Infinite Terms and Recursion in Higher Types

H. Schwichtenberg and S.S. Wainer

Systems of infinite terms defining functionals of finite type were first considered by Tait [10] and further developed by Feferman [3] initially in a proof-theoretic context. Later in unpublished notes Feferman introduced the system T_0 of infinite terms inductively generated from variables of all finite types and constants for the ordinary primitive recursive functions by application, abstraction and autonomous enumeration: if for each n, f(n) codes a term $t_n \in T_0$ and f is itself defined by a term of T_0 then the term $\langle t_n \rangle_{n \in N}$ is in T_0. This definition can be relativized to an arbitrary functional $\mathfrak{F}$ and the resulting system of terms is denoted by $T_0(\mathfrak{F})$. Feferman proved that if $\mathfrak{F}$ is of type 2 then the functions definable in $T_0(\mathfrak{F})$ are precisely the functions recursive in $\mathfrak{F}$ (This also follows from our results here together with [11]). This immediately poses the problem of whether infinite terms can be used to characterize full Kleene recursion in higher types and more specifically whether, for $\mathfrak{F}$ of type n+2, $T_0(\mathfrak{F})$ gives a characterization of the n+1 - section of $\mathfrak{F}$.

We show in §2 that for arbitrary $\mathfrak{F}$ of type n+2 the functionals of types $\leqslant$ n+1 definable in $T_0(\mathfrak{F})$ are just those functionals appearing in a naturally-constructed Kleene-type hierarchy based on $\mathfrak{F}$, which generalizes [11]. (This hierarchy expands primitive recursively though not necessarily recursively since $\mathfrak{F}$ may not be a "jump"). The proof of this equivalence uses normalization for $T_0(\mathfrak{F})$.

As a consequence we obtain a negative answer to the second problem above as follows. The type n+1 functionals definable in $T_0({}^{n+2}E)$ are precisely the functionals obtained in Kleene's hierarchy H_a^{n+1}, $a \in \mathcal{O}^{n+1}$ [5]. But Moschovakis [7] has shown that the hierarchy H_a^2, $a \in \mathcal{O}^2$ does not exhaust the 2-section of ${}^3E$.

In connection with the first problem mentioned above Feferman [4] has recently obtained a new definition of full recursion in higher types which, although not formulated as a system of terms, is nevertheless motivated by the idea of autonomous enumeration. In §3 we investigate ways of generalizing the autonomous sequencing scheme, so as to obtain complete characterizations of higher-type recursion (The obvious idea is first to allow "long" sequences, enumerated by definable functionals of arbitrary pure type, rather than just functions as in T_0. But this is insufficient as it stands, and needs to be modified further.) This leads to a hierarchy of systems of terms T_0, T_1, T_2 and Long Partial Terms, the last one of which turns out to be nothing other than a reformulation of Feferman's definition [4].

§1. The System $T_0(\mathfrak{F})$ of Infinite Terms.

Type symbols are 0 and with σ, τ also $(\sigma \to \tau)$. As usual we write $\sigma_1, \ldots, \sigma_n \to \tau$ for $(\sigma_1 \to (\sigma_2 \to \ldots (\sigma_n \to \tau)..))$. Finite sequences of type symbols are denoted by $\underset{\sim}{\sigma}, \underset{\sim}{\tau}$ etc. and we let $\ulcorner \underset{\sim}{\tau} \urcorner$ be a (canonically defined) code number of $\underset{\sim}{\tau}$. Let M_τ be the class of all (set-theoretic) functionals of type τ, i.e. $M_0 = N$, the natural numbers, and $M_{\sigma \to \tau} = M_\tau^{M_\sigma}$, the set of all mappings from M_σ into M_τ. Elements of $\bigcup_\tau M_\tau$ are denoted by $\mathfrak{F}$, F, G, H, α, β and finite sequences of them $\underset{\sim}{F}, \underset{\sim}{G}, \underset{\sim}{\alpha}$ etc.

We fix a functional $\mathfrak{F}$ of arbitrary type τ. The terms of $T_0(\mathfrak{F})$ will be built up from variables x_0^σ, x_1^σ, x_2^σ, ... for each type σ, the symbol $\mathfrak{J}$, and for each $k \geq 0$ a constant p_k for the k-th primitive recursive function, by means of application, abstraction and autonomous formation of sequences as described in the introduction. Each term will have only finitely-many free variables.

We define inductively (i) a set $C^{\mathfrak{J}} \subseteq N$ of codes, (ii) the term t_a denoted by the code $a \in C^{\mathfrak{J}}$, (iii) a function Typ such that for each $a \in C^{\mathfrak{J}}$, Typ (a) determines the type of t_a and furthermore a sequence of variables containing all variables free in t_a, (iv) for each $a \in C^{\mathfrak{J}}$ the value $[a]^{\underset{\sim}{F}}$ (in $\bigcup_\tau M_\tau$) of t_a under a type-preserving assignment of $\underset{\sim}{F} = F_1, F_2, \ldots, F_n$ to the sequence of variables determined by Typ (a).

For each $a \in C^{\mathfrak{J}}$ Typ (a) will have the form $\ulcorner \underset{\sim}{\tau}, \sigma \urcorner$ where σ is the type of t_a and $\underset{\sim}{\tau} = \tau_1, \ldots, \tau_n$ is to be thought of as determining the sequence $\underset{\sim}{x} = x_1, \ldots, x_n$ of free variables in t_a (i.e. x_i is to be the variable $x_j^{\tau_i}$ if τ_i is the j-th occurrence of that type symbol in $\underset{\sim}{\tau}$). With this $\underset{\sim}{x}$ we also write $t_a(\underset{\sim}{x})$ for t_a. From the definition it will be clear that $\underset{\sim}{x}$ contains all of the free and none of the bound variables of t_a.

I (Variables) $a = \langle 1, i, \ulcorner \underset{\sim}{\tau} \urcorner \rangle \in C^{\mathfrak{J}}$ if $1 \leq i \leq n$ and $\underset{\sim}{\tau} = \tau_1, \ldots, \tau_n$. Typ (a) $= \ulcorner \underset{\sim}{\tau}, \tau_i \urcorner$, $t_a = t_a(\underset{\sim}{x}) = x_i$ and $[a]^{\underset{\sim}{F}} = F_i$.

II (Application) Let $a_1, a_2 \in C^{\mathfrak{J}}$ where Typ $(a_1) = \ulcorner \underset{\sim}{\tau}, \sigma \to \rho \urcorner$ and Typ$(a_2) = \ulcorner \underset{\sim}{\tau}, \sigma \urcorner$. Then $a = \langle 2, a_1, a_2 \rangle \in C^{\mathfrak{J}}$, Typ (a) $= \ulcorner \underset{\sim}{\tau}, \rho \urcorner$, $t_a = \left(t_{a_1} \ t_{a_2} \right)$ and $[a]^{\underset{\sim}{F}} = [a_1]^{\underset{\sim}{F}} [a_2]^{\underset{\sim}{F}}$.

III (<u>Abstraction</u>) Let $a_1 \in C^{\mathfrak{F}}$ and $\mathrm{Typ}(a_1) = \ulcorner \underline{\tau}, \sigma, \rho \urcorner$.

Then $a = \langle 3, a_1 \rangle \in C^{\mathfrak{F}}$, $t_a(\underline{x}) = \lambda y \,.\, t_{a_1}(\underline{x}, y)$, $\mathrm{Typ}\ (a) = \ulcorner \underline{\tau}, \sigma \to \rho \urcorner$ and $[a]^{\underline{F}} G = [a_1]^{\underline{F}, G}$ for all $G \in M_\sigma$.

IV (<u>Autonomous Sequences</u>) Let $a_1 \in C^{\mathfrak{F}}$, $\mathrm{Typ}(a_1) = \ulcorner 0, 0 \urcorner$ and for all n, $[a_1]^n = b_n \in C^{\mathfrak{F}}$ and $\mathrm{Typ}\ (b_n) = \ulcorner \underline{\tau}, 0 \urcorner$. Then $a = \langle 4, \ulcorner \underline{\tau} \urcorner, a_1 \rangle \in C^{\mathfrak{F}}$, $t_a = \langle t_{b_n} \rangle_{n \in \mathbb{N}}$, $\mathrm{Typ}\ (a) = \ulcorner \underline{\tau}, 0 \to 0 \urcorner$ and $[a]^{\underline{F}} n = [b_n]^{\underline{F}}$ for all $n \in \mathbb{N}$.

