The Journal of Symbolic Logic
Volume 2, Number 4, December 1937

COMPUTABILITY AND λ-DEFINABILITY

A. M. TURING

Several definitions have been given to express an exact meaning corresponding to the intuitive idea of 'effective calculability' as applied for instance to functions of positive integers. The purpose of the present paper is to show that the computable[1] functions introduced by the author are identical with the λ-definable[2] functions of Church and the general recursive[3] functions due to Herbrand and Gödel and developed by Kleene. It is shown that every λ-definable function is computable and that every computable function is general recursive. There is a modified form of λ-definability, known as λ-K-definability, and it turns out to be natural to put the proof that every λ-definable function is computable in the form of a proof that every λ-K-definable function is computable; that every λ-definable function is λ-K-definable is trivial. If these results are taken in conjunction with an already available[4] proof that every general recursive function is λ-definable we shall have the required equivalence of computability with λ-definability and incidentally a new proof of the equivalence of λ-definability and λ-K-definability.

A definition of what is meant by a computable function cannot be given satisfactorily in a short space. I therefore refer the reader to *Computable* pp. 230–235 and p. 254. The proof that computability implies recursiveness requires no more knowledge of computable functions than the ideas underlying the definition: the technical details are recalled in §5. On the other hand in proving that the λ-K-definable functions are computable it is assumed that the reader is familiar with the methods of *Computable* pp. 235–239.

The identification of 'effectively calculable' functions with computable functions is possibly more convincing than an identification with the λ-definable or general recursive functions. For those who take this view the formal proof of equivalence provides a justification for Church's calculus, and allows the 'machines' which generate computable functions to be replaced by the more convenient λ-definitions.

Received September 11, 1937.

[1] A. M. Turing, *On computable numbers, with an application to the Entscheidungsproblem*, ***Proceedings of the London Mathematical Society***, ser. 2, vol. 42 (1936–7), pp. 230–265, quoted here as *Computable*. A similar definition was given by E. L. Post, *Finite combinatory processes—formulation 1*, this Journal, vol. 1 (1936), pp. 103–105.

[2] Alonzo Church, *An unsolvable problem of elementary number theory*, ***American journal of mathematics***, vol. 58 (1936), pp. 345–363, quoted here as *Unsolvable*.

[3] S. C. Kleene, *General recursive functions of natural numbers*, ***Mathematische Annalen***, vol. 112 (1935–6), pp. 727–742. A definition of general recursiveness is also to be found in *Unsolvable* pp. 350–351.

[4] S. C. Kleene, *λ-definability and recursiveness*, ***Duke mathematical journal***, vol. 2 (1936), pp. 340–353.

1. Definition of λ-*K*-definability. In this section the notion of λ-*K*-definability is introduced in a form suitable for handling with machines. There will be three differences from the normal, in addition to that which distinguishes λ-*K*-definability from λ-definability. One consists in using only one kind [ ] of bracket istead of three, { }, ( ), [ ]; another is that x, x ¹, x ¹ ¹, · · · are used as variables instead of an indefinite infinite list of the single symbols, and the third is a change in the form of condition (ii) of immediate transformability, not affecting the definition of convertibility except in form.

There are five symbols which occur in the formulae of the conversion calculus. They are λ, x, ¹, [and] . A sequence of symbols consisting of x followed by ¹ repeated any number (possibly 0) of times is called a *variable*. *Properly-formed formulae* are a class of sequences of symbols which includes all variables. Also if ***M*** and ***N*** are[6] properly-formed formulae, then [***M***][***N***] (i.e. the sequence consisting of [followed by ***M*** then by] , [and the sequence ***N***, and finally by]) is a properly-formed formula. If ***M*** is a properly-formed formula and ***V*** is a variable, then λ***V***[***M***] is a properly-formed formula. If any sequence is a properly-formed formula it must follow that it is so from what has already been said.

A properly-formed formula ***M*** will be said to be *immediately transformable* into ***N*** if either:

(i) ***M*** is of the form λ***V***[***X***] and ***N*** is λ***U***[***Y***] where ***Y*** is obtained from ***X*** by replacing the variable ***V*** by the variable ***U*** throughout, provided that ***U*** does not occur in ***X***.

(ii) ***M*** is of the form [λ***V***[***X***]][***Y***] where ***V*** is a variable and ***N*** is obtained by substituting ***Y*** for ***V*** throughout ***X***. This is to be subject to the restriction that if ***W*** be either ***V*** or a variable occurring in ***Y***, then λ***W*** must not occur in ***X***.

(iii) ***N*** is immediately transformable into ***M*** by (ii).

A will be said to be immediately convertible into ***B*** if ***A*** is immediately transformable into ***B*** or if ***A*** is of the form ***X***[***L***]***Y*** and ***B*** is ***X***[***M***]***Y*** where ***L*** is immediately transformable into ***M***. Either ***X*** or ***Y*** may be void. ***A*** is *convertible* to ***B*** (***A*** conv ***B***) if there is a finite sequence of properly-formed formulae, beginning with ***A*** and terminating with ***B***, each immediately convertible into the preceding.

The formulae,

λx[λx ¹ [x ¹]] (abbreviated to 0),
λx[λx ¹ [[x][x ¹]]] (abbreviated to 1),
λx[λx ¹ [[x][[x][x ¹]]]] (abbreviated to 2), etc.,

represent the natural numbers. If *n* represents a natural number then the next natural number is represented by a formula which is convertible to [*S*][*n*] where *S* is

λx ¹ ¹ [λx[λx ¹ [[x][[[x ¹ ¹][x]][x ¹]]]]].

