SOLOMON FEFERMAN

Recursion in total functionals of finite type

Compositio Mathematica, tome 35, nº 1 (1977), p. 3-22
<http://www.numdam.org/item?id=CM_1977__35_1_3_0>

COMPOSITIO MATHEMATICA, Vol. 35, Fasc. 1, 1977, pag. 3–22
Noordhoff International Publishing
Printed in the Netherlands

RECURSION IN TOTAL FUNCTIONALS OF FINITE TYPE

Solomon Feferman*

1. Introduction

Let $\langle M_\tau \rangle_\tau$ be the maximal finite type structure over any domains of individuals $M^{(0)}, M^{(1)}, \ldots, M^{(p)}$ with $M^{(0)} = N$. We use here *all* types τ obtained from ground types $0^{(0)}, 0^{(1)}, \ldots, 0^{(p)}$ by iteration of $(\sigma \to \tau)$. F, G, $H, \ldots$ range over $\cup_\tau M_\tau$ and $\mathfrak{F}$ is used for sequences $\langle F_1, \ldots, F_n \rangle$. A relation

(1) $$G \leq \mathfrak{F}$$

is inductively generated by some simple closure conditions I–V given in §2. It is shown (in §4) that for the case $p = 0$ this is equivalent to:

(2) $$G \text{ is Kleene recursive in } \mathfrak{F},$$

i.e. as defined in [6].

The basic generating conditions are those for *explicit definition* (I–III), *primitive recursion* (V) and a rule called *secondary reference* (IV). This last underlies *effective enumeration*, for which the informal idea is that if $G_0, \ldots, G_m, \ldots$ is a sequence of objects (of the same type τ) presented recursively in $\mathfrak{F}$ then $\lambda x \cdot G_x$ is also recursive in $\mathfrak{F}$. The hypothesis is here taken to mean that we have a function D recursive in $\mathfrak{F}$ which enumerates a sequence $D(0), \ldots, D(m), \ldots$ such that for each m, $D(m)$ is a *definition* of G_m as an object recursive in $\mathfrak{F}$. Such definitions may be coded in the natural numbers. Hence what we actually generate is a relation

(3) $$G \leq_e \mathfrak{F}$$

where $e \in N$ is a *code for* G from $\mathfrak{F}$.

Partial functionals $\lambda \mathfrak{F} \cdot [e]^{\mathfrak{F}}$ are naturally introduced in this framework, where

* Research supported by NSF Grant GP 34091X.

$$(4) \qquad [e]^{\mathfrak{F}} \simeq G$$

holds just in case $G \leq_e \mathfrak{F}$. But they are not given the central role as in [6]. Thus one avoids well-known annoying aspects of dealing with the relation of being *partial recursive in*, such as failure of transitivity. The tack here is thus completely opposite to that taken by Platek [11], which makes thoroughgoing use of partial objects to act as arguments as well as functionals. Since much work on recursion in higher types, e.g. Sacks [12], has concentrated on getting information about all the total G of a given type recursive in a given $\mathfrak{F}$ (the sections of $\mathfrak{F}$), it may be that the present kind of formulation is more directly useful for current purposes.

The equivalence of (1) and (2) for $p = 0$ may be established by fairly routine methods using the ordinary recursion theorem. To prepare the ground for deeper results, one must show how higher types can be eliminated in the $\leq$ relation restricted to given types. This is provided in §6 by a theorem on the *normalization* of the tree of predecessors of a derived $G \leq_e \mathfrak{F}$. The proof is by direct adaptation of standard normalization techniques which have been used with infinite terms for functionals in proof theory.[1] Some immediate consequences of normalization are presented in §7. In particular, it is shown that certain reduced schemata R1–R5 for restricted types are equivalent to I–V on those types. These schemata are of the same character as Kleene's S1–S9, but fall out quite naturally at this point. Some possible directions for further work are explored in §7. The development should move fairly directly into a treatment of ordinals, selection theorems and hierarchies associated with (suitable) $\mathfrak{F}$. The present schemata should also be compared with other proposed generalizations of recursion theory to arbitrary structures, particularly those of Moschovakis [10] and Platek [11]. In this connection a new rule VI called *unique selection* is introduced, which happens to be derivable when the only ground domain is N.

While the rules here are quite perspicuous and intuitively appealing, their consideration is open to the same objections as for Kleene's S1–S9. The paper concludes with some speculation and questions to see if there is more basic significance to such schemata.