V (<u>Primitive Recursion</u>) Let $a_1, \ldots, a_n \in C^{\mathfrak{F}}$ where $n \geqslant 0$ is the number of arguments of the k-th. primitive recursive function p_k and for $1 \leqslant i \leqslant n$ $\mathrm{Typ}\ (a_i) = \ulcorner \underline{\tau}, 0 \urcorner$. Then $a = \langle 5, k, \ulcorner \underline{\tau} \urcorner, a_1, \ldots, a_n \rangle \in C^{\mathfrak{F}}$, $t_a = p_k(t_{a_1}, \ldots, t_{a_n})$, $\mathrm{Typ}\ (a) = \ulcorner \underline{\tau}, 0 \urcorner$ and $[a]^{\underline{F}} = p_k([a_1]^{\underline{F}}, \ldots, [a_n]^{\underline{F}})$

VI (<u>The Constant $\mathfrak{F}$</u>) $a = \langle 6, \ulcorner \underline{\tau} \urcorner \rangle \in C^{\mathfrak{F}}$, $t_a =$ the symbol $\mathfrak{F}$, $\mathrm{Typ}(a) = \ulcorner \underline{\tau}, \tau \urcorner$ and $[a]^{\underline{F}} = \mathfrak{F}$.

Obviously Typ can be chosen as a primitive recursive function.

$T_0(\mathfrak{F})$ is the set of all terms t_a, $a \in C^{\mathfrak{F}}$. We want to normalize the terms of $T_0(\mathfrak{F})$, that is eliminate all subterms of the form $(\lambda x t)s$. For a system of nonconstructive infinite terms (in the sense that no restriction is imposed on the formation of infinite sequences) this was done by Tait [10], extending earlier work of Lorenzen, Novikov and Schütte concerning infinite proofs. We assume here some knowledge of Tait's paper. Now it is more or less standard how such operations on nonconstructive infinite terms can be paralleled by operations on their constructive counterparts such as terms in $T_0(\mathfrak{F})$ or, more precisely, codes in $C^{\mathfrak{F}}$ (see e.g. Feferman [4], Lopez-Escobar [6], Schwichtenberg [9]) . Hence we do not give proofs but merely state the proper lemmata, following mainly Feferman [4].

Most of them (Lemmas 1-4) are proved using the primitive recursion theorem.

The type level $L\tau$ of a type symbol τ is defined by $L0 = 0$, $L(\sigma \to \tau) = \max (L\sigma + 1, L\tau)$. The rank of a code $a \in C^{\mathfrak{F}}$ is defined as the supremum of the type levels of all subcodes of the form $a_1 = \langle 3, \ldots \rangle$ occurring in a context $\langle 2, a_1, a_2 \rangle$. More precisely we inductively define Ra for $a \in C^{\mathfrak{F}}$ as follows:

$R\langle 1, i, \ulcorner\underset{\sim}{\tau}\urcorner\rangle = 0$

$R\langle 2, a_1 a_2\rangle = \max (Ra_1, Ra_2, La_1)$ if a_1 has the form $\langle 3, \ldots\rangle$.

$\qquad = \max (Ra_1, Ra_2)$ otherwise.

$R\langle 3, a_1\rangle = Ra_1$

$R\langle 4, \ulcorner\underset{\sim}{\tau}\urcorner, a_1\rangle = \max (Ra_1, \sup_n R[a_1]^n)$

$R\langle 5, k, \ulcorner\underset{\sim}{\tau}\urcorner, a_1, \ldots, a_n\rangle = \max (Ra_1, \ldots, Ra_n)$

$R\langle 6, \ulcorner\underset{\sim}{\tau}\urcorner\rangle = 0$.

Here $La_1 = L\sigma$ where $\mathrm{Typ}(a_1) = \ulcorner\underset{\sim}{\tau}, \sigma\urcorner$. Clearly we have $Ra \leqslant \omega$.

A code $a \in C^{\mathfrak{F}}$ (and the corresponding term t_a) is called <u>irreducible</u> or <u>normal</u> if and only if $Ra = 0$.

<u>Lemma 1</u> (Extension)

There is a primitive recursive function Ext such that for all $a \in C^{\mathfrak{F}}$ and all types σ the following holds.

Let $\mathrm{Typ}(a) = \ulcorner\underset{\sim}{\tau}, \rho\urcorner$. Then $\mathrm{Ext}(a, \ulcorner\sigma\urcorner) \in C^{\mathfrak{F}}$, $\mathrm{Typ}(\mathrm{Ext}(a, \ulcorner\sigma\urcorner)) = \ulcorner\sigma, \underset{\sim}{\tau}, \rho\urcorner$, $R\,\mathrm{Ext}(a, \ulcorner\sigma\urcorner) = Ra$ and for all $G, \underset{\sim}{F}$ of the appropriate types, $[a]^{\underset{\sim}{F}} = [\mathrm{Ext}(a, \ulcorner\sigma\urcorner)]^{G, \underset{\sim}{F}}$.

<u>Lemma 2</u> (Interchange)

There are primitive recursive functions p_{ij} such that for all $a \in C^{\mathfrak{F}}$ the following holds. Let $\mathrm{Typ}(a) = \ulcorner\underset{\sim}{\tau}, \sigma\urcorner$. Then $p_{ij}(a) \in C^{\mathfrak{F}}$, $\mathrm{Typ}(p_{ij}(a)) = \ulcorner\pi_{ij}(\underset{\sim}{\tau}), \sigma\urcorner$, $R\,p_{ij}(a) = Ra$ and for all $\underset{\sim}{F}$ of the appropriate types, $[a]^{\underset{\sim}{F}} = [p_{ij}(a)]^{\pi_{ij}(\underset{\sim}{F})}$, where π_{ij} interchanges the i-th. and j-th. components in the respective n-tuple.

Lemma 3 (Substitution)

There is a primitive recursive function Sub such that for all $a,b \in C^{\mathfrak{I}}$ with $\mathrm{Typ}(a) = \ulcorner\sigma, \underset{\sim}{\tau}, \rho\urcorner$ and $\mathrm{Typ}(b) = \ulcorner\underset{\sim}{\tau}, \sigma\urcorner$ the following holds. $\mathrm{Sub}(a,b) \in C^{\mathfrak{I}}$, $\mathrm{Typ}(\mathrm{Sub}(a,b)) = \ulcorner\underset{\sim}{\tau}, \rho\urcorner$, $R\,\mathrm{Sub}(a,b) \leqslant \max(Ra, Rb, Lb)$ and for all $\underset{\sim}{F}$ of the appropriate types, $[a]^{[b]^{\underset{\sim}{F}}, \underset{\sim}{F}} = [\mathrm{Sub}(a,b)]^{\underset{\sim}{F}}$.

Lemma 4 (Reduction)

There is a primitive recursive function Red such that for all m and all $a \in C^{\mathfrak{I}}$ with $Ra \leqslant m+1$ the following holds. $\mathrm{Red}(a,m) = a' \in C^{\mathfrak{I}}$, $\mathrm{Typ}(a') = \mathrm{Typ}(a)$, $Ra' \leqslant m$ and for all $\underset{\sim}{F}$ of the appropriate types, $[a']^{\underset{\sim}{F}} = [a]^{\underset{\sim}{F}}$.

Normalization Theorem 1

There is a primitive recursive function N such that for all $a \in C^{\mathfrak{I}}$ the following holds. $N(a) = a^* \in C^{\mathfrak{I}}$, $\mathrm{Typ}(a^*) = \mathrm{Typ}(a)$, a^* is in normal form, i.e. $Ra^* = 0$, and for all $\underset{\sim}{F}$ of the appropriate types, $[a^*]^{\underset{\sim}{F}} = [a]^{\underset{\sim}{F}}$.

Each term t_a in $T_0(\mathfrak{I})$ defines a functional, namely $\lambda\underset{\sim}{F}.[a]^{\underset{\sim}{F}}$, whose arguments correspond to the free variables occurring in the term. We wish to give a recursion-theoretic characterization of the functionals definable in $T_0(\mathfrak{I})$, and since arbitrary finite types can be canonically coded into pure types it will henceforth be more convenient for us to restrict attention to those functionals h whose arguments $\underset{\sim}{\alpha} = \alpha_1, \dots, \alpha_m$ are of pure types $\leqslant n$ and whose values are of type 0. $\mathfrak{I}$ is now assumed to be an arbitrary but fixed type n+2 object.

If $h(\alpha_1,\dots,\alpha_m)$ is definable in $T_o(\mathfrak{J})$ then it is defined by a normal term of type 0. Such a term can only be either a variable of type 0 or a term of the form $p_k(s_1,\dots,s_r)$ where $s_1,\dots s_r$ are normal terms of type 0, or else a term of the form st where s and t are normal. In this latter case s cannot be of the form $((s_o s_1)\dots)s_k$ with $k \geqslant 1$ since s_o would then have to be a variable of impure type, so s must be either $\mathfrak{J}$ or a variable of pure type $\geqslant 1$ or a term of the form $\langle t_{a_o}, t_{a_1}, t_{a_2},\dots\rangle$ where $\lambda x.a_x$ is defined by a normal term. Hence t must be either of type 0 or else of the form $\lambda y.t'$ where t' is of type 0 (If t were of the form $\langle t_{b_o}, t_{b_1}, t_{b_2},\dots\rangle$ then we could replace it by $\lambda y.\langle t_{b_o}, t_{b_1}, t_{b_2},\dots\rangle y$). Thus it is clear that each of the functionals $h(\alpha_1,\dots,\alpha_m)$ definable in $T_o(\mathfrak{J})$ can be generated by means of the schemes 1,...,7 below. The converse, that the functionals generated by schemes 1,...,7 are all definable in $T_o(\mathfrak{J})$, should be clear and can easily be proved by a simple application of the primitive recursion theorem. Each scheme defines a functional h_e where the index e codes up (in the usual way) all relevant details of the particular scheme being applied. We now let $\underset{\sim}{x} = x_1,\dots,x_k$ denote variables of type 0, $\underset{\sim}{\alpha} = \alpha_1,\dots,\alpha_m$ variables of pure types $\leqslant$ n and β a variable of the appropriate pure type $\leqslant$ n.