A function *f*(*n*) of the natural numbers, taking natural numbers as values will be said to be λ-*K*-definable if there is a formula ***F*** such that [***F***][*n*] is con-

[6] Heavy type capitals are used to stand for variable or undetermined sequences of symbols. In expressions involving brackets and heavy type letters it is to be understood that the possible substitutions of sequences of symbols for these letters is to be subject to the restriction that the pairing of the explicitly shown brackets is unaltered by the substitution; thus in ***X***[***L***]***Y*** the number of occurrences of [in ***L*** must equal the number of occurrences of] .

vertible to the formula representing $f(n)$ if **n** is the formula representing n. The formula $[F][n]$ can never be convertible to two formulae representing different natural numbers, for it has been shown[6] that if two properly-formed formulae are in normal form (i.e., have no parts of the form $[\lambda V[M]][N]$) and are convertible into one another, then the conversion can be carried out by the use of (i) only. The formulae representing the natural numbers are in normal form and the formulae representing two different natural numbers are certainly not convertible into one another by the use of (i) alone.

2. Abbreviations. A number of abbreviations of the same character as those in *Computable* (pp. 235–239) are introduced here. They will be applied in connection with the calculus of conversion, but are necessary for other purposes, e.g. for carrying out the processes of any ordinary formal logic with machines. The abbreviations in *Computable* are taken as known.

'The sequence of symbols marked with α (followed by α)' will be abbreviated to $S(\alpha)$ in the explanations. Sequences are normally identified by the way they are marked, and are as it were lost when their marks are erased.

In the tables $\mathfrak{a}$ will be used as a name for the symbol 'blank.'

$\mathfrak{pem}(\mathfrak{A}, \alpha, \beta)$		$\mathfrak{pe}(\mathfrak{pem}_1, \alpha)$
$\mathfrak{pem}_1$	$R, P\beta$	$\mathfrak{A}$

$\mathfrak{pem}_1$ here stands for $\mathfrak{pem}_1(\mathfrak{A}, \alpha, \beta)$ and similar abbreviations must be understood throughout.

$\mathfrak{pem}(\mathfrak{A}, \alpha, \beta)$. The machine prints α at the end of the sequence of symbols on F-squares and marks it with β. $\rightarrow\mathfrak{A}$.

The tables for $\mathfrak{crm}(\mathfrak{B}, \gamma, \beta)$ and $\mathfrak{cem}(\mathfrak{B}, \gamma, \beta)$ are to be obtained from those for $\mathfrak{cr}(\mathfrak{B}, \gamma)$ and $\mathfrak{ce}(\mathfrak{B}, \gamma)$ by replacing $\mathfrak{pe}(\mathfrak{A}, \alpha)$ by $\mathfrak{pem}(\mathfrak{A}, \alpha, \beta)$ throughout.

$\mathfrak{cpr}(\mathfrak{A}, \mathfrak{E}, \alpha, \beta)$	$\mathfrak{cp}(\mathfrak{cpr}_1, \mathfrak{cpr}_2, \mathfrak{cpr}_3, \alpha, \beta)$
$\mathfrak{cpr}_1$	$\mathfrak{re}(\mathfrak{re}(\mathfrak{cpr}, \mathfrak{b}, \beta, b), \mathfrak{b}, \alpha, a)$
$\mathfrak{cpr}_2$	$\mathfrak{re}(\mathfrak{re}(\mathfrak{E}, b, \beta), a, \alpha)$
$\mathfrak{cpr}_3$	$\mathfrak{re}(\mathfrak{re}(\mathfrak{A}, b, \beta), a, \alpha)$

$\mathfrak{cpr}(\mathfrak{A}, \mathfrak{E}, \alpha, \beta)$. The machine compares $S(\alpha)$ with $S(\beta)$. $\rightarrow\mathfrak{A}$ if they are alike; $\rightarrow\mathfrak{E}$ otherwise. No erasures are made.

The letters a, b occurring in the table for $\mathfrak{cpr}$ should not be used elsewhere in any machine whose table involves $\mathfrak{cpr}$. This can be made automatic by using $a_{\mathfrak{cpr}}$ and $b_{\mathfrak{cpr}}$, say, instead of a and b. We shall however write a and b and understand them to mean $a_{\mathfrak{cpr}}$ and $b_{\mathfrak{cpr}}$. The same applies for the letters $a, \cdots, z$ in all such tables.

$\mathfrak{k}(\mathfrak{A}, \gamma)$	γ	L	$\mathfrak{A}$
	not γ	R, R	$\mathfrak{k}$
$\mathfrak{bk}(\mathfrak{A}, \alpha, \beta, \gamma, \delta)$		E, Pa	$\mathfrak{k}(\mathfrak{bk}_1, \gamma)$

[6] Alonzo Church and J. B. Rosser, *Some properties of conversion*, ***Transactions of the American Mathematical Society***, vol. 39 (1936), pp. 472–482. The result used here is Theorem 1 Corollary 2 as extended to the modified conversion on p. 482.

