TRANSACTIONS OF THE
AMERICAN MATHEMATICAL SOCIETY
Volume 276, Number 1, March 1983

SYNTAX AND SEMANTICS IN HIGHER-TYPE RECURSION THEORY[1]

BY

DAVID P. KIERSTEAD

ABSTRACT. Recursion in higher types was introduced by S. C. Kleene in 1959. Since that time, it has come to be recognized as a natural and important generalization of ordinary recursion theory. Unfortunately, the theory contains certain apparent anomalies, which stem from the fact that higher type computations deal with the intensions of their arguments, rather than the extensions. This causes the failure of the substitution principle (that if $\varphi(\alpha^{j+1}, \mathfrak{A})$ and $\theta(\beta^j, \mathfrak{A})$ are recursive, then there should be a recursive $\psi(\mathfrak{A})$ such that $\psi(\mathfrak{A}) \simeq \varphi(\lambda\beta^j\theta(\beta^j, \mathfrak{A}), \mathfrak{A})$ at least whenever $\lambda\beta^j\theta(\beta^j, \mathfrak{A})$ is total), and of the first recursion principle (that if $\mathbf{F}(\zeta; \mathfrak{A})$ is a recursive functional, then the minimal solution ζ of the equation $\mathbf{F}(\zeta; \mathfrak{A}) \simeq \zeta(\mathfrak{A})$ should be recursive as well). In an effort to remove—or at least explain—these anomalies, Kleene, in 1978, developed a system for computation in higher types which was based entirely on the syntactic manipulation of formal expressions, called j-expressions. As Kleene pointed out, no adequate semantics for these expressions can be based on the classical (total) type structure Tp over $\mathbf{N}$. In a paper to appear in *The Kleene Symposium* (North-Holland), we showed that an appropriate semantics could be based on the type structure $\hat{Tp}$, which is obtained by adding a new object $\mathfrak{u}$ at level 0 and, at level $(j+1)$, allowing all *monotone*, partial functions from type $\hat{j}$ into $\mathbf{N}$. Over $\hat{Tp}$, both of the principles mentioned above do hold. There is a natural embedding to Tp into $\hat{Tp}$.

In this paper, we complement the syntactic structure with a syntax-free definition of recursion over $\hat{Tp}$, and show that the two notions are equivalent. This system admits an enumeration theorem, in spite of the fact that the presence of partial objects complicates the coding of finite sequences. Indeed, it is not possible to code all finite sequences from type $\hat{j}$ as type-$\hat{j}$ objects. We use the combination of the syntactic and semantic systems to prove that, for any $\varphi\colon Tp^{(\sigma)} \underset{p}{\to} \mathbf{N}$, the following are equivalent:

A. φ is recursive in the sense of Kleene [**1959**],

B. φ is recursive in the sense of Kleene [**1978**], and

C. φ is the pull-back in Tp of some recursive $\psi\colon \hat{Tp}^{(\sigma)} \underset{p}{\to} \mathbf{N}$.

Using these equivalences, we give a necessary and sufficient condition on $\theta\colon Tp^{(\sigma)} \underset{p}{\to} \mathbf{N}$, under which the substitution principle mentioned above will hold for any recursive $\varphi\colon Tp^{(\tau)} \underset{p}{\to} \mathbf{N}$. With one trivial exception, the condition is that if $j \geqslant 1$, then $\mathfrak{A}$ must contain a variable of type greater than j. We feel that this result is particularly natural in the current setting.

0. Introduction. The study of recursion in higher types was introduced by Kleene [**1959**]. In this theory, rather than restricting one's attention to number-theoretic

Received by the editors March 16, 1981.

1980 *Mathematics Subject Classification*. Primary 03D65.

[1] This paper is a revised version of the author's doctoral dissertation at the University of Wisconsin, Madison, 1979.

functions, one considers functions taking their values in **N** (the natural numbers), but whose arguments are taken from the collection Tp of *objects of finite type* over **N**. Tp is defined recursively by

$$\begin{aligned} Tp^{(0)} &= \mathbf{N}, \\ Tp^{(j+1)} &= \mathbf{N}^{Tp^{(j)}} = \text{the set of } \textit{total} \text{ functions from } Tp^{(j)} \text{ into } \mathbf{N}, \\ Tp &= \bigcup_{j<\omega} Tp^{(j)}. \end{aligned}$$

One generally uses variables $\alpha^j, \beta^j, \ldots,$ to range over $Tp^{(j)}$, and $\mathfrak{A}, \mathfrak{B}, \ldots,$ for strings of such variables.

From the outset, this theory was recognized as a natural and important extension of ordinary recursion theory. Unfortunately, two important principles fail in this theory. The first is the substitution principle: If $\varphi(\alpha^n, \mathfrak{A})$ and $\theta(\beta^{n-1}, \mathfrak{A})$ are partial recursive, then certainly $\lambda\mathfrak{A}\varphi(\lambda\beta^{n-1}\theta(\beta^{n-1}, \mathfrak{A}), \mathfrak{A})$ should be partial recursive. Indeed, this principle seems so basic that its failure serves, at best, to dismay the novice, and, at worst, to suggest that there may be something fundamentally amiss with the theory. The second principle is the analogue of the First Recursion Theorem of ordinary recursion theory: If $\mathbf{F}(\zeta; \mathfrak{A})$ is a partial recursive functional, then the minimal solution ζ of the equation $\mathbf{F}(\zeta; \mathfrak{A}) \simeq \zeta(\mathfrak{A})$ should also be partial recursive. This second failure seems even more anomalous when one considers that the analogue of the Second Recursion Theorem does hold.[2]

The problem underlying both of these failures is demonstrated by an example of Kleene [**1963**, p. 110]. Here $T(e, a, x)$ is Kleene's T-predicate (see Kleene [**1952**, p. 281]).

0.1 Example (Kleene). Define

$$\chi(e, x) \simeq \begin{cases} 0 & \text{if } \neg T(e, e, x), \\ \text{undefined} & \text{if } T(e, e, x), \end{cases}$$

$$\begin{aligned} \varphi(\sigma^2, e) &\simeq \sigma^2(\lambda x \chi(e, x)), \\ \theta(\alpha) &\simeq 0, \\ \psi(e) &\simeq \varphi(\lambda\alpha\theta(\alpha), e). \end{aligned}$$

φ is partial recursive and $\varphi(\sigma^2, e)$ is defined if and only if $\forall x \neg T(e, e, x)$ (since $\lambda x \chi(e, x)$ is total, hence a type-1 object, if and only if $\forall x \neg T(e, e, x)$). Were ψ partial recursive, then ψ would be partial recursive in the sense of ordinary recursion theory (Kleene [**1959**, XVII]). This is impossible since $\psi(e)$ is defined if and only if $\forall x \neg T(e, e, x)$, and the latter is not recursively enumerable. □

Perhaps one feels cheated by this example. One could easily argue that $\theta(\alpha)$ is defined without using any values of $\alpha(x)$, and therefore $\varphi(\lambda\alpha\theta(\alpha), e)$ should be

[2] Kleene [**1959, 1963**] did obtain restricted versions of both of these principles, but the basic anomalous nature of their failures remained.

defined for all e. It is easy to construct more complicated θ where it is not so clear for what values of x the corresponding values of $\alpha(x)$ are used by θ in determining $\theta(\alpha)$; but there is still the feeling that if $\lambda x\chi(e, x)$ is (correctly) defined for all of these x then $\varphi(\lambda\alpha\theta(\alpha), e)$ should be defined. Kleene seems to have shared these misgivings and made a suggestion for remedying the situation (Kleene **[1963**, p. 111]). Kleene **[1978]** resurrected this idea. He argued that there was no need to attempt a computation of $\chi(e, x)$ until its value was called for in the course of computing $\theta(\lambda x\chi(e, x))$. Thus we might simply carry $\chi(e, x)$ along as a formal expression until such a value is called for. Of course we might soon find ourselves considering formal expressions within formal expressions *ad nauseam*, and it would be no small task to keep all of the syntax straight. Having taken this approach, Kleene was able to include analogues of the substitution and first recursion principles as primitive operations and produce an emiently workable theory.

Kleene **[1978]** developed a system for computation in higher types which was based entirely on the syntactic manipulation of formal expressions called "j-expressions". Under a given assignment Ω to the free variables of the language, each 0-expression is either undefined or defined with some natural number as its value; each $(j+1)$-expression B naturally represents a partial function $[B]_\Omega$ from $T_p^{(j)}$ into the natural numbers. $[B]_\Omega$ need not be total, and thus need not be a type-$(j+1)$ object. In the course of proving certain theorems in §3 of Kleene **[1978]**, it was necessary to know that one j-expression B could be substituted (freely) for another $\bar{B}$ in a 0-expression E without affecting the value of E under Ω. Kleene **[1978**, p. 208] used an example similar to Example 0.2 below to show that the condition $[B]_\Omega = [\bar{B}]_\Omega$ is insufficient to justify these substitutions—even when $[B]_\Omega$ is total. Kleene postponed the justification of these substitutions pending the development of an appropriate semantics for his system. The following example should be comprehensible, even to the reader unfamiliar with Kleene **[1978]**.

0.2 Example. Let B be the 1-expression $\lambda a \cdot cs(a, 0, 0)$. ($cs(a, b, c) \simeq b$ if $a = 0$, c if $a > 0$.) Let $\bar{B}$ be $\lambda a \cdot 0$. Clearly, for any assignment Ω, $[B]_\Omega = [\bar{B}]_\Omega$. It is easy to construct a 0-expression A which is undefined under any assignment (see §3 of Kleene **[1978]**). Let E be the 0-expression $B(A)$. Then, under any Ω, E is undefined, but $\bar{B}(A) = 0$. Thus $\bar{B}$ may not be substituted for B in E—even though $[B]_\Omega = [\bar{B}]_\Omega$ for any Ω. The problem is that the "obvious" interpretation $[B]_\Omega$ of B does not take into account whether B "uses" its argument or not. Although $[B]_\Omega$ and $[\bar{B}]_\Omega$ are extensionally the same, their intensions are different. □

These two examples indicate that an appropriate semantics for Kleene's system must be based on a type structure somewhat different from Tp. In order to avoid intentional considerations, we must allow a new object at level 0 which will distinguish between the $[B]_\Omega$ and $[\bar{B}]_\Omega$ of Example 0.2. In order to avoid the problem of Example 0.1, we must allow our functions to be defined even when their arguments are incompletely defined. Finally, since we will only be computing values of arguments when it is absolutely necessary, we must insure that the value of $\alpha^{j+2}(\beta^{j+1})$ depends only on those values of $\beta^{j+1}(\gamma^j)$ which are defined. That is, α^{j+2} must be monotone. In Kierstead **[1980]** we considered such an extension of the type structure and showed that an appropriate semantics for Kleene's 1978 system could be based on this type structure in a natural way.[3]

We use the notations $\varphi(\mathfrak{A})\downarrow$ and $\varphi(\mathfrak{A})\uparrow$ for "$\varphi(\mathfrak{A})$ is defined" and "$\varphi(\mathfrak{A})$ is undefined" respectively.

0.3 *The extended types.* We define $\hat{Tp}^{(j)}$ (the set of *extended type-j objects*, henceforth denoted by *type* $\hat{j}$) by recursion on j.

$$\hat{Tp}^{(0)} = \{\mathrm{u}, 0, 1, 2, \dots\} = \mathbf{N} \cup \{\mathrm{u}\}.$$

(u is a new object suggesting "undefined".) Define $\alpha^0 \subset \beta^0 \leftrightarrow [\alpha^0 = \mathrm{u} \text{ or } \alpha^0 = \beta^0]$.

$\hat{Tp}^{(j+1)} =$ the set of all *monotone*, partial functions from $\hat{Tp}^{(j)}$ into $\mathbf{N}$.

$$\hat{Tp} = \bigcup_{j<\omega} \hat{Tp}^{(j)}.$$

(α^{j+1} is *monotone* if, whenever $\alpha^{j+1}(\beta^j)\downarrow$ and $\beta^j \subset \gamma^j$, then $\alpha^{j+1}(\gamma^j)\downarrow$ and is equal to $\alpha^{j+1}(\beta^j)$—where " $\subset$ " is the relation defined above at type $\hat{0}$ and has the usual, set-theoretical meaning at all higher levels.) Let us write u^j for the totally undefined type-$\hat{j}$ object. Monotonicity requires (among other things) that if $\alpha^{j+1}(\mathrm{u}^j) = w$ then $\alpha^{j+1} = \lambda\beta^j \cdot w$.[4] □

The computations of Kleene [**1978**] were based on a list of schemata (S0–S11). These schemata were not to be thought of as actually defining functions, but merely as rules to guide computations. While investigating the relationship between the systems of Kleene [**1978**] and Kleene [**1959**], it became clear that it would be convenient if S0–S11 actually did define functions in a natural way. This is the case if one considers them as defining functions on $\hat{Tp}$, rather than Tp. There is a natural embedding $\#\colon Tp \to \hat{Tp}$ of the classical type structure into the extended type structure. This embedding extends to include partial functions on Tp. We will describe # in §4.1. For the moment, it suffices to know that # exists.

In §1, we present the syntactic system of Kierstead [**1980**], which is obtained by carrying out the computations of Kleene [**1978**] over $\hat{Tp}$. In §2, we give a syntax-free definition of the class of (partial) recursive functionals on $\hat{Tp}$ which directly mirrors the syntactic system of §1. In §3, we compare these systems. In §4 we use the combination of the syntax and the semantics to derive some results about the original system of Kleene [**1959**]. §5 contains some additional properties of recursion over $\hat{Tp}$. §6 contains some general discussion of the relationship between this paper and some other work in higher-type recursion theory.

Along the way, we will show that the following are equivalent for any partial functions φ and $\Theta \equiv (\theta_1, \dots, \theta_1)$ over Tp:

(i) φ is partial recursive in Θ in the sense of Kleene [**1959**];

(ii) φ is partial recursive in Θ in the sense of Kleene [**1978**];

[3] Kleene [**1980**, to appear] has developed a somewhat different semantics for his system.

[4] That some semantics for Kleene's 1978 system was necessary, and that such a semantics would have to be based on an extension of the type structure along the lines described above, were suggested by Kleene [**1978** and private communications]. Kleene [**1980**, to appear] has also pursued this idea and arrived at a semantics significantly different from ours. Our type structure agrees with his at the 0 and 1 levels, but differs from there on.

(iii) there is a ψ (over $\hat{Tp}$), partial recursive in $\#\Theta$ in the sense of this paper, such that for all $(\alpha_1^{j_1},\ldots,\alpha_n^{j_n}) \in Tp^{(j_1)} \times \cdots \times Tp^{(j_n)}$,

$$\psi(\#\alpha_1^{j_1},\ldots,\#\alpha_n^{j_n}) \simeq \varphi(\alpha_1^{j_1},\ldots,\alpha_n^{j_n}).$$

In the preparation of this paper, we were strongly influenced by the ideas of Kechris and Moschovakis [**1977**] and, of course, by Kleene [**1959, 1963, 1978**, and **1980**].

The reader familiar with Platek [**1966**] will notice that our system is quite similar to his—although we arrive at it from a very different direction. We will return to this similarity in §6. For the moment, suffice it to say that we feel our system is conceptually much simpler, and corresponds more closely to the actual process of computation. It seems that the firm connection between the syntax and the semantics of computations is an important aid to understanding a traditionally difficult subject.

0.4 *Notation and terminology*. Our notion is, for the most part, standard in the literature. All unexplained notation and terminology are from Kleene [**1952** and **1959**]. We use variables α^j, β^j, $\gamma^j,\ldots,$ to range over $Tp^{(j)}$ or $\hat{Tp}^{(j)}$, depending on the context. When it becomes necessary to make the distinction explicit we will use $\hat\alpha^j$, $\hat\beta^j$, $\hat\gamma^j,\ldots,$ to range over $\hat{Tp}^{(j)}$. We may omit the superscript occasionally when there is no danger of confusion—e.g., we may write $\alpha^j(\beta)$ or $\alpha(\beta^{j-1})$ for $\alpha^j(\beta^{j-1})$, or $\lambda\alpha^j\theta(\alpha, \mathfrak{A})$ for $\lambda\alpha^j\theta(\alpha^j, \mathfrak{A})$. We use $\mathfrak{A}, \mathfrak{B}, \mathfrak{C},\ldots,$ for finite strings of such variables. If $\mathfrak{A}$ is $(\alpha_1^{j_1},\ldots,\alpha_n^{j_n})$ and $\mathfrak{B}$ is $(\beta_1^{j_1},\ldots,\beta_n^{j_n})$, we write $\mathfrak{A} \subset \mathfrak{B}$ to mean that $\alpha_1^{j_1} \subset \beta_1^{j_1},\ldots,\alpha_n^{j_n} \subset \beta_n^{j_n}$. If $\sigma = (j_1,\ldots,j_n)$ is a finite sequence from $\mathbf{N}$, we write $Tp^{(\sigma)}$ for $Tp^{(j_1)} \times \cdots \times Tp^{(j_n)}$, and similarly for $\hat{Tp}^{(\sigma)}$. We say that the *character* of $\mathfrak{A}$ is σ ($\mathrm{ch}(\mathfrak{A}) = \sigma$) if $\mathfrak{A}$ ranges over $Tp^{(\sigma)}$ or $\hat{Tp}^{(\sigma)}$. By $\max \mathrm{tp}(\mathfrak{A})$, we mean the maximum type of any variable in $\mathfrak{A}$. Similarly, if $\mathrm{ch}(\mathfrak{A}) = \sigma$, then $\max \mathrm{tp}(\sigma) = \max \mathrm{tp}(\mathfrak{A})$. We use $a, b, c,\ldots,$ as variables whose intended range is $\mathbf{N}$, but do not exclude the possibility that $a = \mathrm{u} \in \hat{Tp}^{(0)}$. Θ will always stand for a list $(\theta_1,\ldots,\theta_l)$ of function variables whose arguments are from various (specified) $Tp^{(\sigma)}$ or $\hat{Tp}^{(\sigma)}$. We have the usual meanings for $=$, $\neq$, $\simeq$, $\not\simeq$. In particular, $\varphi(\mathfrak{A}) \neq \psi(\mathfrak{C})$ means that both are defined and with different values, while $\varphi(\mathfrak{A}) \not\simeq \psi(\mathfrak{B})$ means that it is not the case that $\varphi(\mathfrak{A}) \simeq \psi(\mathfrak{B})$.[5]

Function will always mean: monotone, partial function taking values in $\mathbf{N}$. ("Monotone" is superfluous when working over Tp.) We shall frequently abuse notation slightly by writing: $\varphi(\mathfrak{A}) \subset \psi(\mathfrak{B})$ to mean $[\varphi(\mathfrak{A})\uparrow$ or $\varphi(\mathfrak{A}) \simeq \psi(\mathfrak{B})]$, $\varphi(\mathfrak{A}) \simeq \alpha^0$ to mean $[\varphi(\mathfrak{A})\uparrow$ and $\alpha^0 = \mathrm{u}$, or $\varphi(\mathfrak{A})\downarrow$ and $\varphi(\mathfrak{A}) = \alpha^0]$, etc. (This is an abuse of notation since $\varphi(\mathfrak{A})$ has *no* value if $\varphi(\mathfrak{A})\uparrow$.) By $X \underset{p}{\to} Y$ ($X \to Y$) we mean the collection of monotone, partial (total) functions from X into Y. As usual, we write $\varphi : X \underset{p}{\to} Y$ to mean $\varphi \in X \underset{p}{\to} Y$.

We will also be dealing with monotone (partial) functionals (henceforth referred to as *functionals*) $\varphi(\theta_1,\ldots,\theta_l; \mathfrak{A})$, where each θ_i ranges over some $Tp^{(\sigma_i)} \underset{p}{\to} \mathbf{N}$ (or

[5] Thus, $\varphi(\mathfrak{A}) \neq \psi(\mathfrak{B})$ is *not* the same as $\neg(\varphi(\mathfrak{A}) = \psi(\mathfrak{B}))$.

$\hat{T}p^{(\sigma_i)} \xrightarrow[p]{} \mathrm{N}$) and $\mathfrak{A}$ ranges over some $Tp^{(\sigma)}$ (or $\hat{T}p^{(\sigma)}$). By *monotone* we mean that φ is monotone both in its function arguments and in its object arguments. That is to say: If $\theta_1 \subset \bar{\theta}_1, \dots, \theta_l \subset \bar{\theta}_l$ and $\mathfrak{A} \subset \overline{\mathfrak{A}}$ then $\varphi(\Theta; \mathfrak{A}) \subset \varphi(\overline{\Theta}; \overline{\mathfrak{A}})$. We do not exclude the possibility that $l = 0$ (so φ is a function) or that $\mathfrak{A}$ is empty.

We write dom(φ) for the *domain* of φ.

We use arbitrary Greek letters for ordinals. We trust that the reader will forgive us and sympathize with us for having run out of alphabets. In any event, there should be no cause for confusion.

1. The syntactic system. We describe briefly the schemata S0–S11, and the definition of *j-expression*, from Kleene **[1978]**. The reader desiring more details is referred to that paper.