1. $h_e(\underset{\sim}{x},\underset{\sim}{\alpha}) = p_k(\underset{\sim}{x})$
2. $h_e(\underset{\sim}{\alpha}) \quad = \alpha_i(h_{e_1}(\underset{\sim}{\alpha}))$ where type $\alpha_i = 1$.
3. $h_e(\underset{\sim}{\alpha}) \quad = \alpha_j(\lambda\beta.h_{e_1}(\underset{\sim}{\alpha},\beta))$ where type of $\alpha_j > 1$.
4. $h_e(\underset{\sim}{\alpha}) \quad = \mathfrak{J}(\lambda\beta.h_{e_1}(\underset{\sim}{\alpha},\beta))$
5. $h_e(x,\underset{\sim}{\alpha}) = h_{h_{e_1}(x)}(\underset{\sim}{\alpha})$ provided that for each x, $h_{e_1}(x)$ is an index for a functional with arguments $\underset{\sim}{\alpha}$.

6. $h_e(\underset{\sim}{\alpha}) = h_{e_1}(h_{e_2}(\underset{\sim}{\alpha}), \underset{\sim}{\alpha})$

7. $h_e(\underset{\sim}{\alpha}) = h_{e_1}(\underset{\sim}{\alpha}')$ where $\underset{\sim}{\alpha}'$ is some permutation of $\underset{\sim}{\alpha}$.

To be precise, the above schemes should be interpreted as a simultaneous inductive definition of a set of indices e, and for each index e a functional h_e . We believe however that the intention is clear.

§2. The $\mathfrak{J}$-hierarchy.

We now develop a recursion-theoretic hierarchy based on a fixed but completely arbitrary type n+2 object $\mathfrak{J}$, and prove that the functionals of type $\leqslant$ n+1 appearing in the hierarchy are precisely those functionals definable in $T_0(\mathfrak{J})$. The hierarchy is just a generalization of [11] to higher types.

Let $[\![e]\!]^F(\underset{\sim}{\alpha})$, $e < \omega$, be a standard enumeration of all functionals (with arguments $\underset{\sim}{\alpha}$ of type $\leqslant$ n) primitive recursive in a type n+1 object F (in the sense of Kleene [5]). We assume $[\![e]\!]^F(\underset{\sim}{\alpha}) = 0$ if e is not an index for a functional of the appropriate string of variables.

We associate with $\mathfrak{J}$ an operator $\mathfrak{j}$ defined as follows

$$\mathfrak{j}(F)\ (\langle x,\alpha\rangle) = \left\langle [\![x]\!]^F(\alpha,0^n), \mathfrak{J}(\lambda\beta.[\![x]\!]^F(\alpha,\beta))\right\rangle .$$

The $\mathfrak{J}$-hierarchy is then obtained by iterating $\mathfrak{j}$ over a simultaneously generated set of ordinal notations. Note however that the word "hierarchy" is used in a rather broad sense here, since $\mathfrak{j}$ may not be a jump operator in the usual sense (and although $\mathfrak{j}$ raises "primitive recursive degree" it need not raise "degree"). As a result of this our hierarchies will not in general have the uniqueness property.

Definition.

$\mathcal{O}^{\mathfrak{J}}$, $<_o^{\mathfrak{J}}$, $|\ |^{\mathfrak{J}}$ and $F_a^{\mathfrak{J}}$ for $a \in \mathcal{O}^{\mathfrak{J}}$ are inductively defined as follows, where α,β are variables of type n. (Since $\mathfrak{J}$ is fixed we will usually drop the superscript $\mathfrak{J}$) :

(i) $1 \in \mathcal{O}$, $\neg(b <_o 1)$, $|1| = 0$ and $F_1(\alpha) = 0$.

(ii) If $a \in \mathcal{O}$ then $2^a \in \mathcal{O}$, $b <_o 2^a \leftrightarrow (b <_o a \vee b = a)$,
$|2^a| = |a| + 1$ and $F_{2^a}(\langle x,\alpha\rangle) = \langle [\![x]\!]^{F_a}(\alpha, 0^n), \mathfrak{F}(\lambda\beta . [\![x]\!]^{F_a}(\alpha,\beta))\rangle$
where 0^n here denotes the zero type n object.

(iii) If $a \in \mathcal{O}$ and $\phi = [\![e]\!]^{F_a}$ is a function such that $\phi(0) = a$, $\phi(m) \in \mathcal{O}$
and $\phi(m) <_o \phi(m+1)$ for all m, then
$3^a 5^e \in \mathcal{O}$, $b <_o 3^a 5^e \leftrightarrow (\exists m)(b <_o \phi(m))$, $|3^a 5^e| = \sup_m |\phi(m)|$ and
$F_{3^a 5^e}(\langle x,\alpha\rangle) = F_{\phi(x)}(\alpha)$.

Clearly if $a <_o b$ then F_a is of lower primitive recursive degree than F_b, and every F_b is recursive in $\mathfrak{F}$.

Examples

(1) If $\mathfrak{F}$ is of type 2 then the above hierarchy exhausts the 1-section of $\mathfrak{F}$ (see [11]).

(2) If $\mathfrak{F}$ is the functional ${}^{n+2}E$ which introduces quantification over type n then the above definition gives an alternative version of Kleene's proposed hierarchy of hyper-order n+1 predicates [5]. Our definition differs from Kleene's particularly in the formation of limit levels, where we insist that fundamental sequences ϕ be primitive recursive (rather than just recursive) in previous levels. However standard methods show that the two definitions give rise to the same class of predicates and functionals (and coincide at limit stages). Moschovakis [7] has shown that, for n = 1, the hierarchy does not exhaust the 2-section of ${}^3E$ (nor the 1-section of ${}^3E$).

(3) If $\mathfrak{F}$ is the superjump functional we obtain an alternative version of Platek's hierarchy [8] but again, Aczel and Hinman [1] have shown that this does not exhaust the 1-section of the superjump.

Limit Property.

There are primitive recursive functions M and N such that if for each $m, \lambda\underset{\sim}{\alpha}.G(m,\underset{\sim}{\alpha}) = [\![\psi(m)]\!]^{F_{\phi(m)}}$ where $\psi = [\![i]\!]^{F_a}$, $\phi = [\![e]\!]^{F_a}$ and $a = \phi(0) \leqslant_o \phi(m) <_o \phi(m+1)$, then $3^a5^{M(e)} \in \mathcal{O}$, $\phi(m) <_o 3^a5^{M(e)}$ for each m, and $G = [\![N(i)]\!]^{F_{3^a5^{M(e)}}}$.

Proof

Choose M so that $[\![M(e)]\!]^{F_a}(0) = a$, $[\![M(e)]\!]^{F_a}(m+1) = 2^{\phi(m)}$.

Let $\langle\underset{\sim}{\alpha}\rangle^n$ denote a standard primitive recursive coding of a sequence $\underset{\sim}{\alpha}$ as a single type n object, and let S_o be a primitive recursive function such that $[\![S_o(j)]\!]^F(\langle\underset{\sim}{\alpha}\rangle^n, 0^n) = [\![j]\!]^F(\underset{\sim}{\alpha})$ for any type n+1 object F. Then $G(m,\underset{\sim}{\alpha}) = [\![\psi(m)]\!]^{F_{\phi(m)}}(\underset{\sim}{\alpha}) =$ $[\![S_o(\psi(m))]\!]^{F_{\phi(m)}}(\langle\underset{\sim}{\alpha}\rangle^n, 0^n) = (F_{2^{\phi(m)}}(\langle S_o(\psi(m)), \langle\underset{\sim}{\alpha}\rangle^n\rangle))_o =$ $(F_{3^a5^{M(e)}}(\langle m+1, \langle S_o(\psi(m)), \langle\underset{\sim}{\alpha}\rangle^n\rangle\rangle))_o$. Now let m^n denote the type n object with constant value m and let S_1 be a primitive recursive function such that $[\![S_1(j)]\!]^F(m^n, 0^n) = [\![j]\!]^F(m)$ for any type n+1 object F. Then $\psi(m) = [\![i]\!]^{F_a}(m) = [\![S_1(i)]\!]^{F_a}(m^n, 0^n) = (F_{2^a}(\langle S_1(i), m^n\rangle))_o$ $= (F_{3^a5^{M(e)}}(\langle 1, \langle S_1(i), m^n\rangle\rangle))_o$. We therefore have

$G(m,\alpha) = (F_{3^a5^{M(e)}}(\langle m+1, \langle S_o(F_{3^a5^{M(e)}}(\langle 1, \langle S_1(i), m^n\rangle\rangle))_o, \langle\underset{\sim}{\alpha}\rangle^n\rangle\rangle))_o$

and it remains to choose N so that N(i) is an index of this expression as a function of m and α, primitive recursive in $F_{3^a5^{M(e)}}$.