$\mathfrak{bk}_1$	α	R, E, Pb	$\mathfrak{k}(\mathfrak{bk}_2, \gamma)$
	β	L	$\mathfrak{bk}_2$
	others	R, R, R	$\mathfrak{k}(\mathfrak{bk}_1, \gamma)$
$\mathfrak{bk}_2$	β	R, E, Pb	$\mathfrak{f}(\mathfrak{bk}, \mathfrak{b}, a)$
	not β	R, R, R	$\mathfrak{k}(\mathfrak{bk}_2, \gamma)$
$\mathfrak{bk}_3$	γ or b	$E, P\delta, L, L$	$\mathfrak{bk}_3$
	α	$E, P\gamma$	$\mathfrak{A}$
	others	L, L	$\mathfrak{bk}_3$

$\mathfrak{bk}(\mathfrak{A}, \alpha, \beta, \gamma, \delta)$. This describes the process of finding the partner of a bracket. If α and β are regarded as left and right brackets, then if the machine takes up the internal configuration $\mathfrak{bk}$ when scanning a square next on the right of an α it will find the partner of this α in the sequence $S(\gamma)$, and will mark the part of $S(\gamma)$ which is between the brackets with δ (instead of γ). The final internal configuration is $\mathfrak{A}$ and the scanned square is that which was scanned when the internal configuration $\mathfrak{bk}$ was first taken up.

$\mathfrak{sb}(\mathfrak{A}, \alpha, \beta, \gamma, \delta, \epsilon)$			$\mathfrak{f}'(\mathfrak{sb}_1, \mathfrak{crm}(\mathfrak{re}(\mathfrak{re}(\mathfrak{sb}, a, j), b, \beta), \gamma, \epsilon), \beta)$
$\mathfrak{sb}_1$	σ	R, E, Pb	$\mathfrak{sb}_2(\mathfrak{A}, \alpha, \beta, \gamma, \delta, \epsilon, \sigma)$
$\mathfrak{sb}_2$			$\mathfrak{f}'(\mathfrak{sb}_3, \mathfrak{crm}(\mathfrak{re}(\mathfrak{re}(\mathfrak{re}(\mathfrak{A}, b, \beta), j, \alpha), a, \alpha), a, \delta), \alpha)$
$\mathfrak{sb}_3$	σ	R, E, Pa	$\mathfrak{sb}$
	not σ	R, E, Pa	$\mathfrak{re}(\mathfrak{f}'(\mathfrak{sb}_4, \mathfrak{b}, a), b, \beta)$
$\mathfrak{sb}_4$	τ	R, E, Pj	$\mathfrak{re}(\mathfrak{pem}(\mathfrak{sb}, \tau, \delta), a, \alpha)$
$\mathfrak{sub}(\mathfrak{A}, \alpha, \beta, \gamma, \delta)$			$\mathfrak{sb}(\mathfrak{A}, \alpha, \beta, \gamma, \delta, \delta)$
$\mathfrak{bt}(\mathfrak{A}, \mathfrak{B}, \alpha, \beta)$			$\mathfrak{pem}(\mathfrak{sb}(\mathfrak{f}(\mathfrak{e}(\mathfrak{A}, d), \mathfrak{B}, d), \alpha, \beta, p, \mathfrak{a}, d), r, p)$

$\mathfrak{sub}(\mathfrak{A}, \alpha, \beta, \gamma, \delta)$. $S(\gamma)$ is substituted for $S(\beta)$ throughout $S(\alpha)$. The result is copied down and marked with δ. $\rightarrow\mathfrak{A}$.

$\mathfrak{bt}(\mathfrak{A}, \mathfrak{B}, \alpha, \beta)$. It is determined whether the sequence $S(\beta)$ occurs in $S(\alpha)$. $\rightarrow\mathfrak{A}$ if it does; $\rightarrow\mathfrak{B}$ otherwise.

The tables which follow are particularly important in all cases where an enumeration of all possible results of operations of given types is required. The enumeration may be carried out by regarding the operation as determined by a number of choices, each between two possibilities, L and M say. Each possible sequence of operations is then associated with a finite sequence of letters L and M. These sequences can easily be enumerated. The method used here is to replace L by 0, each M by 1, follow the whole by 1, reverse the order and regard the result as the binary Arabic numeral corresponding to the given sequence. Thus the first few sequences (beginning with the one associated with 1) are: the null sequence, L, M, LL, ML, LM, MM, LLL, MLL, LML, MML. In the general table below ζ and η are used instead of L and M.

$\mathfrak{add}(\mathfrak{A}, \alpha, \zeta, \eta)$			$\mathfrak{f}'(\mathfrak{add}_1, \mathfrak{pem}(\mathfrak{add}_2, \zeta, a), \alpha)$
$\mathfrak{add}_1$	η	R, E	$\mathfrak{pem}(\mathfrak{add}, \zeta, a)$
	ζ	R, E	$\mathfrak{pem}(\mathfrak{add}_2, \eta, a)$
$\mathfrak{add}_2$			$\mathfrak{cem}(\mathfrak{re}(\mathfrak{A}, a, \alpha), \alpha, a)$

$\mathfrak{adb}(\mathfrak{A}, \alpha, \zeta, \eta)$. The sequence S($\alpha$) consisting of letters ζ and η only is transformed into the next sequence. $\rightarrow\mathfrak{A}$.

$\mathfrak{ch}(\mathfrak{A}, \mathfrak{B}, \mathfrak{C}, \alpha, \zeta, \eta)$			$\mathfrak{f}'(\mathfrak{ch}_1, \mathfrak{re}(\mathfrak{C}, b, \alpha), \alpha)$
$\mathfrak{ch}_1$	ζ	R, E, Pb	$\mathfrak{A}$
	η	R, E, Pb	$\mathfrak{B}$

$\mathfrak{ch}(\mathfrak{A}, \mathfrak{B}, \mathfrak{C}, \alpha, \zeta, \eta)$ is an internal configuration which is taken up when a choice has to be made. S(α) is the sequence of letters ζ and η determining the choices. $\rightarrow\mathfrak{A}$ if the first unused letter is ζ; $\rightarrow\mathfrak{B}$ if it is η: it is then indicated that this ζ or η has been used by replacing its mark by b. When the whole sequence has been used up these marks are replaced by α again and $\rightarrow\mathfrak{C}$.