1.1 *The schemata.* For the moment, the following schemata should be viewed as merely formal expressions. In particular, one should not worry about the "meaning" of S4.j or S11. (The numbering comes from Kleene **[1978]** and is designed to correspond, where possible, to that of S0–S9 from Kleene **[1959]**.) In S6.j, $\mathfrak{A}$ is the sequence which results from $\mathfrak{A}_1$ by moving the $(k+1)$st type-j variable to the front of the list (so it should really be called S6.j, k). We will explain the schemata more fully below:

(S0) $\quad \varphi(\Theta; \mathfrak{B}, \mathfrak{C}) \simeq \theta_t(\mathfrak{B}).$

(S1.0) $\quad \varphi(\Theta; a, \mathfrak{B}) \simeq a + 1.$

(S1.1) $\quad \varphi(\Theta; a, \mathfrak{B}) \simeq a \dot{-} 1 \left[\simeq \begin{cases} a - 1 & \text{if } a > 0 \\ 0 & \text{if } a = 0 \end{cases} \right].$

(S2.0) $\quad \varphi(\Theta; \mathfrak{A}) \simeq 0.$

(S3) $\quad \varphi(\Theta; a, \mathfrak{B}) \simeq a.$

(S4.0) $\quad \varphi(\Theta; \mathfrak{A}) \simeq \psi(\Theta; \chi(\Theta; \mathfrak{A}), \mathfrak{A}).$

(S4.j) ($j \geqslant 1$) $\quad \varphi(\Theta; \mathfrak{A}) \simeq \psi(\Theta; \lambda\beta^{j-1}\chi(\Theta; \beta^{j-1}, \mathfrak{A}), \mathfrak{A}).$

(S5.1) $\quad \varphi(\Theta; a, b, c, \mathfrak{B}) \simeq cs(a, b, c) \left[\simeq \begin{cases} b & \text{if } a = 0 \\ c & \text{if } a > 0 \end{cases} \right].$

(S6.j) ($j \geqslant 0$) $\quad \varphi(\Theta; \mathfrak{A}) \simeq \psi(\Theta; \mathfrak{A}_1).$

(S7.j) ($j \geqslant 1$) $\quad \varphi(\Theta; \alpha^j, \alpha^{j-1}, \mathfrak{B}) \simeq \alpha^j(\alpha^{j-1}).$

(S11) $\quad \varphi(\Theta; \mathfrak{A}) \simeq \psi(\lambda\mathfrak{A}'\varphi(\Theta; \mathfrak{A}'), \Theta; \mathfrak{A}).$ [$\simeq \psi(\varphi, \Theta; \mathfrak{A})$ briefly].

The schemata are to be applied under the convention (adopted in Kleene **[1959]**, p. 3]) that only the order of the variables within each type is significant,

While S0–S11 do not actually define functionals by themselves, we will often speak of them as though they did. The ψ and χ on the right sides of some of the schemata are to be symbols for functionals previously generated via S0–S11. In S11, ψ must have one more function argument than does φ, and this argument must range over $\hat{T}p^{(\sigma)} \xrightarrow[p]{} \mathrm{N}$, where ch$(\mathfrak{A}) = \sigma$.

A *derivation of* φ (via S0–S11) is a sequence $\varphi_1, \dots, \varphi_p = \varphi$ in which: (i) each φ_i is derived from zero, one, or two of $\varphi_1 - \varphi_{i-1}$ via one of S0–S11, (ii) each φ_i except φ_p

is used as a ψ or a χ in a later application of one of S0–S11, and (iii) if φ_i follows from ψ (or ψ and χ) via one of S0–S11, then $\varphi_1, \ldots, \varphi_i$ is $\psi_1, \ldots, \psi_r$, φ_i (or $\psi_1, \ldots, \psi_r, \chi_1, \ldots, \chi_q, \varphi_i$), where $\psi_1, \ldots, \psi_r$ and $\chi_1, \ldots, \chi_q$ are derivations of ψ and χ respectively. (Kleene called these "canonical derivations", but we will never consider any other kind, so we drop the adjective.)

It should be noticed that: If $\varphi_1, \ldots, \varphi_p = \varphi$ in a derivation of $\varphi(\Theta; \mathfrak{A})$, then each φ_i is of the form $\varphi_i(\vec{\eta}_i, \Theta; \mathfrak{B})$ for some (possibly empty) list $\vec{\eta}_i$ of function variables. This is because the list of function variables only changes at applications of S11, where the first function variable is phased out. Thus, if φ_i is $\varphi_i(\eta_1, \eta_2, \eta_3, \Theta; \mathfrak{B})$, we have a notion of η_2 having been phased out by identification with $\lambda \mathfrak{C} \varphi_h(\eta_3, \Theta; \mathfrak{C})$ (here $h > i$). Of course, η_3 will also be phased out by some $\lambda \vartheta \varphi_k(\Theta; \vartheta)$ with $k > h$.

Clearly one may devise an indexing scheme to code up every detail of a given derivation (including the number and type of each of the variables of each φ_i). We are trying to avoid such details, so we suppose that this has been done in some reasonable manner.

1.2 *The j-expressions.* We will work with a formal language, the terms of which are called *j-expressions*. We write $A \equiv B$ to mean that A and B are the same formal expression. The language is given relative to a fixed list $\Theta = (\theta_1, \ldots, \theta_l)$ of (typed) "assumed function" symbols.

The language consists of:

(i) the function symbols Θ,

(ii) for each $\sigma \in \omega^{<\omega}$, the function symbols $\varphi_1^{\sigma \to 0}, \varphi_2^{\sigma \to 0}, \ldots,$

(iii) for each $j \geqslant 0$, the object variables $\gamma_0^j, \gamma_1^j, \ldots$ (using α^j, β^j, etc. as "metavariables" to represent arbitrary formal variables),

(iv) the symbols λ, $+1$, $\dot{-}\ 1$, 0, and *cs*,

(v) commas, parentheses, and braces. □

The class of *j-expressions* is defined inductively by the clauses:

E1. A type-j variable γ_i^j is a j-expression.

E2. If $A_1, \ldots, A_n$ are, respectively, j_1-, …, j_n-expressions and $\sigma = (j_1, \ldots, j_n)$, then for each i, $\varphi_i^{\sigma \to 0}(A_1, \ldots, A_n)$ is a 0-expression. (The φ_i's will correspond to the φ_i's of some particular derivation.)

E3. If A is a 0-expression, then $\lambda \alpha^j A$ is a $(j + 1)$-expression.

E4. If A is a $(j + 1)$-expression and B is a j-expression, then $\{A\}\ (B)$ (often abbreviated $A(B)$) is a 0-expression.

E5. 0 is a 0-expression.

E6. If A, B, C are 0-expressions, then so are $(A) + 1$, $(A) \dot{-} 1$ and $cs(A, B, C)$.

E7. If $A_1, \ldots, A_n$ are, respectively, j_1-, …, j_n-expressions and θ_t ranges over

$$\hat{T}p^{(j_1, \ldots, j_n)} \underset{p}{\rightarrow} \mathbf{N},$$

then $\theta_t(A_1, \ldots, A_n)$ is a 0-expression. □

Computations will always be conducted with some fixed derivation $\varphi_1, \ldots, \varphi_p$ in mind, and the derivation will determine completely the σ for the formal symbol $\varphi_i^{\sigma \to 0}$. Therefore we may safely drop the annoying superscript. We have ignored the list of function arguments in the E2 clause because Θ remains fixed throughout, and any additional function arguments $\vec{\eta}$ (as in $\varphi_i(\vec{\eta}, \Theta; \mathfrak{A})$) are associated with the unique function symbols φ_k which phase them out in the given derivation—as will become clear.

1.3 *Computations.* We now take Ω to be an assignment of objects from $\hat{Tp}$ to the object variables of the language, and of monotone, partial functions on these objects to the function symbols Θ. E will always represent a 0-expression. We will only compute values of 0-expressions (although this will yield a notion of "value" $[B]_\Omega$ for a j-expression B). We give an inductive definition of $[E]_\Omega = w$ (E has value $w \in \mathbf{N}$ under Ω). Formally, we define $K = \{\langle E, \Omega, w\rangle \colon [E]_\Omega = w\}$ by stages K^ξ, indexed by ordinals, taking $K = \bigcup_\xi K^\xi$. In §1.4 we will show how this may be viewed as a computation procedure. We will use the following notations:

$\Omega[\gamma^j/\chi^j]$ is the assignment obtained from Ω by assigning χ^j to γ^j,

$$K^{<\xi} = \bigcup_{\delta<\xi} K^\delta,$$

$[\alpha^j]_\Omega =$ the object (in $\hat{Tp}^{(j)}$) assigned to α^j by Ω,

$[\theta_t]_\Omega =$ the function assigned to θ_t by Ω,

$$[E]_\Omega^{<\xi} = \begin{cases} w & \text{if } \langle E, \Omega, w\rangle \in K^{<\xi}, \\ \mathrm{u} & \text{if there is no such } w, \end{cases}$$

and, for any $(j+1)$-expression B,

$$[B]_\Omega^{<\xi} = \lambda\chi^j[B(\gamma^j)]_{\Omega[\gamma^j/\chi^j]}^{<\xi}$$

(so $[B]_\Omega^{<\xi}$ is a partial function from $\hat{Tp}^{(j)}$ into $\mathbf{N}$). ($[B]_\Omega^{<\xi}$ is the part of $[B]_\Omega$ built up before stage ξ.) Note: we really do mean to use "$=$" (not "$\simeq$") in the definition of $[E]_\Omega^{<\xi}$. Thus $[E]_\Omega^{<\xi}$ may be u; but in the definition of $[B]_\Omega^{<\xi}$, we are really defining a function—so its values are restricted to $\mathbf{N}$, with $[B]_\Omega^{<\xi}(\chi^j)$ undefined if $[B(\gamma^j)]_{\Omega[\gamma^j/\chi^j]}^{<\xi} = \mathrm{u}$.

We will need to know that $[B]_\Omega^{<\xi} \in \hat{Tp}^{(j+1)}$, i.e., that it is monotone, so we prove the following lemma simultaneously with the definition of K.

1.3.1. Lemma. *If, for each free variable α^j of E, $[\alpha^j]_\Omega \subset [\alpha^j]_{\Omega^*}$, then $[E]_\Omega^{<\xi} \subset [E]_{\Omega^*}^{<\xi}$. In particular, $[B]_\Omega^{<\xi}$ is monotone.*

The definition of $\langle E, \Omega, w\rangle \in K^\xi$ is by cases, according to the form of E as a 0-expression. The E4 case splits into two subcases according to the form of the "A" in $\{A\}$ (B) as a $(j+1)$-expression. In each case the lemma follows easily from the inductive hypothesis.

1.3.2. Definition. In each of the nine cases, we specify the condition which must be met in order to have $\langle E, \Omega, w\rangle \in K^\xi$.

E1. E is α^0 and $[\alpha^0]_\Omega = w$.

E2. E is $\varphi_i(A_1^{j_1}, \dots, A_n^{j_n})$ and $\langle E^*, \Omega, w\rangle \in K^{<\xi}$, where E^* is the result of making the same substitutions (with proper attention to bound variables) on the right side of the schema introducing φ_i as yields E on the left side. One point deserves some clarification: If φ_i is introduced via S0 with the "θ_t" actually an η to be phased out later via S11, say $\varphi_i(\eta, \Theta; \mathfrak{A}) \simeq \eta(\mathfrak{A})$ and φ_k phases out η by $\varphi_k(\Theta; \mathfrak{A}) \simeq \varphi_n(\varphi_k, \Theta; \mathfrak{A})$, then E^* is to be $\varphi_k(\vec{A})$, not $\eta(\vec{A})$.

E4.λ. E is $\{\lambda\alpha^j A\}$ (B) and $\langle S^{\alpha^j}_B A \mid, \Omega, w\rangle \in K^{<\xi}$. ($S^{\alpha^j}_B A \mid$ is the result of substituting B for each free occurrence of α^j in A, after changing bound variables appropriately.)

E4.j. E is $\alpha^j(B)$ and $\hat{\alpha}^j(\beta^{j-1}) = w$, where $\hat{\alpha}^j = [\alpha^j]_\Omega$ and $\beta^{j-1} = [B]^{<\xi}_\Omega$. (Note that the inductive hypothesis assures us that $\beta^{j-1} \in \widehat{Tp}^{(j-1)}$.)

E5. E is 0 and $w = 0$.

E6 + . E is $A + 1$ and $\langle A, \Omega, w_0\rangle \in K^{<\xi}$, where $w = w_0 + 1$.

E6 $\dot{-}$. E is $A \dot{-} 1$ and $\langle A, \Omega, w_0\rangle \in K^{<\xi}$, where $w = w_0 \dot{-} 1$.

E6cs. E is $cs(A, B, C)$ and either:

(i) $\langle A, \Omega, 0\rangle \in K^{<\xi}$ and $\langle B, \Omega, w\rangle \in K^{<\xi}$, or

(ii) $\langle A, \Omega, n + 1\rangle \in K^{<\xi}$ for some n, and $\langle C, \Omega, w\rangle \in K^{<\xi}$.

E7. E is $\theta_t(—)$ and the condition similar to that of E4.j holds.

This completes the definition of K. □

1.3.3. Definition. Let B be a j-expression and E a 0-expression.

(i) $[B]_\Omega = \bigcup_\xi [B]^{<\xi}_\Omega$. (This agrees with the previous definition for the case $B \equiv \alpha^j$.)

(ii)

$$|E|_\Omega = \begin{cases} \text{the least } \xi \text{ such that } \langle E, \Omega, w\rangle \in K^\xi \text{ for some } w \text{ if such a } \xi \text{ exists,} \\ \infty \quad \text{otherwise.} \quad \square \end{cases}$$

We will generally use induction on $|E|_\Omega$ in situations where Kleene (in Kleene [**1978**]) used "induction on computation trees."

1.4 *Computation trees*. For each E, Ω, we describe the construction of a *computation tree* based on E, Ω. If $[E]_\Omega \downarrow$ the tree will be well founded (in the sense that all paths are finite) and the base node will have rank $|E|_\Omega$. Each completed tree will have a *value* associated with it and this value will be $[E]_\Omega$. These trees are essentially the same as those in §2.4 of Kleene [**1978**], except that we must alter the E1, E4.j and E7 steps (which correspond to the evaluation of a variable) to account for the fact that we are dealing with a different set of objects. We picture our trees as lying on their sides and growing from left to right. The *principal branch* issuing from a vertex is the branch proceeding horizontally to the right. When a tree is completed, we will tag the principal branch with the value of the tree. (In order to make our claim about the rank of the tree hold, we must not consider the tag to be a part of the tree.) Each vertex of the tree is occupied by a 0-expression. With each vertex, there is associated an *assignment in force* at that vertex.

Consider a vertex occupied by E with Ω as the assignment in force. By cases on E, we describe the next level of growth beyond this vertex.

E1. E is α^0. If $[\alpha^0]_\Omega = \text{u}$ there is no growth and the tree cannot be completed. If $[\alpha^0]_\Omega = w \in \mathbf{N}$ then the tree is completed by tagging it with the value w.

E2. E is $\varphi_i(A_1^{j_1}, \ldots, A_n^{j_n})$. Taking E^* as in Definition 1.3.2, we add one more vertex, occupied by E^*, along the principal branch and keep Ω as the assignment in force.

$$E — E^* \cdots .$$

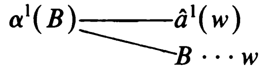
E4.λ. E is $\{\lambda\alpha^j A\}(B)$. We add one more vertex, occupied by $S_B^{\alpha^j}A \mid$, with Ω the assignment in force.

$$\{\lambda\alpha^j A\}(B) - S_B^{\alpha^j}A \mid \cdots .$$

E4.1. E is $\alpha^1(B)$. Take $\hat{\alpha}^1 = [\alpha^1]_\Omega$.

Case 1. $\hat{\alpha}^1(\mathrm{u})\downarrow$. The tree is completed by tagging it with the value $\hat{\alpha}^1(\mathrm{u})$.

Case 2. Otherwise. We place B (under Ω) at the lower, next vertex and begin a computation tree for B. If this tree is completed at some stage with value w and $\hat{\alpha}^1(w)\downarrow$, then the tree for E is completed by tagging it with $\hat{\alpha}^1(w)$.



Otherwise, the tree for E cannot be completed.

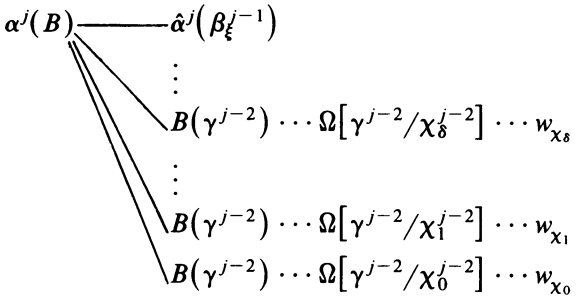
E4.j ($j > 1$). E is $\alpha^j(B)$. Take $\hat{\alpha}^j = [\alpha^j]_\Omega$.

Case 1. $\hat{\alpha}^j(\mathrm{u}^{j-1})\downarrow$. The tree is completed by tagging it with the value $\hat{\alpha}^j(\mathrm{u}^{j-1})$.

Case 2. Otherwise. We begin simultaneous computations of $B(\gamma^{j-2})$ (for some new variable γ^{j-2}) under each assignment $\Omega[\gamma^{j-2}/\chi^{j-2}]$ for $\chi^{j-2} \in \hat{Tp}^{j-2}$. As these computations proceed, one step at a time, some of them may become completed with values $w_{\chi^{j-2}} \in \mathbf{N}$. As computations become completed, we add them, at lower, next vertices, to the computation tree for $\alpha^j(B)$. In this way, we build up in ξ steps, partial functions β_ξ^{j-1}: $\hat{Tp}^{(j-2)} \underset{p}{\rightarrow} \mathbf{N}$ by taking

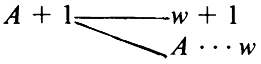
$$\beta_\xi^{j-1}(\chi^{j-2}) \simeq \begin{cases} w_{\chi^{j-2}} & \text{if the computation of} \\ & B(\gamma^{j-2}) \text{ under } \Omega[\gamma^{j-2}/\chi^{j-2}] \\ & \text{was completed in} < \xi \text{ steps,} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Notice that β_ξ^{j-1} is just $[B]_\Omega^{<\xi}$. When and if at some stage ξ, we have $\hat{\alpha}^j(\beta_\xi^{j-1})\downarrow$, the tree for $\alpha^j(B)$ is tagged with this value.

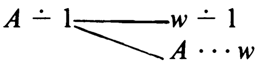

$\alpha^j(B)$
$\hat{\alpha}^j(\beta_\xi^{j-1})$
$B(\gamma^{j-2}) \cdots \Omega[\gamma^{j-2}/\chi_\delta^{j-2}] \cdots w_{\chi_\delta}$
$B(\gamma^{j-2}) \cdots \Omega[\gamma^{j-2}/\chi_1^{j-2}] \cdots w_{\chi_1}$
$B(\gamma^{j-2}) \cdots \Omega[\gamma^{j-2}/\chi_0^{j-2}] \cdots w_{\chi_0}$

E5. E is 0. The tree is tagged with 0.

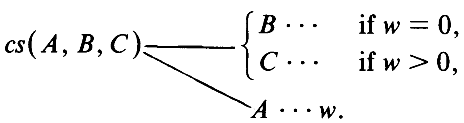
E6 + . E is $A + 1$. We place A at a lower, next vertex and begin computing A under Ω. When and if this computation is completed, say with value w, the tree for $A + 1$ is tagged with $w + 1$.



E6 $\dot{-}$. E is $A \dot{-} 1$. Similarly.



E6 *cs*. E is $cs(A, B, C)$. We compute A (under Ω) at a lower, next vertex. When and if this computation is completed, we place either B or C at the next vertex along the principal branch from E, according as the value for A was 0 or > 0.



E7. E is $\theta_t(A_1^{j_1},\ldots,A_n^{j_n})$. Similarly to E4.$j$, we build up partial functions $\hat{\alpha}_1^{j_1},\ldots,\hat{\alpha}_n^{j_n}$ until we have sufficiently large arguments for $[\theta_t]_\Omega(\hat{\alpha}_1^{j_1},\ldots,\hat{\alpha}_n^{j_n})$ to be defined, at which point we tag the tree for E.

This completes the procedure for constructing computation trees. We shall seldom make actual use of the trees, but they serve well to guide the intuition. If $\langle E, \Omega, w\rangle \in K^\xi$ then the next vertices after E include all those expressions whose values were required in the determination that $\langle E, \Omega, w\rangle \in K^\xi$. The entire tree contains precisely the information that was "used" in determining that $\langle E, \Omega, w\rangle \in K$.

1.5 *Discussion*. In this section, we attempt to justify aspects of our computations which may appear to violate the spirit of computability.

At the E4.j step, we are, in general, obliged to attempt computations of $B(\gamma^{j-2})$ (under assignments $\Omega[\gamma^{j-2}/\chi^{j-2}]$) which cannot be completed. This is a significant deviation from the traditional approach, where a single uncompletable subcomputation will frustrate the entire computation. The next example will show that, under any reasonable semantics for Kleene's formal system, such computations will have to be begun.

If, at step E4.j, we are only allowed to begin computations of $B(\gamma^{j-2})$ under $\Omega^* = \Omega[\gamma^{j-2}/\chi^{j-2}]$ for values of χ^{j-2} such that $[B(\gamma^{j-2})]_{\Omega^*}\downarrow$, and if $[\alpha^j]_\Omega(u^{j-1})\uparrow$, there will have to be a χ^{j-2} such that

$$\forall\beta^{j-1}([\alpha^j]_\Omega(\beta^{j-1})\downarrow \rightarrow \beta^{j-1}(\chi^{j-2})\downarrow).$$

This is because the assignment to γ^{j-2} for the first subcomputation would have to be chosen with no *a priori* knowledge of $[B]_\Omega$, and will have to be in the domain of $[B]_\Omega$.