Lemma 5.

There are primitive recursive functions I and C such that if e is an index of a functional h_e defined by schemes 1,...,7 then for any $b \in \mathcal{O}$, $C(e,b) \in \mathcal{O}$, $b <_o C(e,b)$ and $h_e = [\![I(e,b)]\!]^{F_{C(e,b)}}$.

Proof

First note that the arbitrary $b \in \mathcal{O}$ appears because in order to deal with scheme 5 we need to locate $\lambda\underset{\sim}{\alpha}.h_e(x+1,\underset{\sim}{\alpha})$ above

$\lambda\underline{\alpha}.h_e(x,\underline{\alpha})$ in $\mathcal{O}$ so that the Limit Property can then be used to piece together the whole functional $\lambda x,\underline{\alpha}.h_e(x,\underline{\alpha})$. Also in dealing with scheme 6 we will need to locate h_{e_1} above h_{e_2}. These complications arise because there is no corresponding Uniqueness Property for an arbitrary $\mathfrak{F}$-hierarchy, since Uniqueness requires quantification and we do not in general have ${}^2E$ recursive in $\mathfrak{F}$.

I and C will be defined simultaneously by the primitive recursion theorem, with induction on the definition of h_e by schemes 1,...,7.

Suppose h_e is defined by 1, so from e we can find k so that $h_e(\underline{x},\underline{\alpha}) = p_k(\underline{x})$. Clearly there is a primitive recursive function f_1 such that for any F of type n+1, $[\![f_1(k)]\!]^F(\underline{x},\underline{\alpha}) = p_k(\underline{x})$. Thus we only need to put $I(e,b) = f_1(k)$ and $C(e,b) = 2^b$ in this case.

Suppose h_e is defined from h_{e_1} by 2,3, or 7. By induction hypothesis we can assume $b <_o C(e_1,b)$ and $h_{e_1} = [\![I(e_1,b)]\!]^{F_{C(e_1,b)}}$. But 2,3,7 correspond to Kleene's schemes S7,S8,S6 respectively and hence we can put $C(e,b) = C(e_1,b)$ and in each case compute $I(e,b)$ as a primitive recursive function of e and $I(e_1,b)$.

If $h_e(\underline{\alpha}) = \mathfrak{F}(\lambda\beta.h_{e_1}(\underline{\alpha},\beta))$ by scheme 4 then again by hypothesis we can assume $b <_o C(e_1,b)$ and $h_{e_1} = [\![I(e_1,b)]\!]^{F_{C(e_1,b)}}$. Then there is a primitive recursive function f_2 such that $h_{e_1}(\underline{\alpha},\beta) = [\![f_2(e,I(e_1,b))]\!]^{F_{C(e_1,b)}}(\langle\underline{\alpha}\rangle^n,\beta)$. Therefore $h_e(\underline{\alpha}) = \mathfrak{F}(\lambda\beta.[\![f_2(e,I(e_1,b))]\!]^{F_{C(e_1,b)}}(\langle\underline{\alpha}\rangle^n,\beta)) = (F_{2^{C(e_1,b)}}(\langle f_2(e,I(e_1 b)),\langle\underline{\alpha}\rangle^n\rangle))_1$.

Now put $C(e,b) = 2^{C(e_1,b)}$ and h_e is clearly primitive recursive in $F_{C(e,b)}$ with index $I(e,b)$ primitive recursively computable from e and $I(e_1,b)$.

Next suppose h_e is defined by scheme 5. Then $h_e(x,\underline{\alpha}) = h_{h_{e_1}(x)}(\underline{\alpha})$ where, by the induction hypothesis,

$b <_o C(e_1,b)$ and $h_{e_1} = [\![I(e_1,b)]\!]^{F_{C(e_1,b)}}$, and for each x and all $d \in \mathcal{O}$, $d <_o C(h_{e_1}(x),d)$ and $\lambda\underline{\alpha}.h_e(x,\underline{\alpha})$ is primitive recursive in $F_{C(h_{e_1}(x),\,d)}$ with index $I(h_{e_1}(x),d)$. Define $\phi(0) = C(e_1,b)$ and $\phi(m+1) = C(h_{e_1}(m),\phi(m))$, and define $\psi(0) = 0$, $\psi(m+1) = I(h_{e_1}(m),\phi(m))$. Then for each $m, \lambda\underline{\alpha}.h_e(m,\underline{\alpha}) = [\![\psi(m+1)]\!]^{F_{\phi(m+1)}}$ where ϕ and ψ are primitive recursive in $F_{C(e_1,b)}$ with indices z and i primitive recursively computable from $C(e_1,b)$, $I(e_1,b)$ and primitive recursive indices of C and I. Also $\phi(0) = C(e_1,b) \leqslant_o \phi(m) <_o \phi(m+1)$ for every m, by hypothesis. Therefore by the Limit Property, h_e is primitive recursive in $F_{C(e,b)}$ with index $I(e,b)$ where $C(e,b) = 3^{C(e_1,b)}5^{M(z)}$ and $I(e,b)$ is given by a simple primitive recursive function of $N(i)$.

Finally suppose $h_e(\underline{\alpha}) = h_{e_1}(h_{e_2}(\underline{\alpha}),\underline{\alpha})$ by scheme 6. By induction hypothesis we can assume $b <_o C(e_2,b)$, $h_{e_2} = [\![I(e_2,b)]\!]^{F_{C(e_2,b)}}$ and for all $d \in \mathcal{O}$, $d <_o C(e_1,d)$ and $h_{e_1} = [\![I(e_1,d)]\!]^{F_{C(e_1,d)}}$. Define $\phi(0) = C(e_2,b)$, $\phi(1) = C(e_1,\phi(0))$ and $\phi(m+2) = 2^{\phi(m+1)}$. Then ϕ is primitive recursive (and hence primitive recursive in $F_{C(e_2,b)}$ with an index u primitive recursively computable from e,b, and a primitive recursive index of C. Also $\phi(m) <_0 \phi(m+1)$ by hypothesis and so $3^{C(e_2,b)}5^u \in \mathcal{O}$. Put $C(e,b) = 3^{C(e_2,b)}5^u$. Then $b <_0 C(e_2,b) <_0 C(e,b)$ and since $F_{C(e_2,b)} = \lambda\alpha . F_{C(e,b)}(\langle 0,\alpha\rangle)$ it follows that h_{e_2} is primitive recursive in $F_{C(e,b)}$ with an index primitive recursively computable from $I(e_2,b)$. Now for some fixed primitive recursive function f_3 we have

$$\begin{aligned} h_{e_1}(x,\underset{\sim}{\alpha}) &= [\![I(e_1,C(e_2,b))]\!]^{F_{\phi(1)}}(x,\underset{\sim}{\alpha}) \\ &= [\![f_3(I(e_1,C(e_2,b)))]\!]^{F_{\phi(1)}}(\langle x,\underset{\sim}{\alpha}\rangle^n, 0^n) \\ &= (F_{\phi(2)}(\langle f_3(I(e_1,C(e_2,b))),\langle x,\underset{\sim}{\alpha}\rangle^n\rangle))_0 \\ &= (F_{C(e,b)}(\langle 2,\langle f_3(I(e_1,C(e_2,b))), \langle x,\underset{\sim}{\alpha}\rangle^n\rangle\rangle))_0 \end{aligned}$$

Thus h_{e_1} is also primitive recursive in $F_{C(e,b)}$ with an index primitive recursively computable from e,b and primitive recursive indices of I and C. Hence h_e is primitive recursive in $F_{C(e,b)}$ by Kleene's scheme S4, with index $I(e,b)$ given as a primitive recursive function of $I(e_2,b)$, e, b, and primitive recursive indices of I and C.

We give I and C the value 0 if none of the above cases applies.

Inspection of the above cases shows that $C(e,b)$ and $I(e,b)$ are defined simultaneously from $C(e_1,b)$ $C(e_2,b)$, $I(e_1,b)$ $I(e_2,b)$, e,b and primitive recursive indices of C and I. Since $e_1,e_2 < e$ the simultaneous definition is a primitive recursion on e. Therefore by the simultaneous primitive recursion theorem (e.g. Lemma 2.1 of [2]) we can indeed find primitive recursive indices of C and I which satisfy this definition. This completes the proof.