$\mathfrak{cch}(\mathfrak{A}, \mathfrak{B}, \mathfrak{C}, \alpha, \zeta, \eta)$		R	$\mathfrak{cch}_1$
$\mathfrak{cch}_1$	σ	E, Pa	$\mathfrak{cch}_2(\mathfrak{A}, \mathfrak{B}, \mathfrak{C}, \alpha, \zeta, \eta, \sigma)$
$\mathfrak{cch}_2$			$\mathfrak{ch}(\mathfrak{f}(\mathfrak{cch}_3, \mathfrak{b}, a), \mathfrak{f}(\mathfrak{cch}_4, \mathfrak{b}, a), \mathfrak{C}, \alpha, \zeta, \eta)$
$\mathfrak{cch}_3$		$E, P\sigma, L$	$\mathfrak{A}$
$\mathfrak{cch}_4$		$E, P\sigma, L$	$\mathfrak{B}$

$\mathfrak{cch}(\mathfrak{A}, \mathfrak{B}, \mathfrak{C}, \alpha, \zeta, \eta)$. This differs from $\mathfrak{ch}$ in that the internal configurations $\mathfrak{A}$ and $\mathfrak{B}$ are taken up when the same square is scanned as that which was scanned when the internal configuration $\mathfrak{cch}$ was first reached, provided that this was an F-square.

3. Mechanical Conversion. We are now in a position to show how the conversion process can be carried out mechanically. It will be necessary to be able to perform all of the three kinds of immediate transformation. (iii) can be done most easily if we can enumerate properly-formed formulae. It is principally for this purpose that we introduce the table for $\mathfrak{pff}(\mathfrak{A}, \alpha)$.

$\mathfrak{funf}(\mathfrak{A}, \alpha, \beta, \gamma)$	$\mathfrak{pem}(\mathfrak{crm}(\mathfrak{pem}_2(\mathfrak{crm}(\mathfrak{pem}(\mathfrak{A},], \gamma), \beta, \gamma),], [, \gamma), \alpha, \gamma), [, \gamma)$

$\mathfrak{funf}(\mathfrak{A}, \alpha, \beta, \gamma)$. [S($\alpha$)] [S($\beta$)] is written at the end and marked with γ. $\rightarrow\mathfrak{A}$.

$\mathfrak{ch}(\mathfrak{A}, \mathfrak{B}, \mathfrak{C}, \theta)$	$\mathfrak{ch}(\mathfrak{A}, \mathfrak{B}, \mathfrak{C}, \theta, \text{L}, \text{M})$
$\mathfrak{cch}(\mathfrak{A}, \mathfrak{B}, \mathfrak{C}, \theta)$	$\mathfrak{cch}(\mathfrak{A}, \mathfrak{B}, \mathfrak{C}, \theta, \text{L}, \text{M})$

The choices will be determined by a sequence made up of letters L and M.

$\mathfrak{pff}(\mathfrak{A}, \mathfrak{C}, \alpha, \theta)$			$\mathfrak{pe}_6(\mathfrak{c}(\mathfrak{h}(\mathfrak{af}, \mathfrak{pff}_1, \mathfrak{C}, \theta), :, ;, \text{x}, ;, \text{x})$
$\mathfrak{pff}_1$			$\mathfrak{q}(\mathfrak{pff}_2, :)$
$\mathfrak{pff}_2$	;	R, R	$\mathfrak{cch}(\mathfrak{pff}_3, \mathfrak{pff}_2, \mathfrak{C}, \theta)$
	$\mathfrak{a}$		$\mathfrak{af}$
	others	R, R	$\mathfrak{pff}_2$
$\mathfrak{pff}_3$	;		$\mathfrak{pff}_4$
	$\mathfrak{a}$		$\mathfrak{af}$
	others	R, Pa, R	$\mathfrak{pff}_3$
$\mathfrak{pff}_4$	;	R, R	$\mathfrak{cch}(\mathfrak{pff}_5, \mathfrak{pff}_4, \mathfrak{C}, \theta)$
	$\mathfrak{a}$		$\mathfrak{af}$
	others	R, R	$\mathfrak{pff}_4$
$\mathfrak{pff}_5$	; or $\mathfrak{a}$		$\mathfrak{ar}$
	others	R, Pb, R	$\mathfrak{pff}_5$

ar			ch(ne, ch(comp, ab, $\mathfrak{C}$, θ), $\mathfrak{C}$, θ)
ne			pe$_2$(ne$_1$, ;, x)
ne$_1$			ch(pe(ne$_1$, $^{\prime}$), af, $\mathfrak{C}$, θ)
comp			pe(funf(af, a, b, $\mathfrak{a}$), ;)
ab			pe$_3$(ab$_1$, ;, λ, x)
ab$_1$			ch(pe(ab$_1$, $^{\prime}$), ab$_2$, $\mathfrak{C}$, θ)
ab$_2$			pe(ce(pe(af,]), a), [)
af			e(e(ch(fin, pff$_1$, $\mathfrak{C}$, θ), a), b)
fin			q(r(r(fin$_1$)), ;)
fin$_1$	not $\mathfrak{a}$	R, Pa, R	fin$_1$
	$\mathfrak{a}$		$\mathfrak{A}$