If every j-expression is to represent an object in $\hat{Tp}^{(j)}$, and if semantically equivalent j-expressions are to yield the same value when substituted (freely) in any 0-expression, then the same will have to hold with any j-expression A in place of α^j. That is, given A, Ω with $[A]_\Omega(u^{j-1})\uparrow$, there must be a χ^{j-2} such that

$$(\forall\beta^{j-1})([A]_\Omega(\beta^{j-1})\downarrow \rightarrow \beta^{j-1}(\chi^{j-2})\downarrow).$$

We now present an example of a 3-expression A for which no such χ may be chosen.[6] There are simpler examples, but they make use of specific derivations

[6] This situation does not occur at types 0, 1, or 2.

(including S11). This example shows that the situation is basic to the definition of a j-expression and independent of the specific schemata allowed.

1.5.1. EXAMPLE. Let A be the 3-expression $\lambda\gamma^2[\gamma^2(\lambda a \cdot \gamma^2(\lambda b \cdot a))]$. For each i, let

$$\beta_i^2(\chi^1) \simeq \begin{cases} n & \text{if } \{\langle i, i+n\rangle, \langle i+1, i+n\rangle, \langle i+2, i+n\rangle, \dots\} \subset \chi_i, \\ i & \text{if } \{\langle i, 0\rangle, \langle i+1, 1\rangle, \langle i+2, 2\rangle, \dots\} \subset \chi_i, \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Then $\beta_i^2(\chi^1)\downarrow$ if and only if χ^1 restricted to $\{m \colon m \geqslant i\}$ is either:

(i) $\lambda m \cdot n$ for some $n \geqslant i$, or

(ii) $\lambda m[m \dot{-} i]$.

Clearly each β_i^2 must be an object in any reasonable semantics for Kleene's j-expressions which includes the classical type-1 objects. Now, under any Ω, $[A]_\Omega(\mathrm{u}^2)\uparrow$ and for all i $[A]_\Omega(\beta_i^2) = i$; but there is no χ^1 such that $\forall_i[\beta_i^2(\chi^1)\downarrow]$. □

Having abandoned the hope of completing all computations which are initiated (under fixed assignments), there are various ways to handle the E4.j step. One method is to begin under some assignment $\Omega^* = \Omega[\gamma^{j-2}/\chi^{j-2}]$ (e.g., with $\chi^{j-2} = \mathrm{u}^{j-2}$), and then allow χ^{j-2} to be extended as certain information about the course of the computation is discovered. This is essentially the approach which Kleene takes in Kleene [**1980**, to appear] and we refer the reader to these papers for further details. Our approach is to attempt computations for all possible χ^{j-2} simultaneously (as a generalization of the dovetailing technique from ordinary recursion theory) until we have built up a partial function β_ξ^{j-1} on which α^j is defined. If one views recursion in higher types as a generalization of recursion relative to a partial function, it is natural to think of α^j as embodied in an oracle which will supply a value $\alpha^j(\beta^{j-1})$ when queried with an argument β^{j-1} in the domain of α^j, and stand mute otherwise. The usual custom is to consider the entire computation as being frustrated if (the oracle for) α^j is ever questioned with a β^{j-1} not in the domain of α^j. Perhaps this viewpoint is too restrictive. One could view our computation procedure as a process by which α^j is first questioned with β_0^{j-1}. If we immediately receive an answer $\alpha^j(\beta_0^{j-1})$, then all is well; but, without worrying about whether we will receive an answer to our first question, we could question α^j with a (possibly transfinite) sequence of arguments $\beta_0^{j-1} \subset \beta_1^{j-1} \subset \cdots \subset \beta_\xi^{j-1} \cdots$ where a value of $\alpha^j(\beta_\xi^{j-1})$ at any one of the arguments would suffice. In this way, we would be able to present α^j with arguments on which it was not defined without violating the principle that computations should be based only on information provided by α^j, rather than on the lack of such information.

1.6 *Adequacy of the semantics.* In Kierstead [**1980**], we showed that the formal system described above does indeed have a natural semantics based on $\hat{Tp}$—that is, a semantics in which every j-expression (under some assignment) is interpreted as an object in $\hat{Tp}^{(j)}$, and in which semantically equivalent j-expressions may be (freely) substituted for each other in a k-expression B, without altering the interpretation of B. This was the content of Theorems 4.1 and 4.2 of Kierstead [**1980**]. Lemma 4.3 was a technical lemma which proves to be useful. We restate these here as 1.6.1, 1.6.2, and 1.6.3 respectively.[7]

1.6.1. Theorem. *Let B be a j-expression. Then $[B]_\Omega \in \hat{Tp}^{(j)}$.* □

1.6.2. Theorem. *Let E be a 0-expression. Let D, $\bar{D}$ be j-expressions not containing the variable β^j. Let $E_D \equiv S^{\beta^j}_D E \mid$, $E_{\bar{D}} \equiv S^{\beta^j}_{\bar{D}} E \mid$. Then, for any Ω*

$$[D]_\Omega\{\subseteqq\}[\bar{D}]_\Omega \to [E_D]_\Omega\{\subseteqq\}[E_{\bar{D}}]_\Omega. \quad \square$$

1.6.3. Lemma. *Let B be a j-expression and A a $(j+1)$-expression, where neither A nor B contains the variable γ^j. Suppose $[A(B)]_\Omega\downarrow$, say $|A(B)|_\Omega = \xi$. Then*

$$[A(B)]_\Omega = \left[A(\gamma^j)\right]_{\Omega[\gamma^j/[B]^{<\xi}_\Omega]}$$

and

$$|A(B)|_\Omega \geqslant |A(\gamma^j)|_{\Omega[\gamma^j/[B]^{<\xi}_\Omega]}. \quad \square$$

Finally, we obtain the notion of a (partial) recursive functional over $\hat{Tp}$ by considering a fixed derivation $\varphi_1, \ldots, \varphi_p = \varphi$, and associating with the 0-expression $\varphi(\mathfrak{A})$ the functional

$$\lambda\vec{\eta}, \mathfrak{B}[\varphi(\mathfrak{A})]_{\Omega[\Theta/\vec{\eta};\mathfrak{A}/\mathfrak{B}]}.$$

In Kierstead [**1980**] we showed that, when this notion is pulled back to Tp (via the embedding #), the theory of Kleene [**1978**] is obtained. We shall return to this in §4.

2. The syntax-free system. In this section, we develop a syntax-free characterization of the class of recursive functionals over $\hat{Tp}$. As in other such systems (e.g., Platek [**1966**], and Kechris and Moschovakis [**1977**]), the basic operation is the process of taking the *least fixed point* of a given functional.

2.1 *Fixed points.* Following Kechris and Moschovakis [**1977**], we call a (monotone, partial) functional $\varphi(\eta, \Theta; \mathfrak{A})$ *operative* if η ranges over $\hat{Tp}^{(\sigma)} \underset{p}{\to} \mathbf{N}$, where $\sigma = \mathrm{ch}(\mathfrak{A})$ (i.e., if $\eta(\mathfrak{A})$ makes sense). If $\varphi(\eta, \Theta; \mathfrak{A})$ is operative, we may define a monotone partial functional $\varphi^\infty(\Theta; \mathfrak{A})$, by recursion on ordinals ξ, via the equations:

$$\varphi^\xi(\Theta; \mathfrak{A}) \simeq \varphi\left(\bigcup_{\delta<\xi} \lambda\mathfrak{A}'\varphi^\delta(\Theta; \mathfrak{A}'), \Theta; \mathfrak{A}\right),$$

$$\varphi^\infty(\Theta; \mathfrak{A}) = w \Leftrightarrow (\exists\xi)\varphi^\xi(\Theta; \mathfrak{A}) = w.$$

For convenience, we write

$$\varphi^{<\xi} = \bigcup_{\delta<\xi} \varphi^\delta.$$

We write

$$|\Theta; \mathfrak{A}|_\varphi = \begin{cases} \text{the least } \xi \text{ such that } \varphi^\xi(\Theta; \mathfrak{A})\downarrow & \text{if such exists,} \\ \infty \quad \text{otherwise.} \end{cases}$$

[7] Actually, 1.6.3 is a special case of Lemma 4.3. Lemma 4.3 was stated in a rather complicated manner in order to provide a sufficiently strong inductive hypothesis.

When there is no danger of confusion, we may write $|\Theta; \mathfrak{A}|_\varphi$ as $|\mathfrak{A}|_\varphi$, or $|\Theta; \mathfrak{A}|$, or even $|\mathfrak{A}|$.

It is trivial to show (by induction on ξ) that these equations do indeed define a monotone partial functional φ^∞. By a simple cardinality argument, there is a ξ such that

$$(\forall\delta \geqslant \xi)\varphi^\delta = \varphi^\xi = \varphi^\infty.$$

We call φ^∞ the *least fixed point* of φ (as is justified by the following lemma).

2.1.1. Lemma. *Let* $\varphi(\eta, \Theta; \mathfrak{A})$ *be operative. Then*:

(i) $(\forall\mathfrak{A}, \Theta)[\varphi(\lambda\mathfrak{A}'\varphi^\infty(\Theta; \mathfrak{A}'), \Theta; \mathfrak{A}) \simeq \varphi^\infty(\Theta; \mathfrak{A})]$.

(ii) $(\forall\Theta)$ *if* $\psi\colon \hat{Tp}^{(\sigma)} \underset{p}{\rightarrow} \mathbf{N}$ *satisfies* $\lambda\mathfrak{A}\varphi(\psi, \Theta; \mathfrak{A}) \subset \psi$, *then* $\lambda\mathfrak{A}\varphi^\infty(\Theta; \mathfrak{A}) \subset \psi$.

Proof. (i) is immediate from the fact that $\varphi^\infty = \varphi^\xi = \varphi^{\xi+1}$ for some ξ. (ii) follows by an easy induction on ξ, using the monotonicity of φ (i.e., one proves that $\forall\xi[\lambda\mathfrak{A}\varphi^\xi(\Theta; \mathfrak{A}) \subset \psi]$). □

2.2 *The recursive functionals.* We use S0–S7 exactly as they are written in §1.1; but as we are now viewing them as actually defining functionals (in §1 and Kleene [**1978**], they were merely rules for computation), we refer to them as S′0–S′7. The only schema which is different is S′11, which we state as:

S′11. $\varphi(\Theta; \mathfrak{A}) \simeq \psi^\infty(\Theta; \mathfrak{A})$.

We will show in §3 that S0–S11 are really the same as S′0–S′11. After that, we will drop the distinction.

Since we are now defining functionals, we must say something about the meanings of the various schemata. In S′1.0, S′1.1, and S′3, φ is to be undefined if $a = \mathrm{u}$. In S′2, φ is defined irrespective of what $\mathfrak{A}$ might be. In S′5, φ is to be undefined if: $a = \mathrm{u}$, or $a = 0$ and $b = \mathrm{u}$, or $a > 0$ and $c = \mathrm{u}$; however, φ will be defined if say $a = 1$, $c = 1$, and $b = \mathrm{u}$. Thus $cs(a, b, c)$ is "strong definition by cases".[8] In order for S′4 and S′11 to make sense, we must known that the χ and ψ on the right sides of these schemata are monotone. It follows by an easy induction that only monotone functionals may be obtained via S′0–S′11. Notice that, since we are working over $\hat{Tp}$, there is no problem if $\lambda\beta^{j-1}\chi(\Theta; \beta^{j-1}, \mathfrak{A})$ in S′4.j is incompletely defined. If, in S′4.0, $\chi(\Theta; \mathfrak{A})\uparrow$, then $\varphi(\Theta; \mathfrak{A})$ is to be $\psi(\Theta; \mathrm{u}, \mathfrak{A})$.

We call $\varphi(\Theta; \mathfrak{A})$ *recursive* if φ is definable via S′0–S′11. $\varphi(\vec{\eta}; \mathfrak{A})$ is *recursive in* Θ if there is a recursive ψ such that $(\forall\vec{\eta}, \mathfrak{A})[\varphi(\vec{\eta}; \mathfrak{A}) \simeq \psi(\vec{\eta}, \Theta; \mathfrak{A})]$. Recursive functionals need not be total.

As in §1.1, we have the notion of a (canonical) derivation of φ.

An *operator* D: $\hat{Tp}^{(\sigma)} \rightarrow \hat{Tp}^{(j+1)}$ is *recursive* (in Θ) if there is a recursive (in Θ) function φ: $\hat{Tp}^{(j)} \times \hat{Tp}^{(\sigma)} \underset{p}{\rightarrow} \mathbf{N}$ such that, for all $\mathfrak{A} \in \hat{Tp}^{(\sigma)}$,

$$D(\mathfrak{A}) = \lambda\beta^j\varphi(\beta^j, \mathfrak{A}).$$

[8] There seems to be some question in the literature as to whether this should be called "strong" or "weak". We are using the word as established by Kleene [**1952**, p. 337].

Notice that a recursive operator is monotone, in the sense that

$$\mathfrak{A} \subset \mathfrak{A}' \Rightarrow D(\mathfrak{A}) \subset D(\mathfrak{A}').$$

The value of a recursive operator at some list of arguments may be substituted for an object variable via S′4. $j + 1$.

Since we allow unlimited use of S′4, it is entirely possible that a given derivation of say $\varphi(\alpha^0, \beta^1, \gamma^2)$ involves functions of say type-4 variables. We will see in §5 that this need not be the case, i.e., if $\max tp(\mathfrak{A}) \leqslant r > 0$ and each of Θ has only arguments of types $\leqslant r - 1$, then $\varphi(\Theta; \mathfrak{A})$ has a derivation in which S4. j is only used with $j \leqslant r - 1$.

The class of recursive functionals has several obvious closure properties which we shall use in the sequel without comment. Among these are: permutation and addition of variables of both sorts, definition by cases, identification of variables (i.e., if $\psi(\alpha^j, \beta^j, \mathfrak{B})$ is recursive, then so is $\varphi(\alpha^j, \mathfrak{B}) \simeq \psi(\alpha^j, \alpha^j, \mathfrak{B})$), and explicit definition using $+1$, -1, *cs* and the λ operator. The proofs of all these are routine. The details may be found in Kleene [**1978**]. (His proofs were carried out in a different system, but they remain formally the same in the present context.) As an example, we prove the functional substitution lemma. It should be noted that this lemma was not so easy in the system of Kechris and Moschovakis [**1977**]. This is because they did not allow iterations of S′11.

2.2.1. Lemma (Functional substitution). *Let*

$$\varphi(\rho_1, \ldots, \rho_n, \Theta; \mathfrak{B}), \quad \chi_1(\Theta; \mathfrak{A}_1, \mathfrak{B}), \ldots, \chi_n(\Theta; \mathfrak{A}_n, \mathfrak{B})$$

be recursive. Then so is ψ *where*

$$\psi(\Theta; \mathfrak{B}) \simeq \varphi(\lambda \mathfrak{A}_1 \cdot \chi_1(\Theta; \mathfrak{A}_1, \mathfrak{B}), \ldots, \lambda \mathfrak{A}_n \cdot \chi_n(\Theta; \mathfrak{A}_n, \mathfrak{B}), \Theta; \mathfrak{B}).$$

Proof. For notational convenience, we write $\lambda \mathfrak{A} \vec{\chi}(\Theta; \mathfrak{A}, \mathfrak{B})$ for

$$\lambda \mathfrak{A}_1 \chi_1(\Theta; \mathfrak{A}_1, \mathfrak{B}), \ldots, \lambda \mathfrak{A}_n \chi_n(\Theta; \mathfrak{A}_n, \mathfrak{B}).$$

We first prove the more general statement, that there is a recursive $\bar{\psi}$ such that

$$\bar{\psi}\ (\Theta; \mathfrak{B}, \mathfrak{B}^*) \simeq \varphi(\lambda \mathfrak{A} \vec{\chi}(\Theta; \mathfrak{A}, \mathfrak{B}^*), \Theta; \mathfrak{B}).$$

The proof is by induction on the length of a given derivation of φ. Throughout the induction, ch($\mathfrak{B}^*$) must, of course, remain fixed, but ch($\mathfrak{B}$) will be determined by the function to which the inductive hypothesis is to be applied. Thus, for example, if φ was introduced by S4.0 as

$$\varphi(\vec{\rho}, \Theta; \mathfrak{B}) \simeq \varphi_0(\vec{\rho}, \Theta; \varphi_1(\vec{\rho}, \Theta; \mathfrak{B}), \mathfrak{B})$$

we would assume the existence of a $\bar{\psi}_0$ such that

$$\bar{\psi}_0(\Theta; a, \mathfrak{B}, \mathfrak{B}^*) \simeq \varphi_0(\lambda \mathfrak{A} \vec{\chi}(\Theta; \mathfrak{A}, \mathfrak{B}^*), \Theta; a, \mathfrak{B}).$$

We take cases according to the schema used to introduce φ. The only nontrivial case is S′11.

Suppose φ was introduced by

$$\varphi(\vec{\rho}, \Theta; \mathfrak{B}) \simeq \varphi_0^{\infty}(\vec{\rho}, \Theta; \mathfrak{B})$$

(so φ_0 is $\varphi_0(\eta, \vec{\rho}, \Theta; \mathfrak{B})$). By the inductive hypothesis (and the simple closure properties mentioned above), we may find a recursive, operative $\bar{\psi}_0$ such that

$$\bar{\psi}_0(\zeta, \Theta; \mathfrak{B}, \mathfrak{B}^*) \simeq \varphi_0(\lambda\mathfrak{B}' \cdot \zeta(\mathfrak{B}', \mathfrak{B}^*), \lambda\mathfrak{A} \cdot \vec{\chi}(\Theta; \mathfrak{A}, \mathfrak{B}^*), \Theta; \mathfrak{B}).$$

We show, by induction on ξ, that (for all Θ, $\mathfrak{B}$, $\mathfrak{B}^*$)

$$\bar{\psi}_0^{\xi}(\Theta; \mathfrak{B}, \mathfrak{B}^*) \simeq \varphi_0^{\xi}(\lambda\mathfrak{A}\vec{\chi}(\Theta; \mathfrak{A}, \mathfrak{B}^*), \Theta; \mathfrak{B}).$$

By simple calculation,

$$\begin{aligned}
\bar{\psi}_0^{\xi}(\Theta; \mathfrak{B}, \mathfrak{B}^*) &\simeq \bar{\psi}_0\big(\lambda\mathfrak{B}'\mathfrak{B}^{*\prime} \cdot \bar{\psi}_0^{<\xi}(\Theta; \mathfrak{B}', \mathfrak{B}^{*\prime}), \Theta; \mathfrak{B}, \mathfrak{P}^*\big)\\
&\simeq \varphi_0\big(\lambda\mathfrak{B}'\bar{\psi}_0^{<\xi}(\Theta; \mathfrak{B}', \mathfrak{B}^*), \lambda\mathfrak{A}\vec{\chi}(\Theta; \mathfrak{A}, \mathfrak{B}^*), \Theta; \mathfrak{B}\big)\\
&\simeq \varphi_0\big(\lambda\mathfrak{B}'\varphi_0^{<\xi}(\lambda\mathfrak{A} \cdot \vec{\chi}(\Theta; \mathfrak{A}, \mathfrak{B}^*), \Theta; \mathfrak{B}'), \lambda\mathfrak{A} \cdot \vec{\chi}(\Theta; \mathfrak{A}, \mathfrak{B}^*), \Theta; \mathfrak{B}\big)\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\text{(by the ind. hyp.)}\\
&\simeq \varphi_0^{\xi}(\lambda\mathfrak{A} \cdot \vec{\chi}(\Theta; \mathfrak{A}, \mathfrak{B}^*), \Theta; \mathfrak{B}),
\end{aligned}$$

as desired. In particular,

$$\bar{\psi}_0^{\infty}(\Theta; \mathfrak{B}, \mathfrak{B}^*) \simeq \varphi_0^{\infty}(\lambda\mathfrak{A} \cdot \vec{\chi}(\Theta; \mathfrak{A}, \mathfrak{B}^*), \Theta; \mathfrak{B}).$$

Thus, we may take $\bar{\psi} = \bar{\psi}_0^{\infty}$.

Finally, we use identification of variables to obtain

$$\psi(\Theta; \mathfrak{B}) \simeq \bar{\psi}\ (\Theta; \mathfrak{B}, \mathfrak{B}) \simeq \varphi(\lambda\mathfrak{A} \cdot \vec{\chi}(\mathfrak{A}, \mathfrak{B}), \Theta; \mathfrak{B}). \quad \Box$$

2.2.2. Lemma (Primitive recursion). *If ψ and χ are recursive, then so is the functional φ defined by the equations*:

$$\begin{aligned}
&\varphi(\Theta; 0, \mathfrak{B}) \simeq \psi(\Theta; \mathfrak{B}),\\
&\varphi(\Theta; a+1, \mathfrak{B}) \simeq \chi(\Theta; a, \varphi(\Theta; a, \mathfrak{B}), \mathfrak{B}),\\
&\varphi(\Theta; \mathrm{u}, \mathfrak{B}) \uparrow .
\end{aligned}$$

(*Note. We are working over $\hat{T}p$, so it is entirely possible that $\varphi(\Theta; a, \mathfrak{B})\uparrow$ and $\varphi(\Theta; a+1, \mathfrak{B})\downarrow$—as is the case if $\chi(\Theta; a, \mathrm{u}, \mathfrak{B})\downarrow$ and $\psi(\Theta; \mathfrak{B})\uparrow$.*)

Proof. Define (using $cs(-,-,-)$)

$$\varphi_0(\eta, \Theta; a, \mathfrak{B}) \simeq \begin{cases} \psi(\Theta; \mathfrak{B}) & \text{if } a = 0,\\ \chi(\Theta; a \dot{-} 1, \eta(a \dot{-} 1, \mathfrak{B}), \mathfrak{B}) & \text{if } a > 0. \end{cases}$$

It is easy to check, by induction on a, that $\varphi_0(\lambda a'\mathfrak{B}' \cdot \varphi_0^{\infty}(\Theta; a', \mathfrak{B}'), \Theta; a, \mathfrak{B}) \simeq \varphi(\Theta; a, \mathfrak{B})$. Thus, by Lemma 2.1.1, $\varphi_0^{\infty}(\Theta; a, \mathfrak{B}) \simeq \varphi(\Theta; a, \mathfrak{B})$. $\Box$

By

$$\mu x[\psi(\Theta; x, \mathfrak{A}) \simeq 0],$$

we mean: the least $x \in \mathbf{N}$ such that $\psi(\Theta; 0, \mathfrak{A}), \ldots, \psi(\Theta; x, \mathfrak{A})$ are all defined and $\psi(\Theta; x, \mathfrak{A}) = 0$, if such an x exists—otherwise $\mu x[\psi(\Theta; x, \mathfrak{A}) \simeq 0]$ is undefined.