Next we show that every functional $G(\underset{\sim}{\alpha})$, with arguments $\underset{\sim}{\alpha}$ of pure types $\leqslant n$ and with values of type 0, which appears in the $\mathfrak{J}$-hierarchy, is definable by a term of $T_0(\mathfrak{J})$.

Lemma 6

There are primitive recursive functions p and p_1 such that if the type $n+1$ functional F is defined by a term t_c of $T_0(\mathfrak{J})$ then $[\![e]\!]^F$ is defined by the term $t_{p(c,e)}$ of $T_0(\mathfrak{J})$ and $\lambda x,\underset{\sim}{\alpha}\,.\,[\![x]\!]^F(\underset{\sim}{\alpha})$ is defined by the term $t_{p_1(c)}$ of $T_0(\mathfrak{J})$.

Proof

We first define p by the primitive recursion theorem with cases corresponding to the schemes So,...,S8 by which $[\![e]\!]^F$ is defined. In this proof and the next, u,v will be used to denote variables of $T_0(\mathfrak{J})$ of the appropriate types.

If $[\![e]\!]^F$ is defined by S1, S2, S3 then $[\![e]\!]^F$ is just a primitive recursive function of its numerical arguments and so $p(c.e)$ is given explicitly as a function of e.

If $[\![e]\!]^F = \lambda\underset{\sim}{\alpha}.[\![e_1]\!]^F([\![e_2]\!]^F(\underset{\sim}{\alpha}),\underset{\sim}{\alpha})$ through S4 then we can assume inductively that $t_{p(c,e_1)}$ defines $[\![e_1]\!]^F$ and $t_{p(c,e_2)}$ defines $[\![e_2]\!]^F$. Therefore $[\![e]\!]^F$ is defined by the term $\lambda\underset{\sim}{u}\,.\,t_{p(c,e_1)}(t_{p(c,e_2)}\underset{\sim}{u})\underset{\sim}{u}$ and we can clearly compute $p(c,e)$ as a primitive recursive function of $p(c,e_1)$, $p(c,e_2)$ and e.

If $[\![e]\!]^F$ is defined by S5 then $[\![e]\!]^F(0,\underset{\sim}{\alpha}) = [\![e_1]\!]^F(\underset{\sim}{\alpha})$ and $[\![e]\!]^F(x+1,\underset{\sim}{\alpha}) = [\![e_2]\!]^F([\![e]\!]^F(x,\underset{\sim}{\alpha}),x,\underset{\sim}{\alpha})$ where again we can assume inductively that $t_{p(c,e_1)}$ defines $[\![e_1]\!]^F$ and $t_{p(c,e_2)}$ defines $[\![e_2]\!]^F$. Now let $r(0) = p(c,e_1)$ and $r(x+1) =$ the code for the term $\lambda\underset{\sim}{u}\,.\,t_{p(c,e_2)}(t_{r(x)}\underset{\sim}{u})\,x\underset{\sim}{u}$. Then for each x, $t_{r(x)}$

defines $\lambda\underset{\sim}{\alpha}.[\![e]\!]^F(x,\underset{\sim}{\alpha})$ and therefore $<t_{r(x)}>_{x\in\mathbb{N}}$ defines $[\![e]\!]^F$. But r is primitive recursive, with index i primitive recursively computable from $p(c,e_1)$ $p(c,e_2)$ and e. Hence we can primitive recursively compute from i, first a code for the term defining r, and then the code p(c,e) for the term $<t_{r(x)}>_{x\in\mathbb{N}}$ which defines $[\![e]\!]^F$.

The cases where $[\![e]\!]^F$ is defined by S6 and S7, corresponding to permutation of arguments and function application, are trivial.

If $[\![e]\!]^F(\underset{\sim}{\alpha})=\alpha_i(\lambda\beta.[\![e_1]\!]^F(\underset{\sim}{\alpha},\beta))$ through S8 then it is easy to define p(c,e) primitive recursively from e and $p(c,e_1)$ such that $t_{p(c,e)}=\lambda\underset{\sim}{u}.\ u_i(\lambda v.t_{p(c,e_1)}\underset{\sim}{u}v)$. The case S0 is treated similarly, replacing α_i by F and u_i by t_c.

It is clear from the above cases that p is primitive recursive, as required.

To define p_1 simply note that $\lambda x\underset{\sim}{\alpha}.\ [\![x]\!]^F(\underset{\sim}{\alpha})$ can now be defined by the term $<t_{p(c,x)}>_{x\in\mathbb{N}}$, whose code is given as a primitive recursive function of c.

Lemma 7

There is a primitive recursive function q such that if $a\in O^{\mathfrak{J}}$ then $q(a)\in C^{\mathfrak{J}}$ and $t_{q(a)}$ defines $F_a^{\mathfrak{J}}$.

Proof

Again by the primitive recursion theorem. Define q(1) so that $t_{q(1)}=\lambda u.0$. Now assume $t_{q(a)}$ defines F_a. Since $x=<x,\alpha>_0(0)$ and $\alpha=<x,\alpha>_1$ there are terms t_k and t_ℓ

which define the decoding functions $\lambda\alpha.\alpha_0(0)$ and $\lambda\alpha.\alpha_1$.
But $F_{2^a} = \lambda\alpha . <[\![\alpha_0(0)]\!]^{F_a}(\alpha_1, 0^n), \mathfrak{F}(\lambda\beta.[\![\alpha_0(0)]\!]^{F_a}(\alpha_1,\beta))>$ and so F_{2^a} is defined by the term $\lambda u . <t_{p_1(q(a))}(t_k u)(t_l u)0^n, \mathfrak{F}(\lambda v . t_{p_1(q(a))}(t_k u)(t_l u)v)>$ whose code $q(2^a)$ is clearly given as a primitive recursive function of $q(a)$. If $3^a5^e \in \mathcal{O}$ then $F_{3^a5^e} = \lambda\alpha.\ F_{[\![e]\!]^{F_a}(\alpha_0(0))}(\alpha_1)$, so if $\phi = [\![e]\!]^{F_a}$ we can assume inductively that $F_{\phi(x)}$ is defined by $t_{q(\phi(x))}$ for each x and therefore $F_{3^a5^e}$ is defined by the term $\lambda u. <t_{q(\phi(x))}>_{x\in\mathbb{N}}(t_k u)(t_l u)$.
Now ϕ is defined by the term $t_{p(q(a),e)}$ and so $\lambda x . q(\phi(x))$ is defined by a term whose code is primitive recursively computable from $q(a)$, e and a primitive recursive index of q . Thus we can compute $q(3^a5^e)$ primitive recursively from $q(a)$, e, and a primitive recursive index of q, so that $t_{q(3^a5^e)}$ is the term $\lambda u.\ <t_{q(\phi(x))}>_{x\in\mathbb{N}}(t_k u)(t_l u)$ which defines $F_{3^a5^e}$. The primitive recursion theorem then provides an index of q satisfying the above definition, and this completes the proof.

Putting the above results together we have

<u>Theorem</u> 2

A functional with arguments of pure types $\leqslant$ n and values of type 0 is definable in $T_0(\mathfrak{F})$ if and only if it is primitive recursive in $F_a^{\mathfrak{F}}$ for some $a \in \mathcal{O}^{\mathfrak{F}}$.

<u>Corollary</u>

If $\mathfrak{F}$ is of type $\leqslant 2$ then the functions definable in $T_0(\mathfrak{F})$ are precisely the functions recursive in $\mathfrak{F}$.

But for $\mathfrak{F}$ of type $\geqslant 3$ the functions definable in $T_0(\mathfrak{F})$ do not, in general, exhaust the 1-section of $\mathfrak{F}$.

§3. Extensions of $T_0(\mathfrak{F})$

The reason why $T_0(\mathfrak{F})$ for $\mathfrak{F}$ of type level $\geqslant 3$ does not give full Kleene recursion in $\mathfrak{F}$ seems to be that sequences used to build up terms in $T_0(\mathfrak{F})$ are indexed by natural numbers and so each term can be regarded as a countable well-founded tree, whereas Kleene-computations in types $\geqslant 3$ are in general uncountable. Thus it is tempting to allow sequences indexed by higher-type objects and to consider a system $T_1(\mathfrak{F})$ of infinite terms which is defined just as $T_0(\mathfrak{F})$ in §1 except that clause IV is now generalized to read as follows

IV* (Long autonomous sequences) Assume $a_1 \in C^{\mathfrak{F}}$, $\mathrm{Typ}(a_1) = \ulcorner \underset{\sim}{\tau}, 0 \urcorner$ and for all $\underset{\sim}{F} \in M_{\underset{\sim}{\tau}}$, $[a_1]^{\underset{\sim}{F}} = b_{\underset{\sim}{F}} \in C^{\mathfrak{F}}$ and $\mathrm{Typ}(b_{\underset{\sim}{F}}) = \ulcorner \underset{\sim}{\tau}, 0 \urcorner$. Then $a = \langle 4, a_1 \rangle \in C^{\mathfrak{F}}$, $t_a = \langle t_{b_{\underset{\sim}{F}}} \rangle_{\underset{\sim}{F} \in M_{\underset{\sim}{\tau}}}$, $\mathrm{Typ}(a) = \ulcorner \underset{\sim}{\tau}, \underset{\sim}{\tau} \to 0 \urcorner$ and for all $\underset{\sim}{F}, \underset{\sim}{G}$ of the appropriate types, $[a]^{\underset{\sim}{G}}\underset{\sim}{F} = [b_{\underset{\sim}{F}}]^{\underset{\sim}{G}}$.