We concentrate throughout on definitions and statements of results. Most proofs are straightforward and/or adaptations of arguments

[1] Platek had also used a kind of normalization in [11] to get equivalence with Kleene's theory over N and for selection theorems, but the details there are quite different and more complicated. It is interesting that normalization is not needed to establish the equivalence results here.

from the literature; only indications are given for the most part, with some delicate points given more attention. Two principal sources (conceptually and technically) are: Moschovakis [9] and Tait [14]; connections with this and other work is explained where appropriate in the text.

Notation: The *type symbols* (t.s.) over $p+1$ ground domains are generated as follows:

(1) (i) $0^{(i)}$ is a t.s. for $0 \leq i \leq p$
(ii) if σ, τ are t.s. then $(\sigma \to \tau)$ is a t.s.

$0^{(0)}$ is also written simply as 0, and $n+1 = (n \to 0)$. $(\sigma_1 \to \cdots \to \sigma_n \to \tau)$ abbreviates $(\sigma_1 \to (\ldots(\sigma_n \to \tau)\ldots))$ (association to the right). $\bar{\sigma}$ denotes a sequence $\bar{\sigma} = \langle \sigma_1, \ldots, \sigma_n \rangle$.

The *level* $\text{lev}(\sigma)$ of a type is defined by:

(2) (i) $\text{lev}(0^{(i)}) = 0$
(ii) $\text{lev}(\sigma \to \tau) = \max(\text{lev}(\sigma) + 1, \text{lev}(\tau))$.

For sequences, $\text{lev}(\bar{\sigma}) = \max \text{lev}(\sigma_i)$. The *pure t.s.* are generated by (1)(i) and (1)(ii) restricted as follows: if σ is pure and $\text{lev}(\tau) = 0$ then $(\sigma \to \tau)$ is pure. These are just the types n when $p = 0$.

The *maximal type structure* $\langle M_\sigma \rangle$ over $M^{(0)}, \ldots, M^{(p)}$ is given by:

(3) (i) $M_{0^{(i)}} = M^{(i)}$
(ii) $M_{(\sigma \to \tau)} = M_\tau^{M_\sigma}$.

Let $M = \cup M_\sigma$ [all t.s. σ]. We use $F, G, H, \alpha, \beta, \gamma$ to range over M. $\text{Type}(F) = \sigma$ for $F \in M_\sigma$. $\mathfrak{F}$, $\mathfrak{G}$, $\mathfrak{A}$, $\mathfrak{B}$, are used for sequences of elements of M. For $\mathfrak{F} = \langle F_1, \ldots, F_n \rangle$, $\text{Typ}(\mathfrak{F}) = \bar{\sigma} = \langle \sigma_1, \ldots, \sigma_n \rangle$ where $\sigma_i = \text{Typ}(F_i)$. If $\text{Typ}(F) = (\sigma \to \tau)$, $\text{Typ}(G) = \sigma$ then the value of F at G is denoted both by $F(G)$ and FG. For $\text{Typ}(F) = (\sigma_1 \to \cdots \to \sigma_n \to \tau)$, $\text{Typ}(G_i) = \sigma_i$, $FG_1 \cdots G_n$ designates $(\cdots(FG_1)\cdots)G_n$ (association to the left). Λ denotes the empty sequence.

a, b, c, range over $\cup_{i=0}^{p} M^{(i)}$ and all remaining letters $d, \ldots, z$ range over N. $P_k (k = 0, 1, \ldots)$ is any standard enumeration of all primitive recursive functions $P_k : N^{n_k} \to N$ $(n_k \geq 0)$. $(\ ,\ )$ is a standard prim. rec. pairing function on $N^2 \to N$. T.s. are coded as natural numbers using this function. By the *primitive recursion theorem* we mean the result giving for each prim. rec. $F(z, x, s)$ an e satisfying $P_e(x) = F(e, x, \overline{P_e}(x))$ for all x (where $\overline{P_e}(x)$ is the sequence number for $\langle P_e(i) \rangle_{i<x}$). This is easily proved by the same method as the ordinary recursion theorem.

2. The generating rules

Each rule except the first "axiom" has the general form

$$\frac{\cdots \beta_i \leq_{e_i} \mathfrak{A}_i \cdots}{G \leq_{e'} \mathfrak{F}}. \tag{1}$$

The code e' is built from one or more of the e_i. In the following list, this formation is shown under "code", and any restrictions on types involved is given in the r.h. column.