2.2.3. Lemma (Least-number operator). *If ψ is recursive, then so is*

$$\varphi(\Theta; \mathfrak{A}) \simeq \mu x[\psi(\Theta; x, \mathfrak{A}) \simeq 0].$$

PROOF. Define

$$\chi(\eta, \Theta; x, \mathfrak{A}) \simeq \begin{cases} x & \text{if } \psi(\Theta; x, \mathfrak{A}) \simeq 0, \\ \eta(x+1, \mathfrak{A}) & \text{if } \psi(\Theta; x, \mathfrak{A}) > 0. \end{cases}$$

It is easy to check that

$$\chi^{\infty}(\Theta; x, \mathfrak{A}) \simeq \mu y[y \geqslant x \text{ and } \psi(\Theta; y, \mathfrak{A}) \simeq 0].$$

Thus

$$\chi^{\infty}(\Theta; 0, \mathfrak{A}) \simeq \mu x[\psi(\Theta; x, \mathfrak{A}) \simeq 0]. \quad \square$$

2.2.4. LEMMA. *If* φ: $Tp^{(0,0,\ldots,0)} \underset{p}{\rightarrow} \mathbf{N}$ *is partial recursive in the sense of ordinary recursion theory, then* φ *is recursive in the present sense* (*when* φ *is viewed as a function on* $\hat{T}p^{(0,\ldots,0)}$ *whose domain happens to be a subset of* $Tp^{(0,\ldots,0)}$).

PROOF. In view of the preceding lemmas, the proof is a trivial induction on the definition of φ. The only problem is that we must make sure that, if any of $\mathfrak{B}$ is u, then $\varphi(\mathfrak{B})\uparrow$. This is easily handled using *cs*$(-,-,-)$. For instance, if we set

$$\psi_0(x, y) \simeq \text{cs}(x, \psi(x, y), \psi(x, y)),$$
$$\psi_1(x, y) \simeq \text{cs}(y, \psi_0(x, y), \psi_0(x, y)),$$

then

$$\psi_1(x, y) \simeq \begin{cases} \psi(x, y) & \text{if } x, y \in Tp^{(0)}, \\ \text{undefined} & \text{otherwise.} \quad \square \end{cases}$$

3. The connection between syntax and semantics. In this section, we show that the recursive functionals on $\hat{T}p$ are precisely those obtained via the computations of §1, and that the same functional is produced by a given derivation—whether via S0–S11 or S′0–S′11.

Consider a derivation $\varphi_1, \ldots, \varphi_p$ via S0–S11. By the *corresponding derivation* via S′0–S′11, we mean the derivation $\psi_1, \ldots, \psi_p$ where ψ_i is introduced via S′0–S′11 in exactly the same way that φ_i was introduced via S0–S11. For example, if

(S11) $$\varphi_i(\Theta; \mathfrak{A}) \simeq \varphi_{i-1}(\varphi_i, \Theta; \mathfrak{A})$$

then

(S′11) $$\psi_i(\Theta; \mathfrak{A}) \simeq \psi_{i-1}^{\infty}(\Theta; \mathfrak{A}).$$

Of course, we require that the variables of ψ_i be of the same number and types as those of φ_i.

3.1. THEOREM. *Let* $\varphi_1, \ldots, \varphi_p$ *be a derivation of* $\varphi = \varphi_p$ *via* S0–S11. *Let* $\psi_1, \ldots, \psi_p = \psi$ *be the corresponding derivation via* S′0–S′11. *Then, for all* Θ *and* $\mathfrak{A}$,

$$\psi(\Theta; \mathfrak{A}) \simeq \varphi(\Theta; \mathfrak{A}),$$

in the sense that for all assignments Ω

$$\psi([\Theta]_{\Omega}; [\mathfrak{A}]_{\Omega}) \simeq [\varphi_p(\mathfrak{A})]_{\Omega}.$$

(*Recall that* 0*-expressions never show* Θ *explicitly*.)

Proof. The lists of function arguments of the various φ_t (and ψ_t) will vary with t, but will always be of the form $(\eta_{i+1},\dots,\eta_n,\Theta)$, where n is the number of applications of S11 (see §1.1). For notational convenience, we will write $(\vec{\eta}_t,\Theta)$ to mean $(\eta_{i+1},\dots,\eta_n,\Theta)$ where φ_t has $n-i$ function arguments in addition to Θ. In particular, $\vec{\eta}_p$ is an empty list. We should really write $\mathfrak{A}_t$ as well, since the list of object variables changes also; but it is not quite as important to keep track of these, so we avoid the additional subscript.

Let Ω be an assignment. We will generally use the same symbols, Θ and $\mathfrak{A}$, for the formal variables Θ and $\mathfrak{A}$, and for the functions $[\Theta]_\Omega$ and objects $[\mathfrak{A}]_\Omega$ assigned to them by Ω. However, we must be careful to distinguish between the 0-expression $\varphi_i(\mathfrak{A})$ and the (mythical) function φ_i. As yet, S0–S11 still do not define functions.

Suppose that the n applications of S11 introduce $\varphi_{q_1},\dots,\varphi_{q_n}$. So φ_{q_i} is introduced by

$$\varphi_{q_i}(\eta_{i+1},\dots,\eta_n,\Theta;\mathfrak{A})\simeq\varphi_{q_i-1}(\varphi_{q_i},\eta_{i+1},\dots,\eta_n,\Theta;\mathfrak{A}),$$
$$\left[\varphi_{q_i}(\vec{\eta}_{q_i},\Theta;\mathfrak{A})\simeq\varphi_{q_i-1}(\varphi_{q_i},\vec{\eta}_{q_i},\Theta;\mathfrak{A})\right].$$

We are now ready to give the proof, which we split into two parts ($\subset$ and $\supset$).

($\subset$) Let $\chi_i=\lambda\hat{\mathfrak{A}}[\varphi_{q_i}(\mathfrak{A})]_{\Omega[\mathfrak{A}/\hat{\mathfrak{A}}]}$ (for $1\leqslant i\leqslant n$). We use the notation $\vec{\chi}_t$ in the same way as $\vec{\eta}_t$. That is, $\vec{\chi}_t\equiv(\chi_{i+1},\dots,\chi_n)$. We show, by induction on k, that

$$\psi_k(\vec{\chi}_k,\Theta;\mathfrak{A})\subset[\varphi_k(\mathfrak{A})]_\Omega.$$

In particular, $\psi_p(\Theta;\mathfrak{A})\subset[\varphi_p(\mathfrak{A})]_\Omega$.

The inductive hypothesis applies to all i, Ω^* where $i<k$ and Ω^* agrees with Ω and Θ. We take cases according to the schema used to introduce φ_k.

S0. *Case* 1. $\varphi_k(\eta_k,\Theta;\mathfrak{B},\mathfrak{C})\simeq\theta_t(\mathfrak{B})$. Then

$$\psi_k(\vec{\chi}_k,\Theta;\mathfrak{B},\mathfrak{C})\simeq\theta_t(\mathfrak{B})\simeq[\theta_t(\mathfrak{B})]_\Omega\simeq[\varphi_k(\mathfrak{B},\mathfrak{C})]_\Omega.$$

Case 2. $\varphi_k(\vec{\eta}_k,\Theta;\mathfrak{B},\mathfrak{C})\simeq\eta_i(\mathfrak{B})$. ($\eta_i$ is later phased out by φ_{q_i} via S11.) Then

$$\psi_k(\vec{\chi}_k,\Theta;\mathfrak{B},\mathfrak{C})\simeq\chi_i(\mathfrak{B})\simeq[\varphi_{q_i}(\mathfrak{B})]_\Omega\simeq[\varphi_k(\mathfrak{B},\mathfrak{C})]_\Omega.$$

S1–S3. Trivial.

S4.0. $\varphi_k(\vec{\eta}_k,\Theta;\mathfrak{A})\simeq\varphi_h(\vec{\eta}_k,\Theta;\varphi_i(\vec{\eta}_k,\Theta;\mathfrak{A}),\mathfrak{A})$. (Here $i,h<k$ and $\vec{\eta}_i\equiv\vec{\eta}_h\equiv\vec{\eta}_k$.) Let $\beta^0=[\varphi_i(\mathfrak{A})]_\Omega$ (either u or an element of $\mathbf{N}$). By the inductive hypothesis, $\psi_i(\vec{\chi}_k,\Theta;\mathfrak{A})\subset\beta^0$; so, again by the inductive hypothesis,

$$\begin{aligned}\psi_k(\vec{\chi}_k,\Theta;\mathfrak{A})&\simeq\psi_h(\vec{\chi}_k,\Theta;\psi_i(\vec{\chi}_k,\Theta;\mathfrak{A}),\mathfrak{A})\\&\subset\psi_h(\vec{\chi}_k,\Theta;\beta^0,\mathfrak{A})\qquad(\psi_h\text{ being monotone})\\&\subset[\varphi_h(\gamma^0,\mathfrak{A})]_{\Omega[\gamma^0/\beta^0]}\\&\simeq[\varphi_h(\varphi_i(\mathfrak{A}),\mathfrak{A})]_\Omega\quad(\text{by }1.6.2)\\&\simeq[\varphi_k(\mathfrak{A})]_\Omega.\end{aligned}$$

S4.j ($j\geqslant1$). $\varphi_k(\vec{\eta}_k,\Theta;\mathfrak{A})\simeq\varphi_h(\vec{\eta}_k,\Theta;\lambda\beta^{j-1}\varphi_i(\vec{\eta}_k,\Theta;\beta^{j-1},\mathfrak{A}),\mathfrak{A})$. (Here $i,h<k$ and $\vec{\eta}_i\equiv\vec{\eta}_h\equiv\vec{\eta}_k$.) Let $\beta^j=[\lambda\beta^{j-1}\varphi_i(\beta^{j-1},\mathfrak{A})]_\Omega$. By the inductive hypothesis,

$$\lambda\beta^{j-1}\psi_i(\vec{\chi}_k,\Theta;\beta^{j-1},\mathfrak{A})\subset\beta^j;$$

so, again by the inductive hypothesis,

$$\begin{aligned}\psi_k(\vec{\chi}_k, \Theta; \mathfrak{A}) &\simeq \psi_h\big(\vec{\chi}_k, \Theta; \lambda\beta^{j-1}\psi_i(\vec{\chi}_k, \Theta; \beta^{j-1}, \mathfrak{A}), \mathfrak{A}\big)\\ &\subset \psi_h(\vec{\chi}_k, \Theta; \beta^j, \mathfrak{A}) \qquad (\psi_h \text{ being monotone})\\ &\subset \big[\varphi_h(\gamma^j, \mathfrak{A})\big]_{\Omega[\gamma^j/\beta^j]}\\ &\simeq \big[\varphi_h\big(\lambda\beta^{j-1}\varphi_i(\beta^{j-1}, \mathfrak{A}), \mathfrak{A}\big)\big]_\Omega \quad (\text{by } 1.6.2)\\ &\simeq \big[\varphi_k(\mathfrak{A})\big]_\Omega.\end{aligned}$$

S5–S7. Trivial.

S11. $\varphi_k(\vec{\eta}_k, \Theta; \mathfrak{A}) \simeq \varphi_{k-1}(\varphi_k, \vec{\eta}_k, \Theta; \mathfrak{A})$. Here $k = q_i$ for some $i \leqslant n$, $\vec{\eta}_k \equiv (\eta_{i+1}, \ldots, \eta_n)$, and $\vec{\eta}_{k-1} \equiv (\eta_i, \ldots, \eta_n)$. Now $\psi_k(\vec{\chi}_k, \Theta; \mathfrak{A}) \simeq \psi^\infty_{k-1}(\vec{\chi}_k, \Theta; \mathfrak{A})$; so, by Lemma 2.1.1, it will suffice to show that

$$\lambda\hat{\mathfrak{A}}\psi_{k-1}\big(\lambda\mathfrak{A}'[\varphi_k(\mathfrak{A})]_{\Omega[\mathfrak{A}/\mathfrak{A}']}, \vec{\chi}_k, \Theta; \hat{\mathfrak{A}}\big) \subset \lambda\hat{\mathfrak{A}}[\varphi_k(\mathfrak{A})]_{\Omega[\mathfrak{A}/\hat{\mathfrak{A}}]},$$

i.e., (since $[\varphi_k(\mathfrak{A})]_{\Omega[\mathfrak{A}/\hat{\mathfrak{A}}]} = [\varphi_{k-1}(\mathfrak{A})]_{\Omega[\mathfrak{A}/\hat{\mathfrak{A}}]}$) that for all $\hat{\mathfrak{A}}$

$$\psi_{k-1}(\chi_i, \vec{\chi}_k, \Theta; \hat{\mathfrak{A}}) \subset \big[\varphi_{k-1}(\mathfrak{A})\big]_{\Omega[\mathfrak{A}/\hat{\mathfrak{A}}]}.$$

But this is exactly what the inductive hypothesis says.

($\supset$) Let $\chi_n = \lambda\mathfrak{A}\psi_{q_n}(\Theta; \mathfrak{A})$ $(= \lambda\mathfrak{A}\psi^\infty_{q_n-1}(\Theta; \mathfrak{A}))$, and, for $1 \leqslant i < n$, let

$$\begin{aligned}\chi_i &= \lambda\mathfrak{A}\psi_{q_i}(\chi_{i+1}, \ldots, \chi_n, \Theta; \mathfrak{A})\\ &\big(= \lambda\mathfrak{A}\psi^\infty_{q_{i-1}}(\chi_{i+1}, \ldots, \chi_n, \Theta; \mathfrak{A})\big).\end{aligned}$$

We retain the notation $\vec{\chi}_k$ for $(\chi_{i+1}, \ldots, \chi_n)$, where ψ_k is $\psi_k(\eta_{i+1}, \ldots, \eta_n, \Theta; \mathfrak{A})$.

We prove, by induction on $\xi = |\varphi_k(\mathfrak{A})|_\Omega$ that, for all $k \leqslant p$,

$$\big[\varphi_k(\mathfrak{A})\big]_\Omega \subset \psi_k(\vec{\chi}_k, \Theta; \mathfrak{A}).$$

The desired result is the case where $k = p$. We take cases according to the schema introducing φ_k.

S0. *Case* 1. $\varphi_k(\vec{\eta}_k, \Theta; \mathfrak{B}, \mathfrak{C}) \simeq \theta_t(\mathfrak{B})$. As before.

Case 2. $\varphi_k(\vec{\eta}_k, \Theta; \mathfrak{B}, \mathfrak{C}) \simeq \eta_i(\mathfrak{B})$. ($\eta_i$ is later identified with φ_{q_i} via S11.) Now $[\varphi_k(\mathfrak{B}, \mathfrak{C})]_\Omega \simeq [\varphi_{q_i}(\mathfrak{B})]_\Omega$ and $|\varphi_{q_i}(\mathfrak{B})|_\Omega < |\varphi_k(\mathfrak{B}, \mathfrak{C})|_\Omega$. Thus the inductive hypothesis yields that

$$\big[\varphi_{q_i}(\mathfrak{B})\big]_\Omega \subset \psi_{q_i}(\vec{\chi}_{q_i}, \Theta; \mathfrak{B}) \simeq \chi_i(\mathfrak{B}) \simeq \psi_k(\vec{\chi}_k, \Theta; \mathfrak{A}).$$

S1–S3. Trivial.

S4.0. $\varphi_k(\vec{\eta}_k, \Theta; \mathfrak{A}) \simeq \varphi_h(\vec{\eta}_k, \Theta; \varphi_i(\vec{\eta}_k, \Theta; \mathfrak{A}), \mathfrak{A})$. Here $\vec{\eta}_k \equiv \vec{\eta}_h \equiv \vec{\eta}_i$. Let $\beta^0 = [\varphi_i(\mathfrak{A})]^{<\xi}_\Omega$. By the inductive hypothesis, $\beta^0 \subset \psi_i(\vec{\chi}_k, \Theta; \mathfrak{A})$. By Lemma 1.6.3,

$$\big[\varphi_h(\varphi_i(\mathfrak{A}), \mathfrak{A})\big]_\Omega \simeq \big[\varphi_h(\gamma^0, \mathfrak{A})\big]_{\Omega[\gamma^0/\beta^0]}$$

and

$$|\varphi_h(\gamma^0, \mathfrak{A})|_{\Omega[\gamma^0/\beta^0]} \leqslant |\varphi_h(\varphi_i(\mathfrak{A}), \mathfrak{A})|_\Omega < |\varphi_k(\mathfrak{A})|_\Omega.$$

Thus the inductive hypothesis applies to yield (using the monotonicity of ψ_h)

$$\begin{aligned}\big[\varphi_h(\gamma^0, \mathfrak{A})\big]_{\Omega[\gamma^0/\beta^0]} &\subset \psi_h(\vec{\chi}_k, \Theta; \beta^0, \mathfrak{A})\\ &\subset \psi_h(\vec{\chi}_k, \Theta; \psi_i(\vec{\chi}_k, \Theta; \mathfrak{A}), \mathfrak{A}) \simeq \psi_k(\vec{\chi}_k, \Theta; \mathfrak{A}).\end{aligned}$$

So we have

$$[\varphi_k(\mathfrak{A})]_\Omega \simeq [\varphi_h(\varphi_i(\mathfrak{A}), \mathfrak{A})]_\Omega \subset \psi_k(\vec{\chi}_k, \Theta; \mathfrak{A}).$$

S4.j ($j \geqslant 1$). $\varphi_k(\vec{\eta}_k, \Theta; \mathfrak{A}) \simeq \varphi_h(\vec{\eta}_k, \Theta; \lambda\beta^{j-1}\varphi_i(\vec{\eta}_k, \Theta; \beta^{j-1}, \mathfrak{A}), \mathfrak{A})$. This case is handled similarly to S4.0, by using Lemma 1.6.3 again and the fact that the inductive hypothesis yields

$$[\lambda\beta^{j-1}\varphi_i(\beta^{j-1}, \mathfrak{A})]_\Omega^{<\xi} \subset \lambda\beta^{j-1}\psi_i(\vec{\chi}_k, \Theta; \beta^{j-1}, \mathfrak{A}).$$

S5–S7. Trivial.

S11. $\varphi_k(\vec{\eta}_k, \Theta; \mathfrak{A}) \simeq \varphi_{k-1}(\varphi_k, \vec{\eta}_k, \Theta; \mathfrak{A})$. Here $k = q_i$ for some i, $\vec{\eta}_k \equiv (\eta_{i+1}, \ldots, \eta_n)$, and $\vec{\eta}_{k-1} \equiv (\eta_i, \ldots, \eta_n) \equiv (\eta_i, \vec{\eta}_k)$.

$|\varphi_{k-1}(\mathfrak{A})|_\Omega < |\varphi_k(\mathfrak{A})|_\Omega$, so the inductive hypothesis yields

$$\begin{aligned}
[\varphi_k(\mathfrak{A})]_\Omega \simeq [\varphi_{k-1}(\mathfrak{A})]_\Omega \subset \psi_{k-1}(\vec{\chi}_{k-1}, \Theta; \mathfrak{A}) &\simeq \psi_{k-1}(\chi_i, \vec{\chi}_k, \Theta; \mathfrak{A}) \\
&\simeq \psi_{k-1}(\lambda\mathfrak{A}'\psi_{k-1}^{\infty}(\vec{\chi}_k, \Theta; \mathfrak{A}'), \vec{\chi}_k, \Theta; \mathfrak{A}) \\
&\simeq \psi_{k-1}^{\infty}(\vec{\chi}_k, \Theta; \mathfrak{A}) \quad \text{(by 2.1.1)} \\
&\simeq \psi_k(\vec{\chi}_k, \Theta; \mathfrak{A}).
\end{aligned}$$

Putting the two parts ($\subset$ and $\supset$) together, we have proven Theorem 3.1. □

In view of Theorem 3.1, we need no longer maintain the distinction between S0–S11 and S′0–S′11. They both define the same functionals.

Let z be the index of a derivation $\varphi_1, \ldots, \varphi_p$ from Θ. Then we take z as an index of the functional φ_p as well, and write

$$\{z\}(\Theta; \mathfrak{A}) \simeq \varphi_p(\Theta; \mathfrak{A})$$

or, alternatively, $\{z\}^\Theta(\mathfrak{A})$.

4. Connections with the classical theory. In this chapter, we will prove the equivalence of the following:

(A) The theory of Kleene [**1959**].

(B) The theory of Kleene [**1978**].

(C) The pull-back in Tp of the theory of §§1 and 2.

We will also derive some results about the theory of Kleene [**1959**].