pff($\mathfrak{A}$, $\mathfrak{C}$, α, θ). A properly-formed formula is chosen, written down at the end and marked with α. $\rightarrow\mathfrak{A}$. This is done by writing down successive properly-formed formulae separated by semicolons, and obtaining others from them by abstraction (i.e., the process by which $\lambda V[M]$ is obtained from M), by application of a function to its argument (i.e., obtaining $[M][N]$ from M and N), and by writing down new variables. Before writing down a new formula we have the alternative of taking the last formula as the result of the calculation. In this case the internal configuration fin is taken up. If a new formula is to be constructed then two of the old formulae are chosen and marked with a and b: then one of the internal configurations ab, comp, ne is chosen and the new formula is correspondingly $\lambda V[\mathrm{S}(a)]$, $[\mathrm{S}(a)][\mathrm{S}(b)]$, or V, where V is a new variable. The whole of the work is separated by a colon from the symbols which were on the tape previously. The meanings of pe$_3$ and pe$_5$ are analogous to pe$_2$.

The occurrence of λ in this table is of course as a symbol of the conversion calculus, not as a variable machine symbol.

The immediate transformations (i) and (ii) are described next.

va($\mathfrak{A}$, $\mathfrak{C}$, α, β, θ)			f$'$(va$_1$, $\mathfrak{A}$, α)
va$_1$	λ	R, E, Pa	f$'$(va$_2$, b, α)
	others		crm($\mathfrak{A}$, α, β)
va$_2$	x or $^{\prime}$	R, E, Pb	f$'$(va$_2$, b, α)
	others	R	bf(pe(va$_3$, x), [,], α, c)
va$_3$		R, Pd	ch(pe(va$_3$, $^{\prime}$), bt(va$_4$, va$_5$, c, d), $\mathfrak{C}$, θ)
va$_4$			re(re(re(crm(e($\mathfrak{A}$, d), α, β), b, α), a, α), c, α)
va$_5$			crm(re(re(re(va$_6$, a, α), b, α), c, α), b, f)
va$_6$			sub(e(e($\mathfrak{A}$, d), f), α, f, d, β)

va($\mathfrak{A}$, $\mathfrak{C}$, α, β, θ). An immediate transformation (i) is chosen, and if permissible is carried out on $\mathrm{S}(\alpha)$, the result being marked with β. If the chosen transformation is not permissible then $\mathrm{S}(\beta)$ is identical with $\mathrm{S}(\alpha)$. $\rightarrow\mathfrak{A}$.

reb($\mathfrak{A}$, α, β)	f$'$(reb$_1$, reb$_{13}$, α)

$\mathfrak{red}_1$	[	R	$\mathfrak{bk}(\mathfrak{red}_2, [,], \alpha, c)$
	not [		$\mathfrak{red}_{13}$
$\mathfrak{red}_2$		E, Pf	$\mathfrak{re}(\mathfrak{f}(\mathfrak{red}_3, \mathfrak{b}, \alpha), \mathfrak{b}, \alpha, f)$
$\mathfrak{red}_3$			$\mathfrak{bk}(\mathfrak{red}_4, [,], \alpha, d)$
$\mathfrak{red}_4$			$\mathfrak{f}'(\mathfrak{red}_5, \mathfrak{b}, c)$
$\mathfrak{red}_5$	λ	R, E, Pf	$\mathfrak{f}'(\mathfrak{red}_6, \mathfrak{b}, c)$
	not λ		$\mathfrak{red}_{13}$
$\mathfrak{red}_6$	x or ′	R, E, Pg	$\mathfrak{f}'(\mathfrak{red}_6, \mathfrak{b}, c)$
	[	R, E, Pf	$\mathfrak{q}(\mathfrak{red}_7, c)$
$\mathfrak{red}_7$		E, Pf	$\mathfrak{red}_8$
$\mathfrak{red}_8$			$\mathfrak{f}'(\mathfrak{red}_9, \mathfrak{red}_{10}, c)$
$\mathfrak{red}_9$	λ	R, E, Pk	$\mathfrak{f}'(\mathfrak{red}_{11}, \mathfrak{b}, c)$
	not λ	R, E, Pk	$\mathfrak{red}_8$
$\mathfrak{red}_{11}$	x or ′	R, E, Pj	$\mathfrak{f}'(\mathfrak{red}_{11}, \mathfrak{b}, c)$
	[		$\mathfrak{cpr}(\mathfrak{red}_{13}, \mathfrak{red}_{12}, j, g)$
$\mathfrak{red}_{12}$			$\mathfrak{dt}(\mathfrak{red}_{13}, \mathfrak{re}(\mathfrak{red}_8, j, k), d, j)$
$\mathfrak{red}_{10}$			$\mathfrak{sub}(\mathfrak{re}(\mathfrak{re}(\mathfrak{re}(\mathfrak{re}(\mathfrak{A}, d, \alpha), f, \alpha), k, \alpha), g, \alpha)k, g, d, \beta)$
$\mathfrak{red}_{13}$			$\mathfrak{re}(\mathfrak{re}(\mathfrak{re}(\mathfrak{re}(\mathfrak{re}(\mathfrak{re}(\mathfrak{crm}(\mathfrak{A}, \alpha, \beta), d, \alpha), g, \alpha), c, \alpha), f, \alpha), k, \alpha), j, \alpha)$

$\mathfrak{red}(\mathfrak{A}, \alpha, \beta)$. An immediate transformation (ii) is carried out on $S(\alpha)$, supposing that $S(\alpha)$ is properly-formed. The result is marked with β. $\rightarrow\mathfrak{A}$. If the transformation is not possible or permissible $S(\beta)$ is identical with $S(\alpha)$. Considerable use is made of the hypothesis that $S(\alpha)$ is properly-formed. Thus if its first symbol is [then it must be of form $[L][N]$ and if in addition the second symbol is λ then it is of form $[\lambda V[M]][N]$. The internal configuration $\mathfrak{red}_8$ is never reached unless $S(\alpha)$ is of this form, and in that case it first occurs when V has been marked with g, M with c, and N with d, the remaining symbols of what was $S(\alpha)$ being now marked with α or f. It is then determined whether the immediate transformation (ii) is permissible: if it is then $\mathfrak{red}_{10}$ is taken up and the substitution carried out.

