From S.C. Kleene *Introduction to Metamathematics* North-Holland 1971.

(and writing "φ" for "φ_k", "φ_{j_s}" for "$\varphi_{j_{ks}}$", "f" for "f_k", etc.), only those equations of the form $f(x_1, \ldots, x_n) = x$ which $\varepsilon\ E^{\varphi}_{f}$ are deducible from $E^{\varphi_{j_1} \ldots \varphi_{j_q}}_{f_{j_1} \ldots f_{j_q}}$, E_k; and by Lemma IIc no others become deducible when $E^{\varphi_{j_1} \ldots \varphi_{j_q}}_{f_{j_1} \ldots f_{j_q}}$ is replaced in the list of assumption equations by

$$E^{\psi_1 \ldots \psi_l}_{f_1 \ldots f_l},\ E_{l+1} \ldots E_{k-1}.$$

LEMMA IIe. *Let $\varphi_1, \ldots, \varphi_k$ be a finite sequence of functions such that φ_k is φ and for each i ($i = 1, \ldots, k$), either* (A) *φ_i is one of the functions $\psi_1, \ldots, \psi_l$, or* (B) *φ_i is defined recursively by a system E_i of equations from $\varphi_{j_{i1}}, \ldots, \varphi_{j_{iq_i}}$ ($q_i \geq 0$; $j_{i1}, \ldots, j_{iq_i} < i$). Then there is a system E of equations which defines φ recursively from $\psi_1, \ldots, \psi_l$.*

PROOF OF LEMMA IIe. If it is not already the case that each of the ψ's is introduced under (A) as one of the φ's and is thereafter used under (B) as one of the $\varphi_{j_{i1}}, \ldots, \varphi_{j_{iq_i}}$ for some φ_i, we can make it so (increasing k) by introducing some applications of the identical schema (cf. Lemma IIa). Then, by rearranging and renumbering the φ's and E_i's and changing the function letters in the latter (if necessary), we can bring about the situation described in Lemma IId, with $k > l$, $\varphi_k = \varphi$, and with $f_1, \ldots, f_l$ as the given function letters of $E_{l+1} \ldots E_k$. Let E be $E_{l+1} \ldots E_k$.

PROOF OF THEOREM II. By Lemma IIa and the hypothesis of the theorem, the hypotheses of Lemma IIe are satisfied.

§ 55. General recursive functions. The schemata (I)—(V) are not the only schemes of definition of a number-theoretic function, ab initio or from other number-theoretic functions, which can be expressed by systems of equations, using in the equations only function letters, $'$, number variables and numerals.

Let us consider other examples, calling them all "recursions". We keep the equations in the informal language for the time being; and to keep them of the sort described now, we eliminate certain other modes of expression which were used in Chapter IX, e.g. $\prod_{y<z}$ (#B), $\mu y_{y<z}$ (#E), cases (#F).

Thus (a) of Example 1 § 46 we can write now

$$\text{(a)} \quad \begin{cases} \pi(0, y) = 1, \\ \pi(z', y) = (y + \varphi(z)) \cdot \pi(z, y), \\ \varphi(y) = \pi(y, y) \end{cases}$$

(defining the auxiliary function π as well as φ), while (a) of Example 2 § 46 is already in the form under consideration. We showed in § 46 that these course-of-values recursions are reducible to primitive recursion, i.e. the same function can be defined by a series of applications of Schemata (I) — (V).

As another very simple example, consider the recursion

$$\text{(b)} \quad \begin{cases} \varphi(0, z) = z, \\ \varphi(y', z) = \varphi(y, \sigma(y, z)). \end{cases}$$

This is not primitive, because the z, instead of being held fixed as a parameter, has $\sigma(y, z)$ substituted for it in the induction step of the definition. This recursion too can be reduced to primitive recursion. Expanding (b) for $y = 0, 1, 2, \ldots$ (as we expanded (1) in § 43), we find that the value $\varphi(y, z)$ is

$$\sigma(0, \sigma(1, \sigma(2, \ldots \sigma(y-3, \sigma(y-2, \sigma(y-1, z)))\ldots))).$$