4.1 *Embedding Tp in $\hat{Tp}$.* In transferring some of our results about recursion on $\hat{Tp}$ to recursion on Tp, we make use of an embedding of Tp into $\hat{Tp}$. To avoid confusion, we will (usually) use variables like $\hat{\alpha}^j$ for elements of $\hat{Tp}^{(j)}$ and α^j for elements of $Tp^{(j)}$. Notice that $Tp^{(0)} \subset \hat{Tp}^{(0)}$ and $Tp^{(1)} \subset \hat{Tp}^{(1)}$, but $Tp^{(j+2)} \not\subset \hat{Tp}^{(j+2)}$. The reason for that is that, e.g., $\alpha^2 \in Tp^{(2)}$ is not monotone when considered as a partial function on $\hat{Tp}^{(1)}$. For instance, if $\beta^1 \in Tp^{(1)}$ is defined by $\beta^1(c) = 0$ for all $c \in Tp^{(0)}$, then $\beta^1 \in \hat{Tp}^{(1)}$ and $\beta^1 \underset{\neq}{\subset} \lambda\hat{c} \cdot 0$. Now $\alpha^2(\beta^1)\downarrow$, but $\mathrm{dom}(\alpha^2) = Tp^{(1)}$ and $\lambda\hat{c} \cdot 0 \in \hat{Tp}^{(1)} - Tp^{(1)}$, so $\alpha^2(\lambda\hat{c} \cdot 0)\uparrow$. Of course, α^2 has a minimum monotone extension. This will provide our embedding.

We define $\# : Tp^{(j)} \to \hat{Tp}^{(j)}$ by recursion on j. We will need to prove the following lemma simultaneously with the definition.

4.1.1. Lemma. (i) $\#$ *is one-to-one.*

(ii) *If* $j \geqslant 1$, *then every* $\#\alpha^j$ *has the same domain, namely*

$$\{\hat{\beta}^{j-1} \in \hat{Tp}^{(j-1)}\colon (\exists \beta^{j-1} \in Tp^{(j-1)})(\#\beta^{j-1} \subset \hat{\beta}^{j-1})\}.$$

(iii) *If* $\hat{\alpha}^j \in \hat{Tp}^{(j)}$ *then there is at most one* $\alpha^j \in Tp^{(j)}$ *such that* $\#\alpha^j \subset \hat{\alpha}^j$. □

4.1.2. Definition.

$$\#\alpha^0 = \alpha^0,$$

$$(\#\alpha^{j+1})(\hat{\beta}^j) \simeq \begin{cases} \alpha^{j+1}(\beta^j) & \text{if } \#\beta^j \subset \hat{\beta}^j \text{ for some } \beta^j \in Tp^{(j)}, \\ \text{undefined} & \text{if there is no such } \beta^j. \end{cases}$$

Notice that part (iii) of the lemma (which holds for j, by the inductive hypothesis) assures us that $\#\alpha^{j+1}$ is well defined. It is easy to verify the three parts of the lemma ((iii) follows from (i) and (ii)). Also notice that $\#$ is the identity on $Tp^{(0)}$ and $Tp^{(1)}$.

If $\mathfrak{A} \equiv (\alpha_0^{j_0}, \ldots, \alpha_n^{j_n})$ then we write $\#\mathfrak{A}$ for $(\#\alpha_0^{j_0}, \ldots, \#\alpha_n^{j_n})$.

For any partial function θ on $Tp^{(\sigma)}$, we may define a monotone partial function $\#\theta$ on $\hat{Tp}^{(\sigma)}$ by

$$(\#\theta)(\hat{\mathfrak{A}}) \simeq \begin{cases} \theta(\mathfrak{A}) & \text{if } \#\mathfrak{A} \subset \hat{\mathfrak{A}} \text{ for some } \mathfrak{A} \in Tp^{(\sigma)}, \\ \text{undefined} & \text{if there is no such } \mathfrak{A}. \end{cases}$$

Given any assignment Ω from Tp, we may define an assignment $\#\Omega$ from $\hat{Tp}$ by

$$[\alpha^j]_{\#\Omega} = \#([\alpha^j]_\Omega),$$

$$[\theta]_{\#\Omega} = \#([\theta]_\Omega). \quad \square$$

We will use the following notations:

$$U^{(j)} = \{\#\beta^j \in \hat{Tp}^{(j)}\colon \beta^j \in Tp^{(j)}\},$$

$$V^{(j)} = \{\hat{\beta}^j \in \hat{Tp}^{(j)}\colon (\exists \beta^j \in Tp^{(j)})(\#\beta^j \subset \hat{\beta}^j)\}.$$

The next lemma is trivial, but useful.

4.1.3. Lemma. (i) $U^{(j+1)} = \{\hat{\beta}^{j+1}\colon V^{(j)} = \operatorname{dom}(\hat{\beta}^{j+1})\}$.

(ii) $V^{(j+1)} = \{\hat{\beta}^{j+1}\colon U^{(j)} \subset \operatorname{dom}(\hat{\beta}^{j+1})\} = \{\hat{\beta}^{j+1}\colon V^{(j)} \subset \operatorname{dom}(\hat{\beta}^{j+1})\}$.

Proof. The proof is an easy induction on j. □

The next theorem is Theorem 5.3 of Kierstead [**1980**], which provides the equivalence between (B) and (C) above. We restate it here without proof.

4.1.4. Theorem. *Let E be a 0-expression and Ω an assignment from Tp. Let $[E]_\Omega$ be the result of computing E under Ω as in Kleene* [**1978**]. *Then* $[E]_\Omega \simeq [E]_{\#\Omega}$. □

4.2 *The image of Kleene's* 1959 *theory*. We aim to show that the enumerating functions (whence all recursive functions) of Kleene [**1959**] are the pull-backs in Tp of recursive functions. We call a function $\varphi\colon Tp^{(\sigma)} \underset{p}{\to} \mathbf{N}$ *59-recursive* (*59-recursive in* Θ) if φ is partial recursive (in Θ) in the sense of Kleene [**1959**], and write $\{e\}_{59}(\mathfrak{A})$ ($\{e\}_{59}^{\Theta}(\mathfrak{A})$) for the 59-recursive (59-recursive in Θ) function with index e. An operator $D\colon Tp^{(\sigma)} \to (Tp^{(j)} \underset{p}{\to} \mathbf{N})$ will be called *59-recursive* if there is a 59-recursive

φ: $Tp^{(j)} \times Tp^{(\sigma)} \underset{p}{\to} \mathbf{N}$ such that for all $\mathfrak{A} \in Tp^{(\sigma)}$

$$D(\mathfrak{A}) = \lambda\beta^j\varphi(\beta^j, \mathfrak{A}).$$

We will see in §5 that we cannot code arbitrary finite sequences from $\hat{T}p^{(j)}$ as elements of $\hat{T}p^{(j)}$ (we will have to move up to $\hat{T}p^{(j+1)}$); but there is sufficient coding capability for our immediate needs. The following definitions should be thought of as being carried out over $\hat{T}p$ (although they are formally the same as the corresponding definitions over Tp from Kleene [**1959**]).

$$p_i \simeq \text{ the } (i+1)\text{st prime},$$
$$\langle a_0,\ldots,a_{n-1}\rangle \simeq \prod_{i \leqslant n-1} p_i^{a_i} \quad (\text{undefined if any } a_i \text{ is } \mathrm{u}),$$
$$(\langle\ \rangle \simeq 1),$$
$$(a)_i \simeq \text{ the largest } n \text{ such that } p_i^n \text{ divides } a.$$

For $j \geqslant 1$:

$$\langle \alpha_0^j,\ldots,\alpha_{n-1}^j\rangle \simeq \lambda\beta^{j-1}\langle \alpha_0^j(\beta^{j-1}),\ldots,\alpha_{n-1}^j(\beta^{j-1})\rangle,$$
$$(\alpha^j)_i \simeq \lambda\beta^{j-1}\cdot\big(\alpha^j(\beta^{j-1})\big)_i.$$

These operators should really carry superscripts to indicate the type of their arguments, but this seems to be entirely too pedantic. Clearly (via Lemma 2.2.4) all of these functions and operators are recursive.

As we remarked above, these definitions may be used in either type structure. We shall do so and use the same notation in both cases, relying on context to make the distinction.

We do not, in general, have that $(\langle\hat{\alpha}_0^j,\ldots,\hat{\alpha}_n^j\rangle)_i = \hat{\alpha}_i^j$ (e.g., $(\langle 0, \mathrm{u}\rangle)_0 \uparrow$). Fortunately, we will only need to worry about these coding functions at arguments from $U^{(j)}$, and we do have the following:

4.2.1. Lemma. *For* $\alpha^j, \alpha_0^j,\ldots,\alpha_n^j \in Tp^{(j)}$:
(i) $\langle \#\alpha_0^j,\ldots,\#\alpha_n^j\rangle = \#\langle\alpha_0^j,\ldots,\alpha_n^j\rangle$.
(ii) $(\#\alpha^j)_i = \#(\alpha^j)_i$.
(iii) $(\langle\#\alpha_0^j,\ldots,\#\alpha_n^j\rangle)_i = \#\alpha_i^j$.

Proof. (i) and (ii) are easily verified using Lemma 4.1.3 and the definitions. (iii) follows from (i) and (ii) and the similar result from the classical theory. □

There is a good reason that we did not simply define these operators on Tp and then look at their images under $\#$. The image of a 59-recursive operator is not usually recursive—it is defined too often. Fortunately, for "nice" operators (like $\langle - \rangle$, etc.), the image is sufficiently close to being recursive. This is the motivation for the next definition.

4.2.2. Definition. A 59-recursive operator D: $Tp^{(\sigma)} \to (Tp^{(j)} \underset{p}{\to} \mathbf{N})$ is *embeddable* if there is a recursive operator $\hat{D}$: $\hat{T}p^{(\sigma)} \to \hat{T}p^{(j+1)}$ such that for all $\mathfrak{A} \in Tp^{(\sigma)}$

$$\hat{D}(\#\mathfrak{A}) = \#(D(\mathfrak{A})),$$

i.e., if the following diagram commutes:

$$\begin{tikzcd}
Tp^{(\sigma)} \arrow[r, "D"] \arrow[d, "\#"] & \left(Tp^{(j)} \underset{p}{\to} \mathbf{N}\right) \arrow[d, "\#"] \\
\hat{T}p^{(\sigma)} \arrow[r, "\hat{D}"] & \hat{T}p^{(j+1)}
\end{tikzcd}$$

(Recall that # is defined on all of $(Tp^{(j)} \underset{p}{\to} \mathbf{N})$—not just on $Tp^{(j+1)}$—and if $\varphi \in (Tp^{(j)} \underset{p}{\to} \mathbf{N})$ then $\#\varphi \in \hat{T}p^{(j+1)}$.) If $\hat{D}$ is associated with an embeddable operator D (as above), then $\hat{D}$ is called an *embedded* operator. □

The operators $\langle -, -, - \rangle$ and $(-)_i$ are embedded. While there will, in general, be an infinity of embedded operators $\hat{D}$ associated with a particular embeddable D, they will all agree on $U^{(\sigma)}$ and it is only there that we shall be concerned about their values. Thus, there should be no ambiguity if we use the same notation for both D and $\hat{D}$ (as we do with $\langle -, -, - \rangle$ and $(-)_i$). If the reader is bothered by this, he may console himself by the fact that there will always be a natural choice of a specific $\hat{D}$.

For notational convenience, we define three more embeddable operators Ins$(-, -, -)$, Del$(-, -)$ and Con$(-, -, -)$ which allow us to *ins*ert and *del*ete elements of coded sequences, and form *con*catenations.

4.2.3. DEFINITION.

$$\operatorname{Ins}(a, b, k) \simeq \left(\prod_{i<k} p_i^{(a)_i}\right) \cdot p_k^b \cdot \left(\prod_{k<i<a} p^{(a)_{i \dot{-} 1}}\right),$$

$$\operatorname{Del}(a, k) \simeq \left(\prod_{i<k} p_i^{(a)_i}\right) \cdot \left(\prod_{k \leqslant i<a} p_i^{(a)_{i+1}}\right),$$

$$\operatorname{Con}(a, b, k) \simeq \left(\prod_{i<k} p_i^{(a)_i}\right) \cdot \left(\prod_{k \leqslant i<b+a} p_i^{(b)_{i \dot{-} k}}\right).$$

Con(a, b, k) codes the concatenation of the first k elements of (the sequence coded by) a with (the sequence coded by) b. The extra parameter k is needed because this particular coding scheme does not recognize a terminal string of zeros in a, which we may wish to preserve.

For $j \geqslant 1$:

$$\operatorname{Ins}(\alpha^j, \beta^j, k) = \lambda\gamma^{j-1}\operatorname{Ins}(\alpha^j(\gamma^{j-1}), \beta^j(\gamma^{j-1}), k),$$

$$\operatorname{Del}(\alpha^j, k) = \lambda\gamma^{j-1}\operatorname{Del}(\alpha^j(\gamma^{j-1}), k),$$

$$\operatorname{Con}(\alpha^j, \beta^j, k) = \lambda\gamma^{j-1}\operatorname{Con}(\alpha^j(\gamma^{j-1}), \beta^j(\gamma^{j-1}), k).$$

Clearly, these operators behave as desired on $U^{(j,j,0)}$ (though not on all of $\hat{T}p^{(j,j,0)}$). For example,

$$\operatorname{Ins}(\langle \#\alpha_0^j, \#\alpha_1^j, \#\alpha_2^j \rangle, \#\beta^j, 1) = \langle \#\alpha_0^j, \#\beta^j, \#\alpha_1^j, \#\alpha_2^j \rangle$$

and

$$\operatorname{Del}(\langle \#\alpha_0^j, \#\alpha_1^j, \#\alpha_2^j \rangle, 1) = \langle \#\alpha_0^j, \#\alpha_2^j \rangle.$$

As usual, we will need operators u_j^n, d_j^n which raise and lower the types of objects. Both definitions may be applied in either type structure (and are standard on Tp), but only d_j^n will be embeddable.

4.2.4. Definition. u_j^{j+1} and d_j^{j+1} are defined by recursion on j.

$$u_0^1(\alpha^0) = \lambda\gamma^0[\alpha^0],$$
$$d_0^1(\alpha^1) = \alpha^1(0).$$

For $j \geqslant 1$:

$$u_j^{j+1}(\alpha^j) = \lambda\gamma^j\big[\alpha^j\big(d_{j-1}^j(\gamma^j)\big)\big],$$
$$d_j^{j+1}(\alpha^{j+1}) = \lambda\gamma^{j-1}\big[\alpha^{j+1}\big(u_{j-1}^j(\gamma^{j-1})\big)\big].$$

We define u_j^{j+k} and d_j^{j+k} by iteration.

$$u_j^j(\alpha^j) = d_j^j(\alpha^j) = \alpha^j,$$
$$u_j^{j+k+1}(\alpha^j) = u_{j+k}^{j+k+1}\big(u_{j+k-1}^{j+k}\big(\cdots\big(u_j^{j+1}(\alpha^j)\big)\cdots\big)\big),$$
$$d_j^{j+k+1}(\alpha^{j+k+1}) = d_j^{j+1}\big(d_{j+1}^{j+2}\big(\cdots\big(d_{j+k}^{j+k+1}(\alpha^{j+k+1})\big)\cdots\big)\big).$$

We will write simply $u(\alpha^j)$ and $d(\alpha^{j+1})$ for $u_j^{j+1}(\alpha^j)$ and $d_j^{j+1}(\alpha^{j+1})$. □

The next lemma summarizes the basic properties of u and d. Clearly u and d are recursive; they are also 59-recursive, but we do not need this fact.

4.2.5. Lemma. *For all* $\beta^j \in Tp$ *and* $\hat{\beta}^j, \hat{\gamma}^j \in \hat{T}p$:

(i) $d(u(\beta^j)) = \beta^j$ *and* $d(u(\hat{\beta}^j)) = \hat{\beta}^j$,

(ii) (*for* $j \geqslant 1$) $\hat{\beta}^j \subset \hat{\gamma}^j \overset{\Rightarrow}{\not\Leftarrow} d(\hat{\beta}^j) \subset d(\hat{\gamma}^j)$,

(iii) $\hat{\beta}^j \subset \hat{\gamma}^j \Leftrightarrow u(\hat{\beta}^j) \subset u(\hat{\gamma}^j)$.

Proof. (i) is proven by an easy induction on j. (ii) and the " $\Rightarrow$ " part of (iii) follow from the monotonicity of recursive operators (see p. 80). The " $\Leftarrow$ " part of (iii) follows by applying (ii) and then (i). □

In the next lemma, we are really interested in parts (i) and (iii). The other parts make the proof go smoothly. Part (i) says that d is embeddable.

4.2.6. Lemma. *For all* $\beta^j, \beta^{j+1} \in Tp$ *and all* $\hat{\beta}^j, \hat{\beta}^{j+1} \in \hat{T}p$:

(i) $\#d(\beta^{j+1}) = d(\#\beta^{j+1})$,

(ii) $\#u(\beta^j) \subsetneq u(\#\beta^j)$,

(iii) $\hat{\beta}^{j+1} \in U^{(j+1)} \Rightarrow d(\hat{\beta}^{j+1}) \in U^{(j)}$,

(iv) $\hat{\beta}^{j+1} \in V^{(j+1)} \Rightarrow d(\hat{\beta}^{j+1}) \in V^{(j)}$,

(v) $\hat{\beta}^j \in V^{(j)} \Leftrightarrow u(\hat{\beta}^j) \in V^{(j+1)}$.

Proof. The proof is simultaneous, by induction on j. The $j = 0$ case is trivial, so let $j \geqslant 1$.

(i) We first note that

$$\begin{aligned}\operatorname{dom}\big(d(\#\beta^{j+1})\big) &= \{\hat{\gamma}^{j-1}\colon u(\hat{\gamma}^{j-1}) \in V^{(j)}\} \quad \text{(Lemma 4.1.3)}\\ &= \{\hat{\gamma}^{j-1}\colon \hat{\gamma}^{j-1} \in V^{(j-1)}\} \quad \text{(ind, (v))}\\ &= \operatorname{dom}\big(\#d(\beta^{j+1})\big) \quad \text{(Lemma 4.1.3).}\end{aligned}$$

Thus, it suffices to show that $\#d(\beta^{j+1})$ and $d(\#\beta^{j+1})$ agree on $U^{(j-1)}$ (since their values on $V^{(j-1)}$ are determined by their values on $U^{(j-1)}$, by monotonicity). Now

$$\begin{aligned}\{\#d(\beta^{j+1})\}(\#\gamma^{j-1}) &= \{d(\beta^{j+1})\}(\gamma^{j-1})\\ &= \beta^{j+1}(u(\gamma^{j-1})) = \{\#\beta^{j+1}\}(\#u(\gamma^{j-1}))\\ &= \{\#\beta^{j+1}\}(u(\#\gamma^{j-1})) \quad \text{(ind, (ii))}\\ &= \{d(\#\beta^{j+1})\}(\#\gamma^{j-1}).\end{aligned}$$

(ii) By Lemma 4.1.3, $\mathrm{dom}(\#u(\beta^j)) = V^{(j)}$. Again, the values of $u(\#\beta^j)$ and $\#u(\beta^j)$ on $V^{(j)}$ are determined by their values on $U^{(j)}$. Now

$$\begin{aligned}\{\#u(\beta^j)\}(\#\gamma^j) &= \{u(\beta^j)\}(\gamma^j)\\ &= \beta^j(d(\gamma^j)) = \{\#\beta^j\}(\#d(\gamma^j))\\ &= \{\#\beta^j\}(d(\#\gamma^j)) \quad \text{(ind, (i))}\\ &= \{u(\#\beta^j)\}(\#\gamma^j).\end{aligned}$$

(iii) This is immediate from (i).

(iv) Let $\#\gamma^{j+1} \subset \hat{\beta}^{j+1}$. Then $\#d(\gamma^{j+1}) = d(\#\gamma^{j+1}) \subset d(\hat{\beta}^{j+1})$. (The containment is given by Lemma 4.2.5.)

(v) Suppose $\#\gamma^j \subset \hat{\beta}^j$. Then $\#u(\gamma^j) \subset u(\#\gamma^j) \subset u(\hat{\beta}^j)$. Conversely, $u(\hat{\beta}^j) \in V^{(j+1)} \Rightarrow d(u(\hat{\beta}^j)) \in V^{(j)} \Rightarrow \hat{\beta}^j \in V^{(j)}$. □

Kleene [**1959**, p. 22] defined $\{z\}_{59}[a, \alpha^1, \ldots, \alpha^r]$ to be

$$\{z\}_{59}((a)_0, \ldots, (a)_{n_0-1}, (\alpha^1)_0, \ldots, (\alpha^1)_{n_1-1}, \ldots, (\alpha^r)_{n_r-1})$$

if z is the 59-index of a function of n_j type-j arguments (for each $j \leqslant r$) and no arguments of type $> r$; otherwise $\{z\}_{59}[a, \alpha^1, \ldots, \alpha^r]$ is undefined.