But if $t_a = \langle t_{b_{\underset{\sim}{F}}} \rangle_{\underset{\sim}{F} \in M_{\underset{\sim}{\tau}}}$ is a term formed by IV* then as $\underset{\sim}{F}$ ranges over $M_{\underset{\sim}{\tau}}$ there can still only be countably-many different values of $b_{\underset{\sim}{F}}$. Thus the "depths" of the trees corresponding to terms in $T_1(\mathfrak{F})$ remain countable, so we cannot expect $T_1(\mathfrak{F})$ to be adequate to define all functions recursive in $\mathfrak{F}$. In fact for the case $\mathfrak{F} = {}^3E$ we have :

Theorem 3

The functionals of type $\leqslant 2$ definable in $T_1({}^3E)$ are just those definable in $T_0({}^3E)$.

Proof

For $i = 0,1$ we let $C_i^{\mathfrak{F}}$ be the set of codes for terms in $T_i(\mathfrak{F})$, and for each $a \in C_i^{\mathfrak{F}}$ we denote the corresponding functional by $\lambda \underset{\sim}{F}. [a]_i^{\underset{\sim}{F}}$. We show that there is a primitive recursive function p such that if $a \in C_1^{{}^3E}$ is normal and $\mathrm{Typ}(a) = \ulcorner \underset{\sim}{\tau}, \sigma \urcorner$ where $\underset{\sim}{\tau}$ is a sequence of types 0 or 1, then $p(a) \in C_0^{{}^3E}$ and for all

$\underset{\sim}{\alpha} \in M_{\underset{\sim}{\mathcal{I}}}$, $[a]_1^{\underset{\sim}{\alpha}} = [p(a)]_0^{\underset{\sim}{\alpha}}$. The only non-trivial case is when $a = \langle 4, a_1 \rangle$, $\mathrm{Typ}(a) = \ulcorner \underset{\sim}{\mathcal{I}}, \underset{\sim}{\mathcal{I}} \to 0 \urcorner$. Then for all $\underset{\sim}{\alpha}, \underset{\sim}{\beta} \in M_{\underset{\sim}{\mathcal{I}}}$, $[a]_1^{\underset{\sim}{\alpha}} \underset{\sim}{\beta} = [\,[a_1]_1^{\underset{\sim}{\beta}}]_1^{\underset{\sim}{\alpha}}$. Proceeding by induction on $a \in C_1^{{}^3E}$ we can then assume that $[a]_1^{\underset{\sim}{\alpha}} \underset{\sim}{\beta} = [p[p(a_1)]_0^{\underset{\sim}{\beta}}]_0^{\underset{\sim}{\alpha}}$. Now using the function-quantifier ${}^3E$ we can primitive recursively compute, from $p(a_1)$ and a primitive recursive index of p , codes b and c such that $\lambda n . [b]_0^n$ enumerates all the values of $\lambda \underset{\sim}{\beta}\ p([p(a_1)]_0^{\underset{\sim}{\beta}})$ and $[c]_0^{\underset{\sim}{\beta}} = \mu n\ ([b]_0^n = p([p(a_1)]_0^{\underset{\sim}{\beta}}))$. Then for all α, β ,

$[a]_1^{\underset{\sim}{\alpha}} \underset{\sim}{\beta} = [\langle 4, b \rangle]_0^{\underset{\sim}{\alpha}} [c]_0^{\underset{\sim}{\beta}}$, and so from b and c we can primitive recursively compute $p(a)$ such that $[p(a)]_0^{\underset{\sim}{\alpha}} = \lambda \underset{\sim}{\beta} [\langle 4, b \rangle]_0^{\underset{\sim}{\alpha}} [c]_0^{\underset{\sim}{\beta}} = [a]_1^{\underset{\sim}{\alpha}}$. We finally obtain the required p by the primitive recursion theorem.

Clearly this Theorem will hold for any $\mathfrak{Z}$ such that ${}^3E$ is definable in $T_0(\mathfrak{Z})$, and it will also generalize to higher types when relativized to ${}^4E, {}^5E$ etc.

The depth of a term $t_{\langle 4, a_1 \rangle} = \langle t_{b_{\underset{\sim}{F}}} \rangle_{M_{\underset{\sim}{\mathcal{I}}}}$ formed by IV* is given in the obvious way by $\mathrm{depth}(t_{\langle 4, a_1 \rangle}) = \sup_{\underset{\sim}{F}}(\mathrm{depth}(t_{a_1}) + 1, \mathrm{depth}(t_{b_{\underset{\sim}{F}}}) + 1)$ and since each $b_{\underset{\sim}{F}} = [a_1]^{\underset{\sim}{E}} \in C^{\mathfrak{Z}}$ we are here only taking the supremum of countably - many (countable) ordinals. Now a natural way to get terms with uncountable depth is to allow the $\underset{\sim}{F}$'s to be used as <u>constants</u> in $t_{b_{\underset{\sim}{F}}}$, i.e. to let $b_{\underset{\sim}{F}} \in C^{\mathfrak{Z}, \underset{\sim}{F}}$. Thus, following a suggestion of Feferman, we further extend our systems of terms to give new systems $T_2(\underset{\sim}{F})$ as follows.

This time we inductively define, simultaneously for all $\underset{\sim}{F}$ of the appropriate types, a set $C^{\underset{\sim}{F}}$ of codes and for each $a \in C^{\underset{\sim}{F}}$ a term $t_a \in T_2(\underset{\sim}{F})$ and a total functional $\lambda \underset{\sim}{G} . [a;\underset{\sim}{F}]^{\underset{\sim}{G}}$ defined by that term. We write $[a;\underset{\sim}{F}]^{\underset{\sim}{G}}$ in order to make explicit the relativization to the fixed $\underset{\sim}{F} = F_1, \ldots, F_n$. The clauses in the definition are I, II, III, V and VI as before (but with VI introducing each of the constants $F_1, \ldots, F_n$) together with

IV** (Long relativized autonomous sequences) Assume $a_1 \in C^{\underset{\sim}{F}}$, $\underset{\sim}{F}$ of type $\underset{\sim}{\tau}$, Type $(a_1) = \ulcorner \underset{\sim}{\sigma}, 0 \urcorner$ and for all $\underset{\sim}{G} \in M_{\underset{\sim}{\sigma}}$, $[a_1; \underset{\sim}{F}]^{\underset{\sim}{G}} = b_{\underset{\sim}{G}} \in C^{\underset{\sim}{F}, \underset{\sim}{G}}$ and $\mathrm{Typ}(b_{\underset{\sim}{G}}) = \ulcorner \underset{\sim}{\rho}, 0 \urcorner$. Then $a = \langle 4, a_1 \rangle \in C^{\underset{\sim}{F}}$, $t_a = \langle t_{b_{\underset{\sim}{G}}} \rangle_{\underset{\sim}{G} \in M_{\underset{\sim}{\sigma}}}$, $\mathrm{Typ}(a) = \ulcorner \underset{\sim}{\rho}, \underset{\sim}{\sigma} \to 0 \urcorner$ and for all $\underset{\sim}{G} \in M_{\underset{\sim}{\sigma}}$, $\underset{\sim}{H} \in M_{\underset{\sim}{\rho}}$

$$[a; \underset{\sim}{F}]^{\underset{\sim}{H}} \underset{\sim}{G} = [[a_1; \underset{\sim}{F}]^{\underset{\sim}{G}} ; \underset{\sim}{F}, \underset{\sim}{G}]^{\underset{\sim}{H}} \quad .$$

With $\underset{\sim}{F}$ the empty sequence we thus obtain C and T_2, so if we denote the depth of a term t_a in $T_2(\underset{\sim}{G})$ by $|a|^{\underset{\sim}{G}}$ then the depth $|a|$ of a term $t_a = \langle t_{b_{\underset{\sim}{G}}} \rangle_{\underset{\sim}{G} \in M_{\underset{\sim}{\sigma}}}$ in T_2 is given by

$$|a| = \sup_{\underset{\sim}{G}} (|a_1| + 1 \ , \ |b_{\underset{\sim}{G}}|^{\underset{\sim}{G}} + 1)$$

where $|b_{\underset{\sim}{G}}|^{\underset{\sim}{G}}$ may now, of course, have uncountably many different values, and so $|a|$ will in general be uncountable (cf. definitions 1,2 in Moschovakis [7]).