Consider the sequence of the numbers z, $\sigma(y-1, z)$, $\sigma(y-2, \sigma(y-1, z))$, $\ldots$, $\sigma(0, \sigma(1, \sigma(2, \ldots \sigma(y-3, \sigma(y-2, \sigma(y-1, z)))\ldots)))$, which occur in building up this value from the inside instead of as (b) gives it to us. These are the values for $u = 0, 1, 2, \ldots, y$ of the function $\mu(u, y, z)$ defined by the primitive recursion

$$\text{(b}_1\text{)} \quad \begin{cases} \mu(0, y, z) = z, \\ \mu(u', y, z) = \sigma(y \dot{-} u', \mu(u, y, z)). \end{cases}$$

Since the value for $u = y$ is the same as the value $\varphi(y, z)$,

$$\text{(b}_2\text{)} \quad \varphi(y, z) = \mu(y, y, z);$$

as can also be seen by using induction on u to prove that

$$\text{(c)} \quad \mu(u', y', z) = \mu(u, y, \sigma(y, z)),$$

and thence that the φ defined by (b$_1$) and (b$_2$) satisfies (b).

In a similar manner, Péter (1934, 1935a) showed that every recursion (called "nested") in which $\varphi(0, z)$ is a given function of z, and $\varphi(y', z)$ is expressed explicitly in terms of y, z, given functions (and constants), and $\varphi(y, t)$ as a function of t, is reducible to primitive recursion.

Are there recursions which are not reducible to primitive recursion; and in particular can recursion be used to define a function which is not primitive recursive?

This question arose from a conjecture of Hilbert 1926 on the continuum problem, and was answered by Ackermann 1928. Let $\xi_0(b, a) = a + b$, $\xi_1(b, a) = a \cdot b$, $\xi_2(b, a) = a^b$; and let this series of functions be extended by successive primitive recursions of the form $\xi_{n'}(0, a) = a$, $\xi_{n'}(b', a) = \xi_n(\xi_{n'}(b, a), a)$ ($n \geq 2$), so that e.g. $\xi_3(b, a) = a^{a^{\cdot^{\cdot^{a}}}}$ with b exponents.

Now consider $\xi_n(b, a)$ as a function $\xi(n, b, a)$ of all three variables. Let α be the primitive recursive function defined thus,

$$\text{(d)} \qquad \alpha(n, a) = \begin{cases} 0 & \text{if } n = 0, \\ 1 & \text{if } n = 1, \\ a & \text{otherwise.} \end{cases}$$

Then the following recursion defines $\xi(n, b, a)$,

$$\text{(e)} \qquad \begin{cases} \xi(0, b, a) &= a+b, \\ \xi(n', 0, a) &= \alpha(n, a), \\ \xi(n', b', a) &= \xi(n, \xi(n', b, a), a). \end{cases}$$

This is an example of a "double recursion", i.e. one on two variables simultaneously. If the function $\xi(n, b, a)$ defined by (e) were primitive recursive, then the function $\xi(a)$ of one variable defined explicitly from it thus,

$$\text{(f)} \qquad \xi(a) = \xi(a, a, a),$$

would also be primitive recursive. Ackermann's investigation shows that $\xi(a)$ grows faster with increasing a than any primitive recursive function of a (just as 2^a grows faster than any polynomial in a), i.e. given any primitive recursive function $\varphi(a)$, a natural number c can be found such that $\xi(a) > \varphi(a)$ for all $a \geq c$. Thus $\xi(a)$, and hence also $\xi(n, b, a)$ (since $\xi(a)$ comes from it by the explicit definition (f)), are not primitive recursive. This example was simplified by Péter 1935 (cf. also Hilbert-Bernays 1934 pp. 330 ff.) and Raphael Robinson 1948.