4.2.7. Lemma. *Let* $\Theta = (\theta_1, \ldots, \theta_l)$ *be a list of functions with* θ_t: $Tp^{(\sigma_t)} \underset{p}{\rightarrow} \mathbf{N}$. *Then, for each* $r \geqslant 0$, *there is a recursive functional* $\bar{\chi}_r$ *such that for all* $z, a, \alpha^1, \ldots, \alpha^r \in Tp$,

$$\bar{\chi}_r(\#\Theta;\, z, a, \#\alpha^1, \ldots, \#\alpha^r) \simeq \{z\}^{\Theta}_{59}[a, \alpha^1, \ldots, \alpha^r].$$

Proof. Let $s = \max\{\max \mathrm{tp}(\sigma_t)\colon t \leqslant l\}$. For each z, let r_z be the maximum type of any argument of $\{z\}^{\Theta}_{59}$ (if z is an index for a function 59-recursive in Θ). It will suffice to define, for each $r \geqslant s-1$, $\chi_r(\eta, \hat{\Theta};\, z, a, \hat{\alpha}^1, \ldots, \hat{\alpha}^r)$ such that for all $z, a, \alpha^1, \ldots, \alpha^r \in Tp$

$$\chi^{\infty}_r(\#\Theta;\, z, a, \#\alpha^1, \ldots, \#\alpha^{r_z}, \mathrm{u}^{r_z+1}, \ldots, \mathrm{u}^r) \simeq \{z\}^{\Theta}_{59}[a, \alpha^1, \ldots, \alpha^r],$$

since we may then take, for $r < s-1$,

$$\bar{\chi}_r(\hat{\Theta};\, z, a, \hat{\alpha}^1, \ldots, \hat{\alpha}^r) \simeq \begin{cases} \chi^{\infty}_{s-1}(z, a, \hat{\alpha}^1, \ldots, \hat{\alpha}^r, \mathrm{u}^{r+1}, \ldots, \mathrm{u}^{s-1}) \\ \qquad\qquad \text{if } z \text{ is a 59-index for a function} \\ \qquad\qquad \text{whose arguments have types} \leqslant r, \\ \text{undefined} \quad \text{otherwise.} \end{cases}$$

In what follows, we assume that $r \geqslant s - 1$. We write $\mathfrak{A}$ for $(a, \alpha^1, \dots, \alpha^r)$, and suppress the variables Θ, which remain fixed throughout. We also omit the subscript r on χ.

χ is defined by cases on z; but, as we are avoiding explicit use of indices whenever possible, we state the case hypothesis in terms of the form of $\{z\}_{59}(\mathfrak{B})$. Cases 0-9 correspond to the schemata S0–S9 of Kleene **[1959]**.

Case -1. z or a is $\mathfrak{u}$, or $z \in \mathbf{N}$ but z is not a 59-index for a function whose arguments are of types $\leqslant r$. Take $\chi(\eta; z, \hat{\mathfrak{A}}) \uparrow$.

Case 0. $\{z\}_{59}(-)$ is obtained via S0.t, say with

$$\{z\}_{59}(c, d, \mathfrak{B}) \simeq \theta_t\big(c, d, \lambda\beta^j\{e\}_{59}(\beta^j, \mathfrak{B}), \lambda\beta^k\{f\}_{59}(\beta^k, \mathfrak{B})\big)$$

where $j \leqslant r_z$ and $r_z < k \leqslant s - 1$. (There are really l separate cases here—one for each θ_t.) Clearly t, e, f, etc., may be extracted from z. Take

$$\begin{aligned}\chi(\eta; z, \hat{\mathfrak{A}}) \simeq \hat{\theta}_t\Big[(a)_0, (a)_1, \lambda\hat{\beta}^j \cdot \eta\big[e, \mathrm{Del}(\mathrm{Del}(a,0),0), \hat{\alpha}^1, \dots,&\\ \hat{\alpha}^{j-1}, \mathrm{Con}(\langle\hat{\beta}^j\rangle, \hat{\alpha}^j, 1), \hat{\alpha}^{j+1}, \dots, \hat{\alpha}^r\big],&\\ \lambda\hat{\beta}^k \cdot \eta\Big[f, \mathrm{Del}(\mathrm{Del}(a,0),0), \hat{\alpha}^1, \dots, \hat{\alpha}^{r_z},&\\ d^k_{r_z+1}(\hat{\beta}^k), \dots, d^k_{k-1}(\hat{\beta}^k), \hat{\beta}^k, \hat{\alpha}^{k+1}, \dots, \hat{\alpha}^r\Big]\Big].&\end{aligned}$$

The reason that j and k must be handled differently is that $k = r_f > r_z$, so when we come to prove that χ works, some of the $\hat{\alpha}_i$ with $r_z < i < r_f$ may be $\mathfrak{u}^i$, but we will need elements of $U^{(i)}$ in these positions. These elements will be produced by $d^k_i(\#\beta^j)$. All of this should become clear shortly.

Case 1. $\{z\}_{59}(c, \mathfrak{B}) \simeq c + 1$. Take $\chi(\eta; z, \hat{\mathfrak{A}}) \simeq (a)_0 + 1$.

Case 2,3. Similarly.

Case 4. $\{z\}_{59}(\mathfrak{B}) \simeq \{g\}_{59}(\{h\}_{59}(\mathfrak{B}), \mathfrak{B})$. Take

$$\chi(\eta; z, \hat{\mathfrak{A}}) \simeq \eta\big(g, \mathrm{Con}(\langle\eta(h, \hat{\mathfrak{A}})\rangle, a, 1), \hat{\alpha}^1, \dots, \hat{\alpha}^r\big).$$

Case 5.

$$\begin{cases}\{z\}_{59}(0, \mathfrak{B}) \simeq \{g\}_{59}(\mathfrak{B}),\\ \{z\}_{59}(c + 1, \mathfrak{B}) \simeq \{h\}_{59}(c, \{z\}_{59}(c, \mathfrak{B}), \mathfrak{B}).\end{cases}$$

Take

$$\chi(\eta; z, \hat{\mathfrak{A}}) \simeq \begin{cases}\eta\big(g, \mathrm{Del}(a,0), \hat{\alpha}^1, \dots, \hat{\alpha}^r\big) \quad \text{if } (a)_0 = 0,\\ \eta\big[h, \mathrm{Con}(\langle(a)_0 \dot{-} 1, \eta(z, a \div 2, \hat{\alpha}^1, \dots, \hat{\alpha}^r)\rangle,\\ \qquad\qquad \mathrm{Del}(a,0),2), \hat{\alpha}^1, \dots, \hat{\alpha}^r\big] \quad \text{if } (a)_0 > 0.\end{cases}$$

Case 6. $\{z\}_{59}(\mathfrak{B}) \simeq \{g\}_{59}(\mathfrak{B}_1)$ where $\mathfrak{B}$ comes from $\mathfrak{B}_1$ by moving the $(k + 1)$st type-j variable to the front of the list. Take

$$\chi(\eta; z, \hat{\mathfrak{A}}) \simeq \eta\big(g, a, \hat{\alpha}^1, \dots, \hat{\alpha}^{j-1}, \mathrm{Ins}(\mathrm{Del}(\hat{\alpha}^j, 0), (\hat{\alpha}^j)_0, k), \hat{\alpha}^{j+1}, \dots, \hat{\alpha}^r\big).$$

Case 7. $\{z\}_{59}(\gamma^1, c, \mathfrak{B}) \simeq \gamma^1(c)$. Take $\chi(\eta; z, \hat{\mathfrak{A}}) \simeq \{(\hat{\alpha}^1)_0\}((a)_0)$.

Case 8. $\{z\}_{59}(\gamma^j, \mathfrak{B}) \simeq \gamma^j(\lambda\gamma^{j-2}\{h\}_{59}(\gamma^{j-2}, \gamma^j, \mathfrak{B}))$. Take

$$\chi(\eta;\, z, \hat{\mathfrak{A}}) \simeq (\alpha^j)_0\big[\lambda\hat{\gamma}^{j-2}\cdot \eta\big(h, a, \hat{\alpha}^1,\ldots,\hat{\alpha}^{j-3},\\ \mathrm{Con}(\langle\hat{\gamma}^{j-2}\rangle, \hat{\alpha}^{j-2}, 1), \hat{\alpha}^{j-1},\ldots,\hat{\alpha}^r)\big].$$

Case 9. $\{z\}_{59}(d, \mathfrak{B}, \mathfrak{C}) \simeq \{d\}_{59}(\mathfrak{B})$. Take

$$\chi(\eta;\, z, \hat{\mathfrak{A}}) \simeq \eta\big((a)_0, \mathrm{Del}(a,0), \hat{\alpha}^1,\ldots,\hat{\alpha}^r\big).$$

This completes the definition of χ. We now show that for all $(z, \mathfrak{A}) \in Tp^{(0,0,1,\ldots,r)}$

$$\chi^{\infty}(\#\Theta;\, z, a, \#\alpha^1,\ldots,\#\alpha^{r_z}, \mathrm{u}^{r_z+1},\ldots,\mathrm{u}^r) \simeq \{z\}^{\Theta}_{59}[\mathfrak{A}].$$

($\subset$) We cannot use Lemma 2.1.1 since, in general, $\#(\lambda z\mathfrak{A}\cdot\{z\}^{\Theta}_{59}[\mathfrak{A}])$ will not be a fixed point of χ (but the differences will be at arguments outside of $U^{(0,0,1,\ldots,r)}$). Instead, we show by induction on ξ that for all $(z, \mathfrak{A}) \in Tp^{(0,0,1,\ldots,r)}$

$$\chi^{\xi}(\#\Theta;\, z, \#\mathfrak{A}) = w \in \mathbf{N} \Rightarrow \{z\}^{\Theta}_{59}[\mathfrak{A}] = w.$$

This is a slightly stronger statement than we need, since

$$\chi^{\xi}(\#\Theta;\, z, a, \#\alpha^1,\ldots,\#\alpha^{r_z}, \mathrm{u}^{r_z+1},\ldots,\mathrm{u}^r) = w \Rightarrow \chi^{\xi}(\#\Theta;\, z, \#\mathfrak{A}) = w,$$

but it is notationally more convenient. Again, we frequently suppress the variables Θ. Suppose

$$\chi^{\xi}(\#\Theta;\, z, \#\mathfrak{A}) = w.$$

We take cases on z. Let us consider four representative cases; the others are similar.

Case 0. $\{z\}_{59}(-)$ is as in the exemplary case treated in the definition of χ. Here

$$(*)\quad \{z\}_{59}[\mathfrak{A}] \simeq \theta_t\Big((a)_0, (a)_1, \lambda\beta^j\cdot\{e\}_{59}\big[\mathrm{Del}(\mathrm{Del}(a,0),0), \alpha^1,\ldots,\\ \alpha^{j-1}, \mathrm{Con}(\langle\beta^j\rangle, \alpha^j, 1), \alpha^{j+1},\ldots,\alpha^r\big],\\ \lambda\beta^k\cdot\{f\}_{59}\big[\mathrm{Del}(\mathrm{Del}(a,0),0), \alpha^1,\ldots,\alpha^{r_z},\\ d^k_{r_z+1}(\beta^k),\ldots,d^k_{k-1}(\beta^k), \beta^k, \alpha^{k+1},\ldots,\alpha^r\big]\Big).$$

The reason that we may use $d^k_{r_z+i}(\beta^k)$ instead of α^{r_z+i} is that the value of $\{f\}_{59}[-]$ is independent of the arguments of types between r_z and k, since $\{f\}_{59}(-)$ has no such arguments.

Now

$$w = \chi^{\xi}(\#\Theta;\, z, \#\mathfrak{A}) = \chi(\chi^{<\xi}, \#\Theta;\, z, \#\mathfrak{A}) = \#\theta_t(-)$$

where $(-)$ is obtained by substituting $\chi^{<\xi}$, $\#\mathfrak{A}$ for η, $\hat{\mathfrak{A}}$ in the definition of χ (we prefer not to rewrite it here). But $\#\theta_t$ is only defined when its arguments come from $V^{(0,0,j+1,k+1)}$, so there are $\gamma^{j+1}, \gamma^{k+1} \in Tp$ such that

$$\theta_t\big((a)_0, (a)_1, \gamma^{j+1}, \gamma^{k+1}\big) = w,$$

$$\#\gamma^{j+1} \subset \lambda\hat{\beta}^j\cdot\chi^{<\xi}\big(e, \mathrm{Del}(\mathrm{Del}(a,0),0), \#\alpha^1,\ldots,\#\alpha^{j-1},\\ \mathrm{Con}(\langle\hat{\beta}^j\rangle, \#\alpha^j, 1), \#\alpha^{j+1},\ldots,\#\alpha^r\big),$$

and

$$\#\gamma^{k+1} \subset \lambda\hat{\beta}^k \cdot \chi^{<\xi}\Big(f, \mathrm{Del}(\mathrm{Del}(a,0),0), \#\alpha^1,\dots,\#\alpha^{r_z},$$
$$d^k_{r_z+1}(\hat{\beta}^k),\dots,d^k_{k-1}(\hat{\beta}^k), \hat{\beta}^k, \#\alpha^{k+1},\dots,\#\alpha^r\Big).$$

By the inductive hypothesis and the fact that Con($---$) and $\langle - \rangle$ are embeddable (so $\mathrm{Con}(\langle \#\beta^j\rangle, \#\alpha^j, 1) = \#\mathrm{Con}(\langle\beta^j\rangle, \alpha^j, 1)$), we have that, for all $\beta^j \in Tp^{(j)}$,

$$\gamma^{j+1}(\beta^j) = \{\#\gamma^{j+1}\}(\#\beta^j) = \chi^{<\xi}(-)$$
$$= \{e\}_{59}\big[\mathrm{Del}(\mathrm{Del}(a,0),0), \alpha^1,\dots,\alpha^{j-1}, \mathrm{Con}(\langle\beta^j\rangle, \alpha^j, 1), \alpha^{j+1},\dots,\alpha^r\big].$$

Further, using the fact that $d^k_{r_z+i}(-)$ is embeddable, we have that, for all $\beta^k \in Tp^{(k)}$,

$$\gamma^{k+1}(\beta^k) = \{\#\gamma^{k+1}\}(\#\beta^k) = \chi^{<\xi}(-)$$
$$= \{f\}_{59}\Big[\mathrm{Del}(\mathrm{Del}(a,0),0), \alpha^1,\dots,\alpha^{r_z}, d^k_{r_z+1}(\beta^k),\dots,d^k_{k-1}(\beta^k),$$
$$\beta^k, \alpha^{k+1},\dots,\alpha^r\Big].$$

Putting these last two equations into (*), we obtain

$$\{z\}_{59}[\mathfrak{A}] \simeq \theta_t((a)_0, (a)_1, \gamma^{j+1}, \gamma^{k+1}) = w.$$

Case 4. Here $\{z\}_{59}[\mathfrak{A}] \simeq \{g\}_{59}[\mathrm{Con}(\langle\{h\}_{59}[\mathfrak{A}]\rangle, a, 1), \alpha^1,\dots,\alpha^r]$. $w = \chi^{\xi}(z, \#\mathfrak{A}) = \chi^{<\xi}(g, \mathrm{Con}(\langle\chi^{<\xi}(h, \#\mathfrak{A})\rangle, a, 1), \#\alpha^1,\dots,\#\alpha^r)$. $\chi^{<\xi}$ is undefined if either of its type-$\hat{0}$ arguments is u, so $\chi^{<\xi}(h, \#\mathfrak{A})\downarrow$. By the inductive hypothesis

$$\chi^{<\xi}(h, \#\mathfrak{A}) = \{h\}_{59}[\mathfrak{A}].$$

Using this, and the inductive hypothesis again, we have

$$\chi^{<\xi}\big(g, \mathrm{Con}(\langle\chi^{<\xi}(h, \#\mathfrak{A})\rangle, a, 1), \#\alpha^1,\dots,\#\alpha^r\big)$$
$$\simeq \{g\}_{59}\big[\mathrm{Con}(\langle\{h\}_{59}[\mathfrak{A}]\rangle, a, 1), \alpha^1,\dots,\alpha^r\big] \simeq \{z\}_{59}[\mathfrak{A}].$$

Case 6. Here

$$\{z\}_{59}[\mathfrak{A}] \simeq \{g\}_{59}\big[a, \alpha^1,\dots,\alpha^{j-1}, \mathrm{Ins}(\mathrm{Del}(\alpha^j,0), (\alpha^j)_0, k), \alpha^{j+1},\dots,\alpha^r\big].$$

Since $(-)_0$, Ins($-$) and Del($-$) are embeddable (i.e., $\#(\mathrm{Ins}(\mathrm{Del}(\alpha^j,0),(\alpha^j)_0, k)) = \mathrm{Ins}(\mathrm{Del}(\#\alpha^j,0),(\#\alpha^j)_0, k))$, the inductive hypothesis yields the result.

Case 8. This is just like Case 0, except that we do not have the problem of r_f being greater than r_z to complicate things.

($\supset$) One may now show, by induction on $\{z\}_{59}(-)$ (see Kleene [**1959**, p. 14]), that for all $(z, \mathfrak{A}) \in Tp^{(0,0,1,\dots,r)}$

$$\{z\}^{\Theta}_{59}[\mathfrak{A}] = w \Rightarrow \chi^{\infty}(\#\Theta; z, a, \#\alpha^1,\dots,\#\alpha^{r_z}, \mathrm{u}^{r_z+1},\dots,\mathrm{u}^r) = w.$$

Again, we take cases on z. The cases are just the reverses of the corresponding cases for "$\subset$". The only place where the arguments u^{r_z+i} (instead of $\#\alpha^{r_z+i}$) could possibly cause trouble is at Case 0. We sketch this case.

Case 0. $\{z\}_{59}(-)$ is as in the exemplary case treated in the definition of χ. Since $\{z\}_{59}(c, d, \mathfrak{B})\downarrow$, we have that, for some γ^{j+1},

$$\gamma^{j+1} = \lambda\beta^j \cdot \{e\}_{59}(\beta^j, \mathfrak{B}) = \lambda\beta^j \cdot \{e\}_{59}\big[\mathrm{Del}(\mathrm{Del}(a,0),0), \alpha^1,\dots,\alpha^{j-1},$$
$$\mathrm{Con}(\langle\beta^j\rangle, \alpha^j, 1), \alpha^{j+1},\dots,\alpha^r\big]$$

is total. Now $r_e = r_z$ (since $j \leqslant r_z$), so the inductive hypothesis yields that, for each $\beta^j \in Tp^{(j)}$,

$$\chi^\infty\big(e, \mathrm{Del}(\mathrm{Del}(a,0),0), \#\alpha^1,\dots, \#\alpha^{j-1}, \#\mathrm{Con}(\langle\beta^j\rangle, \alpha^j, 1),$$
$$\#\alpha^{j+1},\dots, \#\alpha^{r_z}, \mathrm{u}^{r_z+1},\dots,\mathrm{u}^r\big) = \gamma^{j+1}(\beta^j).$$

Notice that $j \leqslant r_z$ is necessary in order for the left side of the last equation to be meaningful.

Similarly, for some γ^{k+1},

$$\begin{aligned}\gamma^{k+1} &= \lambda\beta^k \cdot \{f\}_{59}(\beta^k, \mathfrak{B})\\ &= \lambda\beta^k \cdot \{f\}_{59}\Big[\mathrm{Del}(\mathrm{Del}(a,0),0), \alpha^1,\dots,\alpha^{r_z},\\ &\qquad d^k_{r_z+1}(\beta^k),\dots,d^k_{k-1}(\beta^k), \beta^k, \alpha^{k+1},\dots,\alpha^r\Big]\end{aligned}$$

is total. Now $r_f = k$ (since $k > r_z$), so the inductive hypothesis yields that, for all $\beta^k \in Tp^{(k)}$,

$$\chi^\infty\big(f, \mathrm{Del}(\mathrm{Del}(a,0)0), \#\alpha^1,\dots, \#\alpha^{r_z},$$
$$\#d^k_{r_z+1}(\beta^k),\dots, \#d^k_{k-1}(\beta^k), \#\beta^k, \mathrm{u}^{k+1},\dots,\mathrm{u}^r\big) = \gamma^{k+1}(\beta^k).$$

Putting these facts together with the embeddability of Con$(-)$, $\langle - \rangle$ and $d^k_{r_z+i}(-)$, and substituting into the definition of χ, we have that

$$\chi\big(\chi^\infty; \#\Theta; z, a, \#\alpha^1,\dots, \#\alpha^{r_z}, \mathrm{u}^{r_z+1},\dots,\mathrm{u}^r\big) = w.$$

Finally, we apply Lemma 2.1.1 to obtain $\chi^\infty(\#\Theta; ---) = w$ as desired.

Why did we treat j and k differently in the definition of χ? If we had not, we would have needed to known that for all β^k

$$\chi^\infty\big(f, \mathrm{Del}(\mathrm{Del}(a,0),0), \#\alpha^1,\dots, \#\alpha^{r_z}, \mathrm{u}^{r_z+1},\dots,\mathrm{u}^r\big) = \gamma^{k+1}(\beta^k).$$

But this would not have followed from the inductive hypothesis. We needed $\#\beta^k$ in the kth position, and we needed elements of $U^{(i)}$ in the ith position, for each $i \leqslant k$. These elements were produced by $d^k_i(\#\beta^k)$. □

4.2.8. Theorem. *Let* $\varphi(\mathfrak{A})$ *be* 59*-recursive in* Θ. *Then there is a recursive* $\psi(\hat{\Theta}; \hat{\mathfrak{A}})$ *such that for all* $\mathfrak{A} \in Tp$

$$\psi(\#\Theta; \#\mathfrak{A}) \simeq \varphi(\mathfrak{A}).$$

Proof. Say $\varphi(\mathfrak{A})$ is $\varphi(a, \alpha^1, \beta^1, \alpha^4, \beta^4)$. Let z be a 59-index of φ. With $\bar{\chi}_4$ as in Lemma 4.2.7, set

$$\psi(\hat{\Theta}; a, \hat{\alpha}^1, \hat{\beta}^1, \hat{\alpha}^4, \hat{\beta}^4) \simeq \bar{\chi}_4\big(\hat{\Theta}; z, a, \langle\alpha^1, \beta^1\rangle, d^4_2(\hat{\alpha}^4), d^4_3(\hat{\alpha}^4), \langle\hat{\alpha}^4, \hat{\beta}^4\rangle\big).$$

The result now follows from Lemma 4.2.7 and the embeddability of $\langle - \rangle$ and $d^k_j(-)$. □

Remark. Our ability to define $\bar{\chi}_r$, even when Θ might have arguments of types $> r$, hinged on our use of u^{r+i}. We had to be able to produce some type-$\widehat{r + i}$ object to be an argument for χ^∞_{s-1}; and we had to be able to use objects which were not in $U^{(r+i)}$ since elements of $U^{(r+i)}$ could only be produced if an element of $U^{(r+i+j)}$

were available (for some j). If each of Θ were total, then we could have used Θ to produce an element of $U^{(s+1)}$. Thus the only reason for the added confusion of the u^{r+i} is that we wanted the full generality of relativization of partial Θ.