$\mathfrak{imc}(\mathfrak{A}, \mathfrak{C}, \alpha, \beta, \theta)$			$\mathfrak{ch}(\mathfrak{f}'(\mathfrak{imc}_1, \mathfrak{imc}_2, \alpha), \mathfrak{re}(\mathfrak{imc}_4, \alpha, a), \mathfrak{C}, \theta)$
$\mathfrak{imc}_1$	[	R, E, Pc	$\mathfrak{ch}(\mathfrak{imc}, \mathfrak{imc}_3, \mathfrak{C}, \theta)$
	not [	R, E, Pc	$\mathfrak{imc}$
$\mathfrak{imc}_2$			$\mathfrak{re}(\mathfrak{crm}(\mathfrak{A}, \alpha, \beta), c, \alpha)$
$\mathfrak{imc}_3$			$\mathfrak{q}(\mathfrak{bk}(\mathfrak{imc}_4, [,], \alpha, a), c)$
$\mathfrak{imc}_4$			$\mathfrak{ch}(\mathfrak{vc}, \mathfrak{ch}(\mathfrak{rc}, \mathfrak{ex}, \mathfrak{C}, \theta), \mathfrak{C}, \theta)$
$\mathfrak{vc}$			$\mathfrak{va}(\mathfrak{imc}_5, \mathfrak{C}, a, b, \theta)$
$\mathfrak{rc}$			$\mathfrak{red}(\mathfrak{imc}_5, a, b)$
$\mathfrak{ex}$			$\mathfrak{pff}(\mathfrak{red}(\mathfrak{ex}_1, b, d), \mathfrak{C}, b, \theta)$

$\mathfrak{ex}_1$	cpr(e(imc$_6$, d), e(e(crm(imc$_6$, a, b), d), b), d, a)
imc$_6$	crm(crm(crm(re(re(e($\mathfrak{A}$, b), c, α), a, α), α, β), b, β), c, β)

imc($\mathfrak{A}$, $\mathfrak{C}$, α, β, θ). An immediate conversion is chosen and performed on S(α). The result is marked with β. $\rightarrow\mathfrak{A}$.

conv($\mathfrak{A}$, α, β, θ)			pe(crm(conv$_1$, α, d), .)
conv$_1$			ch(imc(conv$_2$, au, d, f, θ), re($\mathfrak{A}$, d, β), au, θ)
conv$_2$			e(re(conv$_1$, f, d), d)
au			q(au$_1$, .)
au$_1$	$\mathfrak{a}$		crm($\mathfrak{A}$, α, β)
	not $\mathfrak{a}$	R, E, R	au$_1$

conv($\mathfrak{A}$, α, β, θ). A conversion is chosen and performed on S(α). The result is marked with β. $\rightarrow\mathfrak{A}$. The sequence determining the choices is S(θ). If it should happen that this sequence is exhausted before the conversion is completed then the final formula is the same as the original, i.e. S(α). The half finished conversion work is effectively removed from the tape by erasing the marks.

4. Computability of λ-*K*-definable functions. It is now comparatively simple to show that a λ-*K*-definable function is computable, i.e., that[7] if $f(n)$ is λ-*K*-definable then the sequence γ_f in which there are $f(n)$ figures 1 between the nth and the $(n+1)$th 0, and $f(0)$ figures before the first 0, is computable.

To simplify the table for the machine which computes γ_f we use the abbreviation $\mathfrak{Wr}$($\mathfrak{A}$, M, α) for an internal configuration starting from which the machine writes the sequence M of symbols at the end, marking it with α and finishing in the internal configuration $\mathfrak{A}$. Thus the table for $\mathfrak{Wr}$($\mathfrak{A}$, λx′, α) would be:

$\mathfrak{Wr}$($\mathfrak{A}$, λx′, α)		pe($\mathfrak{Wr}_1$, $\mathfrak{a}$)
$\mathfrak{Wr}_1$	$P\lambda, R, P\alpha, R, P\mathrm{x}, R, P\alpha, R, P', R, P\alpha$	$\mathfrak{A}$

We use one more skeleton table:

pls($\mathfrak{A}$, α, β)	funk(e(re($\mathfrak{A}$, a, α), α), β, α, a)

If F is the formula which λ-*K*-defines $f(n)$ then the table for the machine which computes γ_f is:

b	$P\text{ə}, R, P\text{ə}$	$\mathfrak{Wr}$(b$_1$, F, h)
b$_1$		$\mathfrak{Wr}$(b$_2$, λx[λx′ [x′]], i)
b$_2$		crm(b$_3$, i, k)
b$_3$		$\mathfrak{Wr}$(ba, λx″ [λx[λx′ [[x][[[x″][x]][x′]]]]], u)
ba		funk(cn$_1$, h, k, v)
cn		add(cn$_1$, s, L, M)
cn$_1$		crm(cn$_2$, i, d)
cn$_2$		ch(re(cn$_3$, d, m), pls(cn$_2$, d, u), cn$_6$, s)
cn$_3$		conv(cn$_4$, v, w, s)
cn$_4$		cpr(cn$_5$, cn$_6$, w, m)

[7] *Computable* p. 254.