We shall show (Theorems 4 and 5) that for arbitrary a with $\mathrm{Typ}(a) = \ulcorner 0 \urcorner$ the partial functionals $\lambda \underset{\sim}{\alpha}.[a; \underset{\sim}{\alpha}]$ are just the Kleene partial recursive functionals $\lambda \underset{\sim}{\alpha}.\{e\}(\underset{\sim}{\alpha})$. It then follows by the lemma below, that the total functionals $[a; \mathfrak{Y}]$ with $a \in C^{\mathfrak{Y}}$, exhaust the functionals recursive in $\mathfrak{Y}$.

Lemma

For each $\underset{\sim}{\tau}, \underset{\sim}{\sigma}$ there is a primitive recursive function f such that (with $\underset{\sim}{F}, \underset{\sim}{G}$ ranging over $M_{\underset{\sim}{\tau}}, M_{\underset{\sim}{\sigma}}$, respectively)

(i) $\forall \underset{\sim}{G} (a \in C^{\underset{\sim}{F}, \underset{\sim}{G}}) \leftrightarrow f(a) \in C^{\underset{\sim}{F}}$

(ii) $\forall G (a \in C^{\underset{\sim}{F}, \underset{\sim}{G}}) \to [a; \underset{\sim}{F}, \underset{\sim}{G}]^{\underset{\sim}{H}} = [f(a); \underset{\sim}{F}]^{\underset{\sim}{G}, \underset{\sim}{H}} \ \& \ |a|^{\underset{\sim}{F}, \underset{\sim}{G}} < |f(a)|^{\underset{\sim}{F}}$.

Proof

Given $\underset{\sim}{\tau}, \underset{\sim}{\sigma}$, we can easily find a primitive recursive function q such that for all $\underset{\sim}{F} \in M_{\underset{\sim}{\tau}}$, $\underset{\sim}{G} \in M_{\underset{\sim}{\sigma}}$, $[q(a); \underset{\sim}{F}]^{\underset{\sim}{G}} = a$. Hence

$$\begin{aligned} [a; \underset{\sim}{F}, \underset{\sim}{G}]^{\underset{\sim}{H}} &\simeq [[q(a); \underset{\sim}{F}]^{\underset{\sim}{G}}; \underset{\sim}{F}, \underset{\sim}{G}]^{\underset{\sim}{H}} \\ &\simeq [\langle 4, q(a) \rangle ; \underset{\sim}{F}]^{\underset{\sim}{H}} \underset{\sim}{G} && \text{by IV}^{**} \\ &\simeq [f(a); \underset{\sim}{F}]^{\underset{\sim}{G}, \underset{\sim}{H}} \end{aligned}$$

with f(a) depending primitive recursively on q(a). The proof of the lemma is now obvious.

Theorem 4

There is a primitive recursive function g such that

$$\text{(i)}\ \{e\}(\underset{\sim}{\alpha})\downarrow \leftrightarrow g(e)\in C^{\underset{\sim}{\alpha}}$$

$$\text{(ii)}\ \{e\}(\underset{\sim}{\alpha})\downarrow \rightarrow [g(e);\underset{\sim}{\alpha}] = \{e\}(\underset{\sim}{\alpha})$$

Proof

We shall define g from its own primitive recursive index using the primitive recursion theorem in the usual manner. The definition is by cases depending on the form of e.

The implication for left to right in (i) together with (ii) are proved by induction on $\{e\}(\underset{\sim}{\alpha})\simeq w$. The proof of the implication from right to left in (i) is by induction on $|g(e)|^{\underset{\sim}{\alpha}}$ and will be clear after the definition is completed.

We restrict ourselves to the cases S4, S8 and S9, the other cases being obvious or similar.

Case S4; $\{e\}(\underset{\sim}{\alpha}) \simeq \{e_1\}(\{e_2\}(\underset{\sim}{\alpha}),\underset{\sim}{\alpha})$.
First note that as in §1 we can easily obtain a primitive recursive function Sub such that $b\in C^{\underset{\sim}{F}}$ implies

$$\text{(i)}\ [a;[b;\underset{\sim}{F}]^{\underset{\sim}{G}};\underset{\sim}{F}]^{\underset{\sim}{G}} \simeq [\mathrm{Sub}(a,b);\underset{\sim}{F}]^{\underset{\sim}{G}}$$

$$\text{(ii)}\ |a|^{[b;\underset{\sim}{F}]^{\underset{\sim}{G}},\underset{\sim}{F}} < |\mathrm{Sub}(a,b)|^{\underset{\sim}{F}} \text{ and } |b|^{\underset{\sim}{F}} < |\mathrm{Sub}(a,b)|^{\underset{\sim}{F}}.$$

(However, note that if $\mathrm{Sub}_0(a,b)$ is the function corresponding as usual to term-substitution we have to put $\mathrm{Sub}(a,b) = \langle 2,\langle 2, c_0, \mathrm{Sub}_0(a,b)\rangle, b\rangle$ with $c_0\in C^{\underset{\sim}{F}}$ such that $[c_0;\underset{\sim}{F}]^{\underset{\sim}{G}}H_1H_2 = H_1$).
We now obtain

$$\{e\}(\underset{\sim}{\alpha}) = [g(e_1);[g(e_2);\underset{\sim}{\alpha}],\underset{\sim}{\alpha}] \quad \text{by ind.hyp.}$$

$$= [\mathrm{Sub}(g(e_1),g(e_2));\underset{\sim}{\alpha}].$$

Hence it suffices to put $g(e) = \mathrm{Sub}(g(e_1),g(e_2))$.

Case S8: $\{e\}(\underset{\sim}{\alpha}) \simeq \alpha_j(\lambda\beta\{e_1\}(\underset{\sim}{\alpha},\beta))$. By ind. hyp. we have $g(e_1) \in C^{\underset{\sim}{\alpha},\beta}$ and $\{e_1\}(\underset{\sim}{\alpha},\beta) = [g(e_1);\underset{\sim}{\alpha},\beta]$ for all β, and hence by the lemma, $\{e_1\}(\underset{\sim}{\alpha},\beta) = [a_1;\underset{\sim}{\alpha}]^\beta$ with a_1 primitive recursively computable from $g(e_1)$. It is now easy to obtain a_2, a_3 also primitive recursively from a_1 such that

$$\begin{aligned} \lambda\beta\{e_1\}(\underset{\sim}{\alpha};\beta) &= [a_2;\underset{\sim}{\alpha}] \\ \alpha_j(\lambda\beta\{e_1\}(\underset{\sim}{\alpha},\beta)) &= [a_3;\underset{\sim}{\alpha}] \end{aligned}$$

It remains to set $g(e) = a_3$.

Case S9: $\{e\}(x,\underset{\sim}{\alpha}) \simeq \{x\}(\underset{\sim}{\alpha})$. By ind. hyp. we can assume that $\{x\}(\underset{\sim}{\alpha}) = [g(x);\underset{\sim}{\alpha}]$. Now from a primitive recursive index of g we can easily compute a code $a_1 \in C^{x,\underset{\sim}{\alpha}}$ such that $[a_1;x,\underset{\sim}{\alpha}] = g(x)$ and then a code $a_2 \in C^{x,\underset{\sim}{\alpha}}$ such that $[[a_2;x,\underset{\sim}{\alpha}]; x,\underset{\sim}{\alpha}] = [[a_1;x,\underset{\sim}{\alpha}] ; \underset{\sim}{\alpha}] = \{x\}(\underset{\sim}{\alpha})$. But then an application of IV^{**} yields $\langle 4,a_2\rangle \in C^{x,\underset{\sim}{\alpha}}$ such that $[\langle 4,a_2\rangle; x,\underset{\sim}{\alpha}] = [[a_2;x,\underset{\sim}{\alpha}] ; x,\underset{\sim}{\alpha}] = \{x\}(\underset{\sim}{\alpha})$ and it then remains simply to put $g(e) = \langle 4,a_2\rangle$.

<u>Theorem 5</u>.

There is a primitive recursive function h such that

(i) $a \in C^{\underset{\sim}{\alpha}} \leftrightarrow \{h(a)\}(\underset{\sim}{\alpha})\downarrow$

(ii) $a \in C^{\underset{\sim}{\alpha}} \rightarrow \{h(a)\}(\underset{\sim}{\alpha}) = [a;\underset{\sim}{\alpha}]$

It is fairly straightforward to define such an h using the primitive recursion theorem; we omit the details.