4.3 *The* 59-*recursiveness of the pull-backs of recursive functionals.* The proof of the next lemma could be simplified somewhat by using Theorem LXI of Kleene [**1963**], which provides for restricted substitution in 59-recursive functions. But there was a great deal of machinery used to prove this result, and with very little extra work we can avoid this machinery—all of the machinery is already a part of the computations of §1. The most general form of substitution (Theorem 4.4.3) will come out of our work in this section and in Theorem 4.2.8.

Let us suppose that some reasonable Gödel numbering of j-expressions has been chosen. We denote the Gödel number of A by $\ulcorner A \urcorner$.

For any 0-expression E, let Γ_E be the list of (formal) free variables of E in the order in which they occur. Let $\mathfrak{A}_E$ be a list of "informal" variables of the same number and types as Γ_E.

4.3.1. Lemma. *For each list* Θ, *there is a primitive recursive function* $\pi(z, e)$ *such that: if* z *is the index of a derivation from* Θ *and* e *is the Gödel number of a* 0-*expression* E *based on this derivation, then* $\pi(z, e)$ *is a* 59-*index for a function* (*rec. in* Θ) *of the variables* $\mathfrak{A}_E$ *and for all* $\mathfrak{A}_E \in Tp^{(\sigma)}$ (*where* $\sigma = \mathrm{ch}(\mathfrak{A}_E)$)

$$\{\pi(z, e)\}^{\Theta}_{59}(\mathfrak{A}_E) \simeq [E]_{\Omega[\Gamma_E/\#\mathfrak{A}_E]}$$

(*where* Ω *is any assignment which assigns the functions* $\#\Theta$ *to the function variables of* E).

Proof.[9] We will choose a 59-index p for π via the Second Recursion Theorem (Kleene [**1959**, XIV]). For clarity, we will simply state what $\pi(z, e)$ must be in terms of p—which is only chosen after the fact.

We take cases on e. As always, we suppress Θ.

Case -1. z is not the index of a derivation or e is not the index of a 0-expression based on this derivation. Take $\pi(z, e)$ to be 0.

From now on, assume that Case -1 does not hold. We state the rest of the case hypotheses in terms of the form of E, where $\ulcorner E \urcorner = e$.

E1. E is γ^0. Take $\pi(z, e)$ to be a 59-index of $\lambda a[a]$.

E2. E is $\varphi_i(\vec{A})$. Let E^* be as in Definition 1.3.2, E2. Take $\pi(z, e)$ to be a 59-index of $\lambda \mathfrak{A}_E \cdot \{\pi(z, \ulcorner E^* \urcorner)\}_{59}(\mathfrak{A}_E)$. Let us spell this out a little more carefully (this once only). First pick f such that $\{f\}_{59}(d, \mathfrak{A}_E) \simeq \{d\}_{59}(\mathfrak{A}_E)$. Now, using the 59-index p to be chosen for π, choose h such that

$$\{h\}_{59}(\mathfrak{A}_E) \simeq \{f\}_{59}(\{p\}_{59}(z, \ulcorner E^* \urcorner), \mathfrak{A}_E),$$

and then take $\pi(z, e) = h$. We do not simply take $\pi(z, e) \simeq \pi(z, \ulcorner E^* \urcorner)$, because π would no longer be primitive recursive, nor even total.

[9] The proof consists of imitating the construction of the computation tree for E under $\Omega[\Gamma_E/\#\mathfrak{A}]$. Since, under such an assignment, the construction of the tree is completely deterministic, we are able to embody it in a 59-recursive function.

E4.λ. E is $\{\lambda\beta^j A\}(B)$. Similarly, take $\pi(z,e)$ to be a 59-index for $\lambda\mathfrak{A}_E \cdot \{\pi(z, \ulcorner S_B^{\beta^j}A\,|\urcorner)\}_{59}(\mathfrak{A}_E)$.

E4.1. E is $\gamma^1(B)$. Take $\pi(z, e)$ to be a 59-index for

$$\lambda\mathfrak{A}_E \cdot \alpha^1\big(\{\pi(z, \ulcorner B\urcorner)\}_{59}(\mathfrak{A}_B)\big);$$

where α^1 is the first variable in $\mathfrak{A}_E$.

E4. j ($j > 1$). E is $\gamma^j(B)$. Take $\pi(z, e)$ to be a 59-index for

$$\lambda\mathfrak{A}_E \cdot \alpha^j\Big(\lambda\alpha^{j-2} \cdot \Big\{\pi\Big(z, \ulcorner B\big(\gamma^{j-2}\big)\urcorner\Big)\Big\}_{59}\big(\mathfrak{A}_B, \alpha^{j-2}\big)\Big);$$

where α^j is the first variable in $\mathfrak{A}_E$ and α^{j-2} is a new variable. (Notice that $\mathfrak{A}_{B(\gamma^{j-2})} \equiv (\mathfrak{A}_B, \alpha^{j-2})$.)

E5. E is 0. Take $\pi(z, e)$ to be a 59-index for 0 (as a function of no arguments—see Kleene [**1959**, p. 3]).

E6 $+$, E6 $\dot{-}$. Similar.

E6cs. E is $\mathrm{cs}(A, B, C)$. We use the fact that an index for *strong definition by cases* may be obtained from indices of the fucntions and predicates involved (see Kleene [**1959**, XVIII]). Take $\pi(z, e)$ to be a 59-index for

$$\lambda\mathfrak{A}_E \cdot \begin{cases} \{\pi(z, \ulcorner B\urcorner)\}_{59}(\mathfrak{A}_B) & \text{if } \{\pi(z, \ulcorner A\urcorner)\}_{59}(\mathfrak{A}_A) = 0, \\ \{\pi(z, \ulcorner C\urcorner)\}_{59}(\mathfrak{A}_C) & \text{if } \{\pi(z, \ulcorner A\urcorner)\}_{59}(\mathfrak{A}_A) > 0. \end{cases}$$

(Of course, e.g., by $\mathfrak{A}_A$ we mean those arguments from $\mathfrak{A}_E$ which, in $\mathfrak{A}_E$, correspond to the variables of A.)

E7. Similarly to E4. j.

Now, once the 59-index p for $\pi(z, e)$ (which was used throughout) has been chosen via the Second Recursion Theorem, each value of $\pi(z, e)$ is specified in terms of primitive recursive functions of p, z and e. Thus, π is primitive recursive.

It remains to show that π does what was claimed.

($\subset$) We show by induction on $\{z\}(\text{—})$ (see Kleene [**1959**, p. 14]) that, if z and e are as in the statement of the lemma and $\{\pi(z, e)\}_{59}(\mathfrak{A}_E) = w$, then $[E]_{\Omega[\Gamma_E/\#\mathfrak{A}_E]} = w$. We take cases on the form of E. Let us consider only the case E4. j ($j > 1$).

E4. j ($j > 1$). The inductive hypothesis yields that for each $\alpha^{j-2} \in Tp^{(j-2)}$

$$\Big\{\pi\Big(z, \ulcorner B\big(\gamma^{j-2}\big)\urcorner\Big)\Big\}_{59}\big(\mathfrak{A}_B, \alpha^{j-2}\big) = \big[B\big(\gamma^{j-2}\big)\big]_{\Omega[\gamma^{j-2}/\#\alpha^{j-2}, \Gamma_B/\#\mathfrak{A}_B]}.$$

Thus

$$\begin{aligned} &\alpha^j\Big(\lambda\alpha^{j-2} \cdot \Big\{\pi\Big(z, \ulcorner B\big(\gamma^{j-2+}\big)\urcorner\Big)\Big\}_{59}\big(\mathfrak{A}_B, \alpha^{j-2}\big)\Big) \\ &\qquad\qquad \simeq \{\#\alpha^j\}\big([B]_{\Omega[\Gamma_B/\#\mathfrak{A}_B]}\big) \simeq [E]_{\Omega[\Gamma_E/\#\mathfrak{A}_E]}. \end{aligned}$$

($\supset$) We show by induction on $\xi = |E|_{\Omega[\Gamma_E/\#\mathfrak{A}_E]}$ that for all $\mathfrak{A}_E \in Tp^{(\sigma)}$, if $[E]_{\Omega[\Gamma_E/\#\mathfrak{A}_E]} = w$, then $\{\pi(z, e)\}_{59}(\mathfrak{A}_E) = w$. Again, we consider only one case.

E4. j ($j > 1$). Since $\#\alpha^j$ is only defined on $V^{(j-1)}$, we have some $\alpha^{j-1} \in Tp^{(j-1)}$ such that $\#\alpha^{j-1} \subset [B]^{<\xi}_{\Omega[\Gamma_B/\#\mathfrak{A}_B]}$ and $\alpha^j(\alpha^{j-1}) = w$. That is to say, for all $\alpha^{j-2} \in Tp^{(j-2)}$,

$$\big[B\big(\gamma^{j-2}\big)\big]^{<\xi}_{\Omega[\gamma^{j-2}/\#\alpha^{j-2}, \Gamma_B/\#\mathfrak{A}_B]} = \alpha^{j-1}\big(\alpha^{j-2}\big).$$

Now the inductive hypothesis applies to yield

$$\alpha^{j-1} = \lambda\alpha^{j-2} \cdot \left\{\pi\left(z, \ulcorner B\left(\gamma^{j-2}\right)\urcorner\right)\right\}_{59}\left(\mathfrak{A}_B, \alpha^{j-2}\right).$$

Thus

$$\{\pi(z, e)\}_{59}(\mathfrak{A}_E) = \alpha^j(\alpha^{j-1}) = w. \quad \square$$

4.3.2. Theorem. *Let Θ be a fixed list of functions on Tp. If $\varphi(\hat{\Theta}; \hat{\mathfrak{A}})$ is a recursive functional, then there is a function $\psi(\mathfrak{A})$, 59-recursive (uniformly) in Θ, such that for all $\mathfrak{A} \in Tp^{(\sigma)}$*

$$\psi(\mathfrak{A}) \simeq \varphi(\#\Theta; \#\mathfrak{A}).$$

Proof. Consider a derivation $\varphi_1, \ldots, \varphi_p = \varphi$. Now construct a Gödel number e for the 0-expression $\varphi(\mathfrak{A})$ and apply Lemma 4.3.1. $\square$

Putting Theorems 4.1.4, 4.2.8 and 4.3.2 together, we have, as promised:

4.3.3. Theorem. *Let φ: $Tp^{(\sigma)} \underset{p}{\rightarrow} \mathbf{N}$. Let Θ be a list of functions on Tp. Then the following are equivalent*:

(i) *φ is 59-recursive in Θ.*

(ii) *φ is recursive in Θ in the sense of Kleene* [**1978**].

(iii) *There is a recursive ψ such that for all $\mathfrak{A} \in Tp^{(\sigma)}$,*

$$\psi(\#\Theta; \#\mathfrak{A}) \simeq \varphi(\mathfrak{A}). \quad \square$$

We call such a ψ a *recursive functional corresponding to* φ.

4.4 *Some results in Kleene's* 1959 *theory*. While we have never seen the theorems of this section in print, the facts are undoubtedly known to any researcher in the field. It is our belief that the development above makes them particularly natural. For simplicity, we consider only the unrelativized versions.

The import of the next theorem is: If $\varphi(\mathfrak{A})$ is 59-recursive with $\mathfrak{A} \equiv (\alpha_1^{j_1}, \ldots, \alpha_n^{j_n})$ and $r = \max \operatorname{tp}(\mathfrak{A}) \geqslant 1$, then for each $\mathfrak{A} \in Tp^{(\sigma)}$ for which $\varphi(\mathfrak{A})\downarrow$, the value of $\varphi(\mathfrak{A})$ is determined by the type-0 arguments and the value of the higher type arguments at certain *minimum* subsets of their domains; moreover, if $j_i = r$ then the subset of $\operatorname{dom}(\alpha_i^{j_i})$ has strictly smaller cardinality than $\operatorname{dom}(\alpha_i^{j_i})$.

Recall that $\{z\}(-)$ is the recursive function whose derivation has index z and $\{z\}_{59}(-)$ is the 59-recursive function with 59-index z.

4.4.1. Theorem (Use lemma). *Let $\mathfrak{B} \in Tp^{(0,0,\ldots,0)}$ and $\mathfrak{A} \equiv (\alpha_1^{j_1}, \ldots, \alpha_n^{j_n}) \in Tp^{(\sigma)}$ with $r =$* max *$tp(\mathfrak{A}) \geqslant 1$ and each $j_i \geqslant 1$. Suppose that $\{e\}_{59}(\mathfrak{B}, \mathfrak{A})\downarrow$. Let $\{z\}(-)$ be a recursive function corresponding to $\{e\}_{59}(-)$. Then there are minimum partial subfunctions $\mathfrak{X} \equiv (\chi_1^{j_1}, \ldots, \chi_n^{j_n}) \subset \mathfrak{A}$ such that $\{z\}(\mathfrak{B}, \#\mathfrak{X})\downarrow$, whence, for all $\mathfrak{A}^* \in Tp^{(\sigma)}$,*

$$\mathfrak{X} \subset \mathfrak{A}^* \Rightarrow \{e\}_{59}(\mathfrak{B}, \mathfrak{A}^*) \simeq \{z\}(\mathfrak{B}, \#\mathfrak{X}) \simeq \{e\}_{59}(\mathfrak{B}, \mathfrak{A}).$$

Moreover, $\operatorname{Card}(\operatorname{dom}(\chi_i^{j_i})) \leqslant \beth_{r-2}$ (*finite if $r = 1$*).

Proof. Let $\varphi_1, \ldots, \varphi_p = \{z\}(-)$ be the derivation indexed by z. We prove by induction on $\xi = |E|_\Omega$ that, if E is a 0-expression all of whose free variables have type $\leqslant r$, and Ω assigns $\#\alpha_i^{j_i}$ to $\gamma_i^{j_i}$ and makes all of its other assignments from various $U^{(j)}$'s, and $[E]_\Omega\downarrow$, then there are $\mathfrak{X}$, as in the statement of the theorem, such that

$$[E]_{\Omega[\gamma_1^{j_1}/\#\chi_1^{j_1},\ldots,\gamma_n^{j_n}/\#\chi_n^{j_n}]} \simeq [E]_{\Omega}.$$

The theorem will follow by applying this to the 0-expression $\varphi_p(\mathfrak{B}, \gamma_1^{j_1}, \ldots, \gamma_n^{j_n})$.

We take cases on the form of E, considering only the E4. j ($j > 1$) case (the others are trivial).

E4. j ($j > 1$). E is $\beta^j(D)$. Note that $j \leqslant r$. Since Ω makes its assignments from U, there is a $\beta^{j-1} \in Tp^{(j-1)}$ such that $\#\beta^{j-1} \subset [D]_{\Omega}^{<\xi}$. Let γ^{j-2} be a new variable. For each $\delta^{j-2} \in Tp^{(j-2)}$,

$$\left[D\left(\gamma^{j-2}\right)\right]_{\Omega[\gamma^{j-2}/\#\delta^{j-2}]\downarrow}^{<\xi}.$$

Associated with each $\delta^{j-2} \in Tp^{(j-2)}$, we have (via the inductive hypothesis) subfunctions $\chi_{i,\delta}^{j_i}$ as in the theorem.

If β^j is not one of $\gamma_i^{j_i}$ we may take $\chi_i^{j_i} = \bigcup\{\chi_{i,\delta}^{j_i}: \delta \in Tp^{(j-2)}\}$. Then

$$\#\beta^{j-1} \subset [D]_{\Omega[\gamma_i^{j_i}/\#\chi_i^{j_i}]}^{<\xi}$$

so

$$\left[\beta^j(D)\right]_{\Omega[\gamma_i^{j_i}/\#\chi_i^{j_i}]}\downarrow \text{ (and } \simeq [E]_{\Omega}\text{)}.$$

Clearly any choice for $\chi_i^{j_i}$ which did not contain all of the $\chi_{i,\delta}^{j_i}$'s would cause one of the subcomputations to diverge, whence the entire computation would also diverge. Thus, our choices $\chi_i^{j_i}$ were minimum.

Finally,

$$\operatorname{Card}\left(\operatorname{Dom}\left(\chi_i^{j_i}\right)\right) \leqslant \sum_{\delta \in Tp^{(j-2)}} \operatorname{Card}\left(\operatorname{Dom}\left(\chi_{i,\delta}^{j_i}\right)\right) \leqslant \beth_{j-2} \cdot \beth_{r-2} \leqslant \beth_{r-2}.$$

If β^j is $\gamma_i^{j_i}$ for some i, then we must include the additional value $\chi_i^{j_i}(\beta^{j-1}) = \alpha_i^{j_i}(\beta^{j-1})$. □

Remark. The proof of Theorem 4.4.1 is no more than the formalization of the fact that the computation tree for $[E]_{\Omega}$ has cardinality $\leqslant \beth_{r-2}$ and the value of $[E]_{\Omega}$ is determined by the values of $\alpha_i^{j_i}$ at vertices of the form $\gamma_i^{j_i}(B)$ on the tree.

4.4.2. Lemma (In the unextended theory). *Let* $j \geqslant 1$ *and* $\max \operatorname{tp}(\sigma) \leqslant j$. *The predicate*

$$R(\alpha^j) \Leftrightarrow \forall \beta^{j-1}\left(\alpha^j(\beta^{j-1}) = 0\right)$$

is not 59*-semirecursive in any* $\mathfrak{B} \in Tp^{(\sigma)}$.

Proof. Suppose that, for some 59-recursive φ and some $\mathfrak{B} \in Tp^{(\sigma)}$,

$$R(\alpha^j) \Leftrightarrow \varphi(\alpha^j, \mathfrak{B})\downarrow.$$

Then $\varphi(\lambda\beta^{j-1} \cdot 0, \mathfrak{B})\downarrow$ and (by the use lemma) this fact is determined by the values of $\lambda\beta^{j-1} \cdot 0$ on at most $\beth_{j-2}$ arguments (finitely many if $j = 1$). Now $\operatorname{Card}(Tp^{(j-1)}) = \beth_{j-1}$ so there is a $\delta^j \in Tp^{(j)}$ which is 0 at these arguments and 1 at some other element of $Tp^{(j-1)}$. Then $\varphi(\delta^j, \mathfrak{B})\downarrow$ and $\neg R(\delta^j)$, which is a contradiction. □

One direction of the next theorem is Theorem 16.4 of Kechris and Moschovakis [**1977**], where it is attributed to Kleene. Both directions become trivialities in the current setting.

4.4.3. Theorem ($j \geqslant 1$). *Let $\theta(\beta^j, \mathfrak{B})$ be 59-recursive, $\sigma = \mathrm{ch}(\mathfrak{B})$, and suppose there is some $\mathfrak{B}_0 \in Tp^{(\sigma)}$ such that $\lambda\beta^j\theta(\beta^j, \mathfrak{B}_0)$ is total. Then the following are equivalent*:

(i) *For each 59-recursive $\psi_0(\gamma^{j+1}, \mathfrak{B}, \mathfrak{C})$ there is a 59-recursive $\psi(\mathfrak{B}, \mathfrak{C})$ such that, for values of $\mathfrak{B}$ and $\mathfrak{C}$ with $\lambda\beta^j\theta(\beta^j, \mathfrak{B})$ total,*[10]

$$\psi(\mathfrak{B}, \mathfrak{C}) \simeq \psi_0\big(\lambda\beta^j\theta(\beta^j, \mathfrak{B}), \mathfrak{B}, \mathfrak{C}\big).$$

(ii) $\max \mathrm{tp}(\mathfrak{B}) \geqslant j + 1$.

Proof. ((ii) $\Rightarrow$ (i)) Let α^k be one of the variables in $\mathfrak{B}$ of type $k \geqslant j + 1$. Choose recursive $\bar{\psi}_0$ and $\bar{\theta}$ corresponding to ψ_0 and θ respectively. Set

$$\bar{\theta}_0(\hat{\beta}^j, \hat{\mathfrak{B}}) \simeq \bar{\theta}(\hat{\beta}^j, \hat{\mathfrak{B}}) + cs\big(\{d^k_{j+1}(\hat{\alpha}^k)\}(\hat{\beta}^j), 0, 0\big).$$

Now, for all $\mathfrak{B} \in Tp^{(\sigma)}$ with $\lambda\beta^j\theta(\beta^j, \mathfrak{B})$ total and all $\beta^j \in Tp^{(j)}$,

$$\bar{\theta}_0(\#\beta^j, \#\mathfrak{B}) \simeq \bar{\theta}(\#\beta^j, \#\mathfrak{B}) \simeq \theta(\beta^j, \mathfrak{B}).$$

Thus, $V^{(j)} \subset \mathrm{Dom}(\lambda\hat{\beta}^j\bar{\theta}_0(\hat{\beta}^j, \#\mathfrak{B}))$. On the other hand, since d^k_{j+1} is embeddable, $\mathrm{Dom}(d^k_{j+1}(\#\alpha^k)) = V^{(j)}$. Thus $\mathrm{Dom}(\lambda\hat{\beta}^j\bar{\theta}_0(\hat{\beta}^j, \#\mathfrak{B})) = V^{(j)}$.