A different method was followed by Péter 1935 in constructing another example. The class of the initial functions definable by Schemata (I) — (III) is enumerable. Then the class of the primitive recursive functions definable using Schema (IV) or (V) just once is enumerable, since the $m+1$-tuples $\psi, \chi_1, \ldots, \chi_m$ for (IV) or the pairs ψ, χ (or q, χ) for (V) formed from an enumerable class are enumerable (§ 1). Then the primitive recursive functions definable using Schema (IV) or (V) a second time are enumerable; and so on. Thus the class of all the primitive recursive functions is enumerable, as we could also see by enumerating the systems E for Theorem II § 54. In particular, the primitive recursive functions of one variable are enumerable. Hence by Cantor's diagonal method (§ 2) they cannot comprise all the number-theoretic functions of one variable; and if

$$\varphi_0(a), \; \varphi_1(a), \; \varphi_2(a), \; \ldots$$

is any enumeration of them allowing repetitions (i.e. any infinite list of them in which each occurs at least once), then $\varphi_a(a)+1$ is a number-

theoretic function of one variable not in the enumeration, and so not primitive recursive. The enumerating function $\varphi(n, a)$ such that $\varphi(n, a) = \varphi_n(a)$ is a function of two variables which is not primitive recursive, since $\varphi_a(a)+1 = \varphi(a, a)+1$. This of course only establishes that number-theoretic functions $\varphi_a(a)+1$ and $\varphi(n, a)$ can be found which are not primitive recursive. What Péter did was to show that, for a suitable enumeration (with repetitions) of the primitive recursive functions of one variable, the enumerating function can be defined by a double recursion (besides applications of Schemata (I) — (V)).

EXAMPLE 1. Do double recursions lead to any predicates which are not primitive recursive? Yes, for $1 \dot{-} \varphi(a, a)$ takes only 0 and 1 as values, and cannot occur in the above enumeration, so it is the representing function of a predicate not primitive recursive. (Skolem 1944.)

Péter 1936 studies k-fold recursions for every positive integer k. These comprise primitive recursions for $k = 1$, double recursions for $k = 2$, and so on. She shows that, for each successive k, new functions are obtained. Functions definable using (besides explicit definition) recursions up to order k she calls "k-recursive". She shows that every 2-recursive function is definable by a single double recursion of the form

$$\text{(g)} \qquad \begin{cases} \varphi(0, b) = \varphi(n, 0) = 1, \\ \varphi(n', b') = \alpha(n, b, \varphi(n, \beta(n, b, \varphi(n', b))), \varphi(n', b)) \end{cases}$$

besides applications of Schemata (I) — (V); and similarly (with a scheme reducing to (g) for $k = 2$) for each $k \geq 2$.

EXAMPLE 2. To settle a point raised in § 45, suppose φ is 3-recursive but not 2-recursive, and ψ is 2-recursive but not 1-recursive, i.e. not primitive recursive. Then "if ψ is primitive recursive, then φ is primitive recursive" is vacuously true, but "φ is primitive recursive in ψ" is false, since that would make φ 2-recursive.

These subjects are treated in Péter's monograph 1951 (not available during the writing of the present book).

It is not to be expected that the k-fold recursions with finite k exhaust the possibilities for defining new functions by recursion. In 1950 Péter uses "transfinite recursions" (first employed by Ackermann 1940) to define new functions.

This brings us to the problem, whether we can characterize in any exact way the notion of any "recursion", or the class of all "recursive functions".

The examples (I) — (V), (a), (b), (e) (and others cited) of schemes of definition of a function which we have thus far agreed to call "recursions" possess two features: (i) They are expressed by equations in the manner which we analyzed formally (for (I) — (V) particularly) in § 54. (ii) They are definitions by mathematical induction, in one form or another, except in the trivial case when they are explicit definitions.

The characterization of all "recursive functions" was accomplished in the definition of 'general recursive function' by Gödel 1934, who built on a suggestion of Herbrand. This definition succeeds by a bold generalization, which consists in choosing Feature (i) by itself as the definition.

We say then that a function φ is *general recursive*, if there is a system E of equations which defines it recursively (§ 54, with $l = 0$).

This choice may seem unexpected, since the word "recursive" has its root in the verb "recur", and mathematical induction is our method for handling recurrent processes. The meaning of the choice is not that Feature (ii) will be absent from any particular recursion, but that it is transferred out of the definition itself to the application of the definition. To show by finitary means that a given scheme has Feature (i), except in trivial situations, one will presumably have to make use of mathematical induction somehow. But in defining the totality of general recursive functions, we forego the attempt to characterize in advance in what form the intuitive principle of induction must manifest itself. (By Gödel's theorem § 42 we know that the attempt at such a characterization by the formal number-theoretic system is incomplete.)