Since the treatment of $T_2(\underset{\sim}{F})$ involved a discussion of partial functionals anyway, it seems natural to look for a more direct method of introducing partial recursion in the context of infinite terms. One way of doing this is to return first to the system $T_1(\mathfrak{F})$ and then relax the conditions under which the autonomous sequencing scheme IV^* may be applied, by not requiring any longer that the enumerating functional given by a_1 has only previously defined codes as values. The functionals so defined will now in general be partial. But not only $[a]^{\underset{\sim}{G}}$ as a function of $\underset{\sim}{G}$ will be partial (as we would like) but also the values $[a]^{\underset{\sim}{G}}$ for certain fixed $\underset{\sim}{G}$ may be partial functionals and as such will not even be objects of our underlying domain $\cup_{\underset{\sim}{\tau}} M_{\underset{\sim}{\tau}}$. To avoid this difficulty we instead let t_a be the term $\langle t_{b_{\underset{\sim}{F}}} \rangle_{\underset{\sim}{F} \in M_{\underset{\sim}{\tau}}} \underset{\sim}{x}$, with $\underset{\sim}{x}$ a sequence of variables of type $\underset{\sim}{\tau}$, so that the values of $[a]^{\underset{\sim}{F}}$, when defined, are natural numbers (i.e. total objects of type 0). This leads to a system of infinite "partial" terms $t_a, a \in C$ defined by I, II, III, V and IV*** below (We no longer relativize to $\mathfrak{J}$ since it is not really necessary here. One can easily show, for this new system, that there is a primitive recursive function $\lambda a.a'$ such that if $a \in C^{\mathfrak{J}}$ then $a' \in C$ and for all $\underset{\sim}{F}, [a']^{\mathfrak{J},\underset{\sim}{F}} = [a]^{\underset{\sim}{F}}$) .

IV*** (<u>Long partial autonomous sequences</u>) Assume $a_1 \in C$ and $\mathrm{Typ}(a_1) = \ulcorner \underset{\sim}{\tau}, 0 \urcorner$. Then $a = \langle 4, a_1 \rangle \in C$ and $t_a = \langle t_{b_{\underset{\sim}{F}}} \rangle_{\underset{\sim}{F} \in M_{\underset{\sim}{\tau}}} \underset{\sim}{x}$ where $b_{\underset{\sim}{F}} = [a_1]^{\underset{\sim}{F}}$ and $t_{b_{\underset{\sim}{F}}}$ is undefined if $b_{\underset{\sim}{F}} \notin C$. Furthermore $\mathrm{Typ}(a) = \ulcorner \underset{\sim}{\tau}, 0 \urcorner$ and $[a]^{\underset{\sim}{F}}$ is defined with value m if and only if (i) $[a_1]^{\underset{\sim}{F}}$ is defined, (ii) $[a_1]^{\underset{\sim}{F}} = b_{\underset{\sim}{F}} \in C$ with $\mathrm{Typ}(b_{\underset{\sim}{F}}) = \ulcorner \underset{\sim}{\tau}, 0 \urcorner$, and (iii) $[b_{\underset{\sim}{F}}]^{\underset{\sim}{F}}$ is defined with value m.

Now in what sense do I, II, III, IV***, V constitute a definition of the concepts $a \in C, t_a$ and $[a]^{\underset{\sim}{F}}$? The formerly critical point in the inductive definition of C was the use of quantification over M_τ in IV (with $\tau = 0$) and IV*, IV** (with τ arbitrary), which meant that C was "at least" a complete Π^1_1 set. But this clause has now been removed to give IV*** and so the new C can be defined independently of t_a and $[a]^{\underset{\sim}{F}}$, and is simply primitive recursive (as is the set of indices for partial recursive functionals). Incidentally the primitive recursive function Typ also needs to be redefined so that $\mathrm{Typ}(\langle 4, a_1 \rangle) = \mathrm{Typ}(a_1)$. We next consider $[a]^{\underset{\sim}{F}}$. Since $[a]^{\underset{\sim}{F}}$ may now be undefined we need to give a definition of the relation $[a]^{\underset{\sim}{F}} \simeq G$, to be read "$[a]^{\underset{\sim}{F}}$ is defined with value G". This relation is clearly analogous to Kleene's $\{e\}^{\underset{\sim}{F}}(\underset{\sim}{x}) \simeq z$ and is given by the following induction :

(1) Variables. $[a]^{\underset{\sim}{F}} \simeq F_i$ if $a = \langle 1, i, \ulcorner\underset{\sim}{\tau}\urcorner\rangle$, $\underset{\sim}{F} = F_1, \ldots F_n \in M_{\underset{\sim}{\tau}}$ and $1 \leqslant i \leqslant n$.

(2) Application. If $[a_1]^{\underset{\sim}{F}} \simeq G_1$ and $[a_2]^{\underset{\sim}{F}} \simeq G_2$ where $G_1 \in M_{\sigma \to \rho}$ and $G_2 \in M_\sigma$ then $[a]^{\underset{\sim}{F}} \simeq G_1 G_2$ where $a = \langle 2, a_1, a_2 \rangle$.

(3) Abstraction. If $[a_1]^{\underset{\sim}{F}, G} \simeq HG$ for all $G \in M_\sigma$ then $[a]^{\underset{\sim}{F}} \simeq H$ where $a = \langle 3, a_1 \rangle$.

(4) Long partial autonomous sequences. If $[a_1]^{\underset{\sim}{F}} \simeq b$ and $[b]^{\underset{\sim}{F}} \simeq m$ then $[a]^{\underset{\sim}{F}} \simeq m$ where $a = \langle 4, a_1 \rangle$.

(5) Primitive Recursion. If $[a_i]^{\underset{\sim}{F}} \simeq m_i$ for $1 \leqslant i \leqslant n_k$ then $[a]^{\underset{\sim}{F}} \simeq p_k(m_1, \ldots, m_{n_k})$ where $a = \langle 5, k, \ulcorner\tau\urcorner, a_1, \ldots, a_{n_k}\rangle$, $\underset{\sim}{F} \in M_{\underset{\sim}{\tau}}$ and p_k is the k - the. primitive recursive function.

For the "partial" terms t_a for $a \in C$ we omit corresponding details. Notice however, the problems which can arise when $a = \langle 4, a_1 \rangle$ and $t_a = \langle t_{b_{\underset{\sim}{F}}} \rangle_{\underset{\sim}{F} \in M_{\underset{\sim}{\tau}}} \underset{\sim}{x}$ where $b_{\underset{\sim}{F}} \simeq [a_1]^{\underset{\sim}{F}}$. Since a_1 is quite arbitrary we do not know anything about the values $b_{\underset{\sim}{F}}$; in particular we may have $b_{\underset{\sim}{F}} = a$ for some $\underset{\sim}{F}$ and so in general t_a may have the structure of a non-well-founded tree (analogous to the undefined computations which can arise through Kleene's scheme S9). One can think of a computation of $[a]^{\underset{\sim}{F}}$ from given $a, \underset{\sim}{F}$ as working through t_a starting from the outermost node. In such a computation, an infinite branching occurs in the case of abstraction (where the structure of t_a has only a 1-fold branching), but only a 2-fold branching occurs in the case of sequencing (whereas the structure of t_a in this case has an infinite branching).

We have arrived at an inductive definition (1)...(5) in which terms are not explicitly mentioned. This definition is due to Feferman, and is the starting point of [4]. One can show either directly (as is done in [4]) or by reduction to Theorems 4 and 5, that the partial functionals $\lambda\underset{\sim}{\alpha}.[a]^{\underset{\sim}{\alpha}}$ exhaust the Kleene partial recursive functionals .

REFERENCES.

[1] P.Aczel and P.G. Hinman, "Recursion in the Superjump", in Generalized Recursion Theory (Eds. Fenstad and Hinman), North-Holland (1974).

[2] S. Feferman, "Classifications of Recursive Functions by means of Hierarchies", Trans. Amer. Math. Soc. vol 104 (1962) pp. 101-122.

[3] S. Feferman, "Ordinals and Functionals in Proof Theory", Proc. of Int. Congress of Mathematicians Nice (1970), pp. 229-233.

[4] S. Feferman, "Recursion in Total Functionals of Finite Type", to appear.

[5] S.C. Kleene, "Recursive Functionals and Quantifiers of Finite Types I, II", Trans. Amer. Math. Soc. vol 91 (1959) pp. 1-52, vol 108 (1963) pp. 106-142.

[6] E.G.K. Lopez-Escobar, "Remarks on an Infinitary Language with Constructive Formulas", Journ. Symb. Logic vol 32 (1967) pp. 305-319.

[7] Y.N. Moschovakis, "Hyperanalytic Predicates", Trans. Amer Math. Soc. vol 129 (1967) pp. 249-282.

[8] R.A. Platek, "A Countable Hierarchy for the Superjump", in Logic Colloquium '69 (Eds. Gandy and Yates) North-Holland (1971).

[9] H. Schwichtenberg, "Elimination of Higher Type Levels in Definitions of Primitive Recursive Fnls.by Transfinite Recursion", to appear in Proc. of Bristol Logic Colloquium 1973 (Eds. Rose and Shepherdson), North-Holland.

[10] W.W. Tait, "Infinitely Long Terms of Transfinite Type", in Formal Systems and Recursive Functions (Eds. Crossley and Dummett) North-Holland (1965).

[11] S.S. Wainer, "A Hierarchy for the 1-Section of Any Type Two Object", Journ. Symb. Logic vol 39 (1974) pp. 88-94.