$\mathfrak{cn}_6$			$\mathfrak{e}(\mathfrak{e}(\mathfrak{e}(\mathfrak{cn}_{10}, w), m), d)$
$\mathfrak{cn}_{10}$			$\mathfrak{ch}(\mathfrak{cn}_{10}, \mathfrak{cn}_{10}, \mathfrak{cn}, s)$
$\mathfrak{cn}_5$			$\mathfrak{q}(\mathfrak{cn}_7, m)$
$\mathfrak{cn}_7$		E	$\mathfrak{q}(\mathfrak{cn}_8, m)$
$\mathfrak{cn}_8$		E	$\mathfrak{q}(\mathfrak{l}(\mathfrak{cn}_9), m)$
$\mathfrak{cn}_9$	]	R, E	$\mathfrak{pem}(\mathfrak{q}(\mathfrak{l}(\mathfrak{cn}_9), m), 1, a)$
	not]		$\mathfrak{pem}(\mathfrak{e}(\mathfrak{e}(\mathfrak{e}(\mathfrak{e}(\mathfrak{ba}_1, s), v), w), m), 0, a)$
$\mathfrak{ba}_1$			$\mathfrak{pls}(\mathfrak{ba}, k, u)$

When the machine reaches the internal configuration $\mathfrak{ba}$ for the $(n+1)$th time $(n \geqq 0)$ the tape bears the formula F marked with h, the formula n representing the natural number n (or rather a formula convertible into it) marked with k, 0 marked with i, and S marked with u. A formula convertible into one representing some natural number r is then chosen and marked with m. This brings us to the internal configuration $\mathfrak{cn}_5$. A conversion is then chosen and performed on $\mathrm{S}(v)$, i.e. on $[F][n]$. The result is marked with w and compared with $\mathrm{S}(m)$. If they are not alike the letters w, m are erased and we go back to $\mathfrak{cn}_1$ after transforming the sequence $\mathrm{S}(s)$ which determines the choices into the next sequence. If they are alike then 1 is written at the end repeated r times followed by 0, all of which is marked with a. In order to have the correct number of figures we make use of the fact that the number of brackets occurring consecutively at the end of $\mathrm{S}(m)$ is $r+2$. The machine is back in the internal configuration $\mathfrak{ba}$ as soon as $\mathrm{S}(k)$ has been changed to $[S][\mathrm{S}(k)]$.

No attempt is being made to give a formal proof that this machine has the properties claimed for it. Such a formal proof is not possible unless the ideas of substitution and so forth occurring in the definition of conversion are formally defined, and the best form of such a definition is possibly in terms of machines.

If $f(n)$ $(n \geqq 1)$ is λ-definable, i.e. if F is well-formed (*Unsolvable* p. 346), then the present argument shows also that $f(n)$ is then computable in the sense that a function $g(n)$ of positive integers is computable if there is a computable sequence with $g(n)$ figures 1 between the nth and $(n+1)$th figure 0.

5. Recursiveness of computable functions. It will now be shown that every computable function $f(n)$ of the natural numbers, taking natural numbers as values, is general recursive. We shall in fact find primitive recursive functions $j(x)$, $\phi(x)$ such that if $\xi(x)$ is the $(x+1)$th $(x=0, 1, 2, \cdots)$ natural number y for which $j(y)=0$, then $f(x)$ is given by

$$f(x) = \phi(\xi(x)).$$

It is easily seen that such a function is general recursive (cf. *Unsolvable* p. 353); also it can easily be brought into the form,[8]

$$f(x) = \phi(\epsilon y[i(x, y) = 0])$$

[8] This may be done by defining $i(x, y)$ as follows:

$$\begin{aligned} e(0) &= 0, \\ e(S(x)) &= S(e(x)) && \text{if } j(x) = 0, \\ &= e(x) && \text{otherwise}, \\ i(x, y) &= \mathrm{Max}\ (0, x - e(y)), \end{aligned}$$

$S(x)$ as usual meaning $x+1$.

(where $\epsilon y[i(x, y)=0]$ means 'the least natural number y for which $i(x, y)=0$,' and $i(x, y)$ is primitive recursive) which plays a central part[9] in the theory of general recursive functions. It would be slightly simpler to set up recursion equations for $f(x)$ but in that case it would be necessary to show that they were consistent; this is avoided by confining ourselves to primitive recursions (whose consistency is not likely to be doubted) except at the step from $j(x)$ to $\xi(x)$.

We are given the description of a machine which computes $f(x)$. The machine writes down sysmbols on a tape: amongst these symbols occur figures 0 and 1. The number of figures 1 between the nth and the $(n+1)$th figure 0 is $f(n)$. At any moment there is one of the symbols on the tape which is to be distinguished from the others and is called the 'scanned symbol.' The state (complete configuration) of the system at any moment is described by the sequence of symbols on the tape, with an indication as to which of them is scanned, and the internal configuration (m-configuration in *Computable*) of the machine. As names for the symbols we take $S_0, S_1, \cdots, S_{N-1}$ and for the internal configurations $q_1, q_2, \cdots, q_R$. Certain of these are names of definite symbols and internal configurations independent of the machine; in fact,

S_0 always stands for 'blank,'
S_1 always stands for 0,
S_2 always stands for 1,
q_1 always stands for the initial internal configuration.