We have shown (via Lemma 4.1.3) that, for values of $\mathfrak{B} \in Tp^{(\sigma)}$ with $\lambda\beta^j\theta(\beta^j, \mathfrak{B})$ total,

$$\lambda\hat{\beta}^j\bar{\theta}_0(\hat{\beta}^j, \#\mathfrak{B}) = \#\big(\lambda\beta^j\theta(\beta^j, \mathfrak{B})\big).$$

Thus, taking

$$\bar{\psi}(\hat{\mathfrak{B}}, \hat{\mathfrak{C}}) \simeq \bar{\psi}_0\big(\lambda\hat{\beta}^j\bar{\theta}_0(\hat{\beta}^j, \hat{\mathfrak{B}}), \hat{\mathfrak{B}}, \hat{\mathfrak{C}}\big),$$

we have, for all $\mathfrak{B} \in Tp^{(\sigma)}$ and $\mathfrak{C} \in Tp^{(\tau)}$ with $\lambda\beta^j\theta(\beta^j, \mathfrak{B})$ total,

$$\bar{\psi}(\#\mathfrak{B}, \#\mathfrak{C}) \simeq \psi_0\big(\lambda\beta^j\theta(\beta^j, \mathfrak{B}), \mathfrak{B}, \mathfrak{C}\big).$$

By Theorem 4.3.3,

$$\psi(\mathfrak{B}, \mathfrak{C}) \simeq \bar{\psi}(\#\mathfrak{B}, \#\mathfrak{C})$$

is the desired 59-recursive function.

((i) $\Rightarrow$ (ii)) Suppose $\max tp(\mathfrak{B}) \leqslant j$ and $\mathfrak{B}_0 \in Tp^{(\sigma)}$ is such that $\lambda\beta^j\theta(\beta^j, \mathfrak{B}_0)$ is total. By Lemma 4.4.2, the predicate $\forall\delta^{j-1}(\beta^j(\delta^{j-1}) = 0)$ is not 59-semirecursive in $\mathfrak{B}_0$.

Let

$$\psi_0(\gamma^{j+1}, \beta^j, \mathfrak{B}) \simeq \gamma^{j+1}\left(\lambda\delta^{j-1}\begin{cases} 0 & \text{if } \beta^j(\delta^{j-1}) = 0 \\ \text{undefined} & \text{otherwise} \end{cases}\right).$$

Then $\psi_0(\gamma^{j+1}, \beta^j, \mathfrak{B})\downarrow \Leftrightarrow \forall\delta^{j-1}(\beta^j(\delta^{j-1}) = 0)$. Now suppose that for all β^j

$$\psi(\beta^j, \mathfrak{B}_0) \simeq \psi_0\big(\lambda\bar{\beta}^j\theta(\bar{\beta}^j, \mathfrak{B}_0), \beta^j, \mathfrak{B}_0\big).$$

Where ψ 59-recursive, then $\forall\delta^{j-1}(\beta^j(\delta^{j-1}) = 0)$ would be 59-semirecursive in $\mathfrak{B}_0$, which cannot be.

[10] If we did not require the existence of some $\mathfrak{B}_0$ for which $\lambda\beta^j\Theta(\beta^j, \mathfrak{B}_0)$ is total, this condition would be trivially satisfied by any $\psi(\mathfrak{B}, \mathfrak{C})$.

REMARK. The previous theorem formalizes the fact that: the substitution principle breaks down if the object being substituted for is used to test the totality of some computed function, and this problem may be circumvented if another object, of sufficiently high type, is available to perform this test.

5. Enumeration, $s - m - n$ and second recursion theorems on $\hat{Tp}$. We now leave Tp behind and work entirely within $\hat{Tp}$. In order to prove the Enumeration Theorem, we will need some coding apparatus. As we now mentioned in §4.2, we cannot code sequences from $\hat{Tp}^{(j)}$ as elements of $\hat{Tp}^{(j)}$. The problem is that the recursive functions are all monotone. For example, $\langle u, u\rangle$, $\langle u, 1\rangle$ and $\langle 1, u\rangle$ would all have to receive different codes in $\hat{Tp}^{(0)}$, whence at least two of them would receive different codes $a, b \in \mathbf{N}$; but then monotonicity would require that $\langle 1, 1\rangle = a$ and $\langle 1, 1\rangle = b$, which cannot be. In order to have acceptable codes for sequences from $\hat{Tp}^{(j)}$, we must move up to $\hat{Tp}^{(j+1)}$.

5.1 DEFINITION. We define the coding and decoding operators $\langle \alpha_1^j, \ldots, \alpha_n^j \rangle$ and $(\alpha^j)_i$ by recursion on j.

$$\langle \alpha_1^0, \ldots, \alpha_n^0 \rangle = \lambda\gamma^0 \begin{cases} n & \text{if } \gamma^0 = 0, \\ \alpha_i^0 & \text{if } \gamma^0 = i,\ 1 \leqslant i \leqslant n, \\ 0 & \text{if } \gamma^0 > n, \end{cases}$$

$$(\alpha^1)_i \simeq \alpha^1(i),$$

$(j \geqslant 1)$

$$\langle \alpha_1^j, \ldots, \alpha_n^j \rangle = \lambda\gamma^j \begin{cases} n & \text{if } d_0^{j-1}\big((\gamma^j)_1\big) = 0, \\ \alpha_i^j\big((\gamma^j)_2\big) & \text{if } d_0^{j-1}\big((\gamma^{j-1})_1\big) = i,\ 1 \leqslant i \leqslant n, \\ 0 & \text{if } d_0^{j-1}\big((\gamma^j)_1\big) > n, \end{cases}$$

$$(\alpha^{j+1})_i = \lambda\gamma^{j-1} \cdot \alpha^{j+1}\big(\langle u_0^{j-1}(i), \gamma^{j-1}\rangle\big). \quad \square$$

5.2 LEMMA. *If* $1 \leqslant i \leqslant n$ *then*

$$\big(\langle \alpha_1^j, \ldots, \alpha_n^j \rangle\big)_i = \alpha_i^j.$$

PROOF. Induct on j. $\square$

We need one operator to *alter* coded sequences. $\operatorname{Alt}(\alpha^{j+1}, \beta^j, i)$ will replace the ith entry in the sequence (coded by) α^{j+1} by β^j.

5.3 DEFINITION.

$$\operatorname{Alt}(\alpha^1, \beta^0, i) = \lambda\gamma^0 \begin{cases} \max\{i, \alpha^1(0)\} & \text{if } \gamma^0 = 0, \\ \alpha^1(n) & \text{if } 1 \leqslant \gamma^0 = n \neq i, \\ \beta^0 & \text{if } 1 \leqslant \gamma^0 = i. \end{cases}$$

$(j \geqslant 1)$

$$\operatorname{Alt}(\alpha^{j+1}, \beta^j, i) = \lambda\gamma^j \begin{cases} \max\{i, \alpha^{j+1}(\langle u_0^{j-1}(0)\rangle)\} & \text{if } d_0^{j-1}\big((\gamma^j)_1\big) = 0, \\ \alpha^{j+1}(\gamma^j) & \text{if } 1 \leqslant d_0^{j-1}\big((\gamma^j)_1\big) = n \neq i, \\ \beta^j\big((\gamma^j)_2\big) & \text{if } 1 \leqslant d_0^{j-1}\big((\gamma^j)_1\big) = i. \quad \square \end{cases}$$

Clearly Alt is recursive and behaves as desired; e.g.,

$$\mathrm{Alt}\big(\langle \alpha_1^j, \alpha_2^j, \alpha_3^j\rangle, \beta^j, 2\big) = \langle \alpha_1^j, \beta^j, \alpha_3^j\rangle$$

and

$$\mathrm{Alt}\big(\langle \alpha_1^j, \alpha_2^j, \alpha_3^j\rangle, \beta^j, 5\big) = \langle \alpha_1^j, \alpha_2^j, \alpha_3^j, \lambda\gamma^{j-1}\cdot 0, \beta^j\rangle.$$

We now come to the Enumeration Theorem. We ask the reader to notice that, in defining the enumerating functional for functions φ of arguments having types $\leqslant r$ ($r > 0$) recursive in Θ where each of Θ has arguments only of types $\leqslant r-1$, S4. j is only used with $j \leqslant r-1$. (We will need this fact in the proof of Theorem 5.5.)

It is convenient to use a combination of the syntactic system of §1 and the semantic system of §2.

5.4 LEMMA. *Let $r > 0$ and $\Theta \equiv (\theta_1, \ldots, \theta_1)$, where each θ_t has only arguments of types $\leqslant r-1$. For each q and s, there is a recursive functional*

$$\psi\big(\Theta; z, e, \alpha^1, \ldots, \alpha^{r-1}, \alpha_1^{r-1}, \ldots, \alpha_q^{r-1}, \alpha_1^r, \ldots, \alpha_s^r\big)$$

such that, if

(i) *z is the index of a derivation $\varphi_1, \ldots, \varphi_p$ from Θ, and*

(ii) *e is the Gödel number of a 0-expression E based on $\varphi_1, \ldots, \varphi_p$ whose free variables of types $r-1$ and r are among $\gamma_1^{r-1}, \ldots, \gamma_q^{r-1}, \gamma_1^r, \ldots, \gamma_s^r$, and*

(iii) *Ω is the assignment obtained from $\alpha^1, \ldots, \alpha^{r-1}$, $\alpha_1^{r-1}, \ldots, \alpha_q^{r-1}$, $\alpha_1^r, \ldots, \alpha_s^r$ by assigning: α_i^{r-1} to the formal variable γ_i^{r-1}, α_i^r to γ_i^r, and, for $j < r-1$, $(\alpha^{j+1})_i$ to γ_i^j, then*

$$\psi\big([\Theta]_\Omega; z, e, \alpha^1, \ldots, \alpha^{r-1}, \alpha_1^{r-1}, \ldots, \alpha_q^{r-1}, \alpha_1^r, \ldots, \alpha_s^r\big) \simeq [E]_\Omega;$$

and $\psi([\Theta]_\Omega; z, e, \ldots) \uparrow$ if z and e are not as above.

PROOF. For simplicity, we suppress Θ and write

$$\mathfrak{A} \equiv \big(\alpha^1, \ldots, \alpha^{r-1}, \alpha_1^{r-1}, \ldots, \alpha_q^{r-1}, \alpha_1^r, \ldots, \alpha_s^r\big).$$

We define $\chi(\eta; z, e, \mathfrak{A})$ and show that $\psi = \chi^\infty$ has the required properties. The definition is by cases on the form of E (where $\ulcorner E \urcorner = e$).

Case 0. z and e are not as specified in (i) and (ii) above. Take $\chi(\eta; z, e, \mathfrak{A})$ to be undefined.

E1. E is γ_i^0. If $r = 1$ (so $1 \leqslant i \leqslant q$) take $\chi(\eta; z, e, \mathfrak{A}) \simeq \alpha_i^0$. Otherwise, take $\chi(\eta; z, e, \mathfrak{A}) \simeq (\alpha^1)_i$.

E2. E is $\varphi_i(\vec{A})$. Let E^* be as in Definition 1.3.2, E2. Take $\chi(\eta; z, e, \mathfrak{A}) \simeq \eta(z, \ulcorner E^* \urcorner, \mathfrak{A})$.

E4.λ. E is $\{\lambda\beta^j A\}(B)$. Take $\chi(\eta; z, e, \mathfrak{A}) \simeq \eta(z, \ulcorner S_B^{\beta^j} A \,|\urcorner, \mathfrak{A})$.

E4. j. E is $\gamma_i^j(B)$. We suppose $j > 1$ ($j = 1$ is almost identical).

If $j = r$ (and similarly if $j = r-1$) take

$$\chi(\eta; z, e, \mathfrak{A}) \simeq \alpha_i^r\Big(\lambda\delta^{r-2}\cdot\eta\Big(z, \ulcorner B\big(\gamma_k^{r-2}\big)\urcorner, \alpha^1, \ldots, \alpha^{r-2},$$
$$\mathrm{Alt}\big(\alpha^{r-1}, \delta^{r-2}, k\big), \alpha_1^{r-1}, \ldots, \alpha_q^{r-1}, \alpha_1^r, \ldots, \alpha_s^r\Big)\Big)$$

where γ_k^{r-2} is the first type-($r-2$) variable not free in B.

If $1 < j < r-1$, take

$$\chi(\eta; z, e, \mathfrak{A})$$

$$\simeq (\alpha^{j+1})_i\Big[\lambda\delta^{j-2}\cdot\eta\Big(z, \ulcorner B\big(\gamma_k^{j-2}\big)\urcorner, \alpha^1,\ldots,\alpha^{j-2}, \mathrm{Alt}(\alpha^{j-1}, \delta^{j-2}, k),$$
$$\alpha^j,\ldots,\alpha^{r-1}, \alpha_1^{r-1},\ldots,\alpha_q^{r-1}, \alpha_1^r,\ldots,\alpha_s^r\Big)\Big]$$

where γ_k^{j-2} is the first type-$(j-2)$ variable not free in B.

E5, E6. Obvious.

E7. Similarly to E4.j. (If θ_t has a type-$(r-1)$ argument, then θ_t is "essentially" a type-r object.)

This completes the definition of χ.

Now let $\bar{\psi}(z, e, \mathfrak{A})$ have all of the properties claimed for ψ except, possibly, recursiveness. It will suffice to show that, for all z, e, $\mathfrak{A}$,

$$\bar{\psi}(z, e, \mathfrak{A}) \simeq \chi^\infty(z, e, \mathfrak{A}).$$

($\supset$) It is easy to see that $\lambda ze\mathfrak{A}\cdot\chi(\bar{\psi}; z, e, \mathfrak{A}) \subset \bar{\psi}$, whence, by Lemma 2.1.1, $\chi^\infty \subset \bar{\psi}$.

($\subset$) We show, by induction on $\xi = |E|_\Omega$, that, if z and $e = \ulcorner E\urcorner$ are as specified in (i) and (ii) above and Ω is the asignment obtained from $\mathfrak{A}$, and $[E]_\Omega\downarrow$ then

$$\chi^\infty(z, e, \mathfrak{A}) \simeq [E]_\Omega(\simeq \bar{\psi}(z, e, \mathfrak{A})).$$

We consider only the case E4.j $(1 < j < r-1)$, the others being similar or trivial.

E4.j $(1 < j < r-1)$. E is $\gamma_i^j(B)$. Now $[\gamma_i^j]_\Omega = (\alpha^{j+1})_i$, so it suffices to show that

$$(*)\ \ [B]_\Omega^{<\xi} \subset \lambda\delta^{j-2}\cdot\chi^\infty\Big(z, \ulcorner B\big(\gamma_k^{j-2}\big)\urcorner, \alpha^1,\ldots,\alpha^{j-2}, \mathrm{Alt}(\alpha^{j-1}, \delta^{j-2}, k),$$
$$\alpha^j,\ldots,\alpha^{r-1}, \alpha_1^{r-1},\ldots,\alpha_q^{r-1}, \alpha_1^r,\ldots,\alpha_s^r\Big).$$

Now $[B]_\Omega^{<\xi} = \lambda\delta^{j-2}\cdot[B(\gamma_k^{j-2})]^{<\xi}_{\Omega[\gamma_k^{j-2}/\delta^{j-2}]}$ and $\Omega[\gamma_k^{j-2}/\delta^{j-2}]$ is the assignment obtained from $(\alpha^1,\ldots,\mathrm{Alt}(\alpha^{j-1}, \delta^{j-2}, k),\ldots,\alpha_s^r)$, so $(*)$ follows from the inductive hypothesis. □

5.5 Theorem (Enumeration Theorem). *For each* σ *and each* $\Theta \equiv (\theta_1,\ldots,\theta_1)$, *there is a recursive functional* $\psi(\Theta; z, \mathfrak{A})$ (*where* $\mathrm{ch}(\mathfrak{A}) = \sigma$) *such that, if* z *is the index of a functional* $\varphi(\Theta; \mathfrak{A})$, *then for all* Θ *and* $\mathfrak{A}$

$$\psi(\Theta; z, \mathfrak{A}) \simeq \varphi(\Theta; \mathfrak{A}) \simeq \{z\}^\Theta(\mathfrak{A}).$$

Moreover, if $\max tp(\mathfrak{A}) \leqslant r$ *and each argument of each of* Θ *has type* $\leqslant r-1$, *then* ψ *may be defined entirely in the theory of types* $\leqslant r$ *with* S4.j *used only with* $j \leqslant r-1$. (*In particular, no functional in the derivation of* ψ *will have type-*r *arguments other than those of* ψ, *and none of type* $> r$.)

Proof. We have only to code up the type-j arguments of ψ (for $j \leqslant r-2$) as type-$(j+1)$ objects and substitute them into the enumerating function of Lemma 5.4. This uses S4.k with $k \leqslant j+1 \leqslant r-1$. The construction in Lemma 5.4 also used S4.k only for $k \leqslant r-1$. □

As usual (see Kleene [**1959**, XIII, XIV]) we have:

5.6 Theorem ($s - m - n$ Theorem). *For each $m \geqslant 0$, there is a primitive recursive function $s^m(z, a_1, \ldots, a_m)$ such that, if z is the index of a recursive function $\{z\}(a_1, \ldots, a_m, \mathfrak{B})$ and $a_1, \ldots, a_m \in \mathbf{N}$, then for all $\mathfrak{B}$*

$$\{s^m(z, a_1, \ldots, a_m)\}(\mathfrak{B}) \simeq \{z\}(a_1, \ldots, a_m, \mathfrak{B}). \quad \square$$

5.7 Theorem (Second Recursion Theorem). *Let $\psi(x, \mathfrak{A})$ be recursive. Then there is an $e \in \mathbf{N}$ such that for all $\mathfrak{A}$*

$$\{e\}(\mathfrak{A}) \simeq \psi(e, \mathfrak{A}).$$

Beyond this point, it is not clear whether $\hat{Tp}$ possesses an interesting recursion theory. In particular, monotonicity requires that all recursive (total) predicates be constant. There may be some interesting results about the semirecursive predicates on $\hat{Tp}$, but the principal value of recursion over $\hat{Tp}$ is the setting that it provides for studying 59-recursion over Tp.

6. The relationship with some other theories. As we mentioned in §0, the system which we have developed here is very similar to that of Platek [**1966**]. Platek wished to develop a recursion theory on a (nearly) arbitrary collection Ob of "objects" (i.e., basic elements). His basic operation was that of taking the least fixed point of a recursive functional $\mathbf{F}(\eta; \vec{x})$. This required him to consider functionals whose function arguments might not be total. Perhaps, he suggested, a larger class of functions on Ob could be obtained if one also allowed these $\mathbf{F}$ to be defined as the least fixed points of type-3 functionals $\mathbf{\Phi}(\mathbf{F}, \eta; \vec{x})$. This required functionals $\mathbf{\Phi}$ whose type-2 arguments $\mathbf{F}$ were partial, and moreover, $\mathbf{F}$ might be defined on some partial type-1 arguments. Continuing in this manner, he arrived at his *hereditarily consistent* functionals HC. Platek's Reduction Theorem shows that there is no need to go up through the entire structure of HC—the same recursive functions of objects are obtained if one stops at level 2 (i.e., $\mathbf{F}(\eta; \vec{x})$) and performs recursions at that level.

Platek obtained an image of Kleene's 1959 theory by considering recursion on Ob $= \mathbf{N}$ and associating $Tp^{(n)}$ with a certain subset of the unary functions at the nth level of HC. Thus the partial functionals entered the picture as arguments with which to perform recursions—not as the basic objects of the theory.

Kechris and Moschovakis [**1977**] obtained Kleene's 1959 theory by taking Ob $= Tp$ and only going up to level 2 in HC (which the Reduction Theorem tells us is far enough). For this reason, "partial objects" do not occur in their system.

We arrived at $\hat{Tp}$ in an attempt to provide a semantics for Kleene's 1978 computations. Essentially, it became clear that recursive functions of higher type are not naturally limited to total arguments—even at the lowest level. This accounts for the presence of u in $\hat{Tp}^{(0)}$; u has no counterpart in HC.

Thus, while HC and $\hat{Tp}$ may appear to be very similar, the role of HC in Platek's theory is significantly different (at least conceptually) from that of $\hat{Tp}$ in our theory. Conceptually, our system is much closer to that of Kechris and Moschovakis [**1977**] than to that of Platek [**1966**].

Within all of these theories is that of Kleene **[1959]**. That this theory may be obtained by such varied means says something about the soundness of the original system.

References

A. S. Kechris and Y. N. Moschovakis

[1977] *Recursion in higher types*, Handbook of Mathematical Logic, K. J. Barwise, Editor, North-Holland, Amsterdam, pp. 681–737.

S. C. Kleene

[1952] *Introduction to metamathematics*, North-Holland, Amsterdam.

[1959] *Recursive functionals and quantifiers of finite types*. I, Trans. Amer. Math. Soc. **91**, 1–52.

[1963] *Recursive functionals and quantifiers of finite types*. II, Trans. Amer. Math. Soc. **108**, 106–142.

[1978] *Recursive functionals and quantifiers of finite types revisited*. I, Generalized Recursion Theory. II, J. E. Fenstad, R. O. Gandy and G. E. Sacks, Editors, North-Holland, Amsterdam, pp. 185–222.

[1980] *Recursive functionals and quantifiers of finite types revisited*. II, The Kleene Symposium, K. J. Barwise, H. J. Keisler and K. Kunen, Editors, North-Holland, Amsterdam, pp. 1–29.

[to appear] *Recursive functionals and quantifiers of finite types revisited*. III.

D. P. Kierstead

[1980] *A semantics for Kleene's j-expressions*, The Kleene Symposium, K. J. Barwise, H. J. Keisler and K. Kunen, Editors, North-Holland, Amsterdam, pp. 253–366.

R. A. Platek

[1966] *Foundations of recursion theory*, Ph.D. Thesis, Stanford University, Stanford, Calif.

Daniel Wagner Associates, Station Square One, Paoli, Pennsylvania 19301