *Hierarchies of number-theoretic predicates*, Bull. Amer. Math. Soc. vol. 61 (1955) pp. 193–213.
17. ———, *Extension of an effectively generated class of functions by enumeration*, Colloquium Mathematicum (Wrocław), vol. 6.
18. ———, *Countable functionals*, Constructivity in Mathematics, Amsterdam, North-Holland.
19. S. C. Kleene and E. L. Post, *The upper semi-lattice of degrees of recursive unsolvability*, Ann. of Math. (2) vol. 59 (1954) pp. 379–407. Erratum: p. 404, the next to the last “=” in (56) should be “ $\neq$ ”.
20. D. König, *Über eine Schlussweise aus dem Endlichen ins Unendliche*, Acta Sci. Math. Szeged vol. 3 (1927) pp. 121–130.
21. G. Kreisel, *Constructive interpretations of analysis by means of functionals of finite type*, Constructivity in Mathematics, Amsterdam, North-Holland.
22. R. Péter, *Über den Zusammenhang der verschiedenen Begriffe der rekursiven Funktion*, Math. Ann. vol. 110 (1934) pp. 612–632.
23. ———, *Rekursive Funktionen*, Budapest, Akademischer Verlag, 1951, 206 pp.
24. ———, *Probleme der Hilbertschen Theorie der höheren Stufen von rekursiven Funktionen*, Acta Math. Acad. Sci. Hungar. vol. 2 (1951) pp. 247–274.
25. E. L. Post, *Degrees of recursive unsolvability* (Preliminary report), Bull. Amer. Math. Soc. Abstract 54-7-269.
26. C. Spector, *Recursive well-orderings*, J. Symb. Logic, vol. 20 (1955) pp. 151–163.
27. ———, *On degrees of recursive unsolvability*, Ann. of Math. (2) vol. 64 (1956) pp. 581–592.
28. A. Tarski, *Einige Betrachtungen über die Begriffe der  $\omega$ -Widerspruchsfreiheit und der  $\omega$ -Vollständigkeit*, Monatshefte für Mathematik und Physik vol. 40 (1933) pp. 97–112.
29. ———, *A problem concerning the notion of definability*, J. Symb. Logic vol. 13 (1948) pp. 107–111.
30. A. M. Turing, *Systems of logic based on ordinals*, Proc. London Math. Soc. (2) vol. 45 (1939) pp. 161–228.

THE UNIVERSITY OF WISCONSIN,  
MADISON, WIS.  
PRINCETON UNIVERSITY,  
PRINCETON, N. J.