If at any time there is the sequence

$$S_{s_1}, S_{s_2}, \cdots, S_{s_k}, \cdots, S_{s_{k+l}} \qquad (k > 0, l \geqq 0)$$

of symbols on the tape, with the kth symbol scanned and the internal configuration q_t, this complete configuration may be described by the four numbers,

$$w = s_{k-1} + Ns_{k-2} + \cdots + N^{k-2}s_1,$$

s_k, t, and

$$v = s_{k+1} + Ns_{k+2} + \cdots + N^{l-1}s_{k+l}$$

or by the single number,

$$u = p(w, s_k, t, v),$$

where

$$p(x_1, x_2, x_3, x_4) = 2^{x_1}3^{x_2}5^{x_3}7^{x_4}.$$

Each complete configuration of the machine is determined by the preceding one. The manner of this determination is given in the description of the machine, which consists of a number of expressions each of one of the forms $q_tS_sS_{s'}Lq_{t'}$ or $q_tS_sS_{s'}Nq_{t'}$ or $q_tS_sS_{s'}Rq_{t'}$. The occurrence of the first of these means that if in any complete configuration the scanned symbol was S_s and the internal configuration q_t, then the machine goes to the next complete configuration by replacing the scanned symbol by $S_{s'}$ and making the new scanned symbol the symbol on the left of it and the new internal configuration $q_{t'}$. In other words if a complete configuration be described by the number,

$$p(s_{k-1} + Ns_{k-2} + \cdots + N^{k-2}s_1, s, t, s_{k+1} + Ns_{k+2} + \cdots + N^{l-1}s_{k+l})$$
$$= p(s_{k-1} + Nf, s, t, s_{k+1} + Ng),$$

[9] Compare the two papers by Kleene already quoted.

and if $q_t S_s S_{s'} L q_{t'}$ occurs in the description of the machine, then the number describing the next complete configuration is

$$p(f,\ s_{k-1},\ t',\ s' + N(s_{k+1} + Ng)).$$

In the case where we have $q_t S_s S_{s'} N q_{t'}$ the next complete configuration will be described by

$$p(s_{k-1} + Nf,\ s',\ t',\ s_{k+1} + Ng),$$

and in the case of $q_t S_s S_{s'} R q_{t'}$ by

$$p(s' + N(s_{k-1} + Nf),\ s_{k+1},\ t',\ g).$$

We may define a primitive recursive function $d_1(s, t)$ (or $d_2(s, t)$ or $d_3(s, t)$) to have the value 1 or 0 according as an expression of the form $q_t S_s S_{s'} L q_{t'}$ (or $q_t S_s S_{s'} N q_{t'}$ or $q_t S_s S_{s'} R q_{t'}$) does or does not occur in the description of the machine. In each of the three cases $z(s, t)$ is to have the value s' and $c(s, t)$ to have the value t'. $q(x)$, $r(x)$ are to be respectively quotient and remainder of x on division by N, and $\varpi_r(x)$ $(r=2, 3, 5, 7)$ is to be the greatest integer k for which r^k divides x. These functions are primitive recursive.

Then if we put

$$\begin{aligned}\theta(x) = {} & d_1(\varpi_3(x), \varpi_5(x))p(q(\varpi_2(x)), r(\varpi_2(x)), c(\varpi_3(x), \varpi_5(x)), z(\varpi_3(x), \varpi_5(x)) + N\varpi_7(x)) \\ & + d_2(\varpi_3(x), \varpi_5(x))p(\varpi_2(x), z(\varpi_3(x), \varpi_5(x)), c(\varpi_3(x), \varpi_5(x)), \varpi_7(x)) \\ & + d_3(\varpi_3(x), \varpi_5(x))p(z(\varpi_3(x), \varpi_5(x)) + N\varpi_2(x), r(\varpi_7(x)), c(\varpi_3(x), \varpi_7(x)), q(\varpi_7(x))),\end{aligned}$$

and

$$\begin{aligned} u(0) &= p(0, 0, 1, 0) = 5, \\ u(S(x)) &= \theta(u(x)),\end{aligned}$$

$u(x)$ will be the number describing the $(x+1)$th complete configuration of the machine.

$g(x, y)$ is to be defined by

$$\begin{aligned} g(S(x), y) &= 2 \quad (\text{all } x, y \geqq 0), & g(0, 1) &= 0, \\ g(0, 0) &= 2, & g(0, 2) &= 1, \\ g(0, x) &= 2 \quad (x \geqq 3), \end{aligned}$$

and $j(x)$ by,

$$j(x) = g(\varpi_3(u(x)), z(\varpi_3(u(x)), \varpi_5(u(x)))).$$

Then $j(x)=0$ means that in going from the $(x+1)$th to the $(x+2)$th complete configuration the machine prints a figure 0: if $j(x)=1$ it prints 1: $j(x)=2$ otherwise. $\xi(x)$ is defined to be the $(x+1)$th natural number y for which $j(y)=0$, and $\phi(x)$ as follows:

$$\begin{aligned} \phi(0) &= 0, \\ \phi(S(x)) &= 0 & \text{if } j(x) &= 0, \\ &= \phi(x) & \text{if } j(x) &= 2, \\ &= S(\phi(x)) & \text{if } j(x) &= 1. \end{aligned}$$

Then $\phi(x)$ is the number of times 1 has been printed since the last 0, reckoned at the $(x+1)$th complete configuration. $\phi(\xi(x))$ is the number of times 1 occurs between the xth and the $(x+1)$th figure 0, its value when $x=0$ being the number of figures 1 which precede all figures 0. But these are the properties which define $f(x)$.

PRINCETON UNIVERSITY