In stating the Herbrand-Gödel definition of general recursive function exactly, there is some latitude as to the details of the formalization, so that versions of the definition can be given which are equivalent to Gödel's but a bit simpler (cf. Kleene 1936, and 1943 § 8). The present version is that of Kleene 1943, except for inconsequential changes in R1 and R2 which simplify § 56 slightly, and the inclusion of functions of 0 variables in the treatment. (To relate the present treatment to Kleene 1943, we note: (1) The inclusion of functions of 0 variables does not alter the notion of general recursiveness for functions of $n > 0$ variables. For one can show that, if an auxiliary function letter h occurs as a term with 0 arguments in the assumption equations, all occurrences of this term may be changed to k(c), where k is a new function letter and c a new variable, without altering the class of the deducible equations containing only the principal function letter. After this: (2) One can show in a few lines that exactly the same equations of the form $f(\mathbf{x}_1, \ldots, \mathbf{x}_n) = \mathbf{x}$, where f is a function letter and $\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{x}$ are numerals, are deducible from given assumption equations by the present R1 and R2 as by the R1 and R2 of 1943; or with only a little more trouble one can carry out the treatment of §§ 54 and 56 with the R1 and R2 of 1943.)

A function φ is *general recursive in* functions $\psi_1, \ldots, \psi_l$, if there is a system E of equations which defines φ recursively from $\psi_1, \ldots, \psi_l$ (§ 54). This includes the definition of general recursive function as the case $l = 0$. For $l > 0$ (Kleene 1943), we are usually considering a scheme or functional $\varphi = \mathsf{F}(\psi_1, \ldots, \psi_l)$ (§ 47) which defines a number-theoretic function φ of n variables from $\psi_1, \ldots, \psi_l$, for any l number-theoretic functions $\psi_1, \ldots, \psi_l$ of $m_1, \ldots, m_l$ variables respectively, or any such functions subject to some stated restrictions. Then if the E can be given independently of $\psi_1, \ldots, \psi_l$ (for the fixed $n, l, m_1, \ldots, m_l$), we say that the scheme F is *general recursive*, or that φ *is general recursive uniformly in* $\psi_1, \ldots, \psi_l$. Since our treatment will always give uniformity in the ψ's (subject to any restrictions stated), we usually omit the word "uniformly" except for emphasis. (Unlike the primitive recursive case § 47, if the original scheme is for some restriction on $\psi_1, \ldots, \psi_l$, it is not implied that the scheme can necessarily be extended to a general recursive one defining a φ without restriction on the $\psi_1, \ldots, \psi_l$.)

Using the present terminology to restate the results of Lemmas IIa and IIe, we now have:

THEOREM II (second version). *If φ is defined from $\psi_1, \ldots, \psi_l$ by a succession of applications of general recursive schemes, then φ is general recursive in $\psi_1, \ldots, \psi_l$.*

In particular, Schemata (I) — (V) *are general recursive.* Hence: *If φ is primitive recursive in $\psi_1, \ldots, \psi_l$, it is general recursive in $\psi_1, \ldots, \psi_l$. Any primitive recursive scheme is general recursive. If φ is primitive recursive, it is general recursive.*

The definition of general recursiveness has been stated for the case that the function φ is already known, by intuitive use of the same equations which are formalized as the E, or by some other means. This anticipates our purpose of showing that various functions and schemes, known to us independently of the formalism of recursive functions, are general recursive (as we have just done for the primitive recursive functions and schemes). For the case that the φ is not previously known, we then have: A system E of equations defines recursively a function of n variables from $\psi_1, \ldots, \psi_l$, if for each n-tuple $x_1, \ldots, x_n$ of natural numbers, there is exactly one numeral $\mathbf{x}$ such that $E^{\psi_1 \ldots \psi_l}_{g_1 \ldots g_l}, E \vdash f(\mathbf{x}_1, \ldots, \mathbf{x}_n) = \mathbf{x}$, where f is the principal function letter of E, and $g_1, \ldots, g_l$ are the given