Rule	Code	Type restrictions
I. *Identity*		
$F_i \leq_e \mathfrak{F}$	$e = (1, i, \bar{\sigma})$	
II. *Application*		
$\dfrac{G_1 \leq_{e_1} \mathfrak{F} \quad G_2 \leq_{e_2} \mathfrak{F}}{G_1(G_2) \leq_{e'} \mathfrak{F}}$	$e' = (2, e_1, e_2)$	$\begin{cases} \mathrm{Typ}\,(G_1) = (\rho \to \tau) \\ \mathrm{Typ}\,(G_2) = \rho \end{cases}$
III. *Abstraction*		
$\dfrac{\forall\, \alpha \in M_\rho[G(\alpha) \leq_e \mathfrak{F}, \alpha]}{G \leq_{e'} \mathfrak{F}}$	$e' = (3, e)$	$\mathrm{Typ}\,(G) = (\rho \to \tau)$
IV. *Secondary reference*		
$\dfrac{d \leq_e \mathfrak{F} \quad a \leq_d \mathfrak{F}}{a \leq_{e'} \mathfrak{F}}$	$e' = (4, e, i)$	$\mathrm{Typ}\,(a) = 0^{(i)}$
V. *Primitive recursion*		
$\dfrac{m_i \leq_{e_i} \mathfrak{F} (1 \leq i \leq n_k)}{P_k(m_1, \ldots, m_{n_k}) \leq_{e'} \mathfrak{F}}$	$e' = (5, k, e_1, \ldots, e_{n_k})$.	

It suffices to take a few special cases of V to obtain all, using I–IV.

2.1. Definition: (i) $G \leq_e \mathfrak{F}$ holds if it is derivable using I–V, i.e., if $\langle G, e, \mathfrak{F}\rangle$ is in the smallest class of triples closed under these rules.
(ii) $G \leq \mathfrak{F}$ if $\exists\, e(G \leq_e \mathfrak{F})$
(iii) $e \in C^{\mathfrak{F}} \Leftrightarrow \exists\, G(G \leq_e \mathfrak{F})$.

2.2. Lemma: *If $G \leq_e \mathfrak{F}$ and $G' \leq_e \mathfrak{F}$ then $G = G'$.*

2.3. Definition: For $e \in C^{\mathfrak{F}}$, $[e]^{\mathfrak{F}}$ is the unique G with $G \leq_e \mathfrak{F}$.

2.4. DEFINITION: The prim. rec. function $\mathrm{Ty}(z)$ is defined by:

(i) $\mathrm{Ty}(1, i, \bar{\sigma}) = (\bar{\sigma}; \sigma_i)$

(ii) $\mathrm{Ty}(2, z_1, z_2) = (\bar{\sigma}; \rho)$ when $\mathrm{Ty}(z_1) = (\bar{\sigma}; (\rho \to \tau))$, $\mathrm{Ty}(z_2) = (\bar{\sigma}; \rho)$

(iii) $\mathrm{Ty}(3, z) = (\bar{\sigma}; (\rho \to \tau))$ when $\mathrm{Ty}(z) = (\bar{\sigma}, \rho; \tau)$

(iv) $\mathrm{Ty}(4, z, i) = (\bar{\sigma}; 0^{(i)})$ when $\mathrm{Ty}(z) = (\bar{\sigma}; 0)$

(v) $\mathrm{Ty}(5, k, z_1, \ldots, z_{n_k}) = (\bar{\sigma}; 0)$ when $\mathrm{Ty}(z_1) = \cdots = \mathrm{Ty}(z_{n_k}) = (\bar{\sigma}; 0)$

(vi) $\mathrm{Ty}(z) = 0$ in all other cases.

Thus $\mathrm{Ty}(e) = (\bar{\sigma}; \tau)$ when $\mathrm{Typ}(\mathfrak{F}) = \bar{\sigma}$, $e \in C^{\mathfrak{F}}$ and $\mathrm{Typ}([e]^{\mathfrak{F}}) = \tau$. We also write $e \in C_\tau^{\mathfrak{F}}$ in this case.

2.5. DEFINITION: For any z with $\mathrm{Ty}(z) = (\bar{\sigma}; \tau)$, $[z]$ is taken to be the (possibly) partial functional from $M_{\sigma_1} \times \cdots \times M_{\sigma_n} \to M_\tau$ with $[z](\mathfrak{A})$ defined just in case $z \in C^{\mathfrak{A}}$, in which case $[z](\mathfrak{A}) = [z]^{\mathfrak{A}}$. More generally, take $[z]^{\mathfrak{F}}(\mathfrak{A}) = [z](\mathfrak{F}, \mathfrak{A}) = [z]^{\mathfrak{F},\mathfrak{A}}$, which is defined just in case $z \in C^{\mathfrak{F},\mathfrak{A}}$.

When $\mathrm{Ty}(z) = (\Lambda; \tau)$ i.e., with $\bar{\sigma}$ empty, then $[z]$ is an object of M_τ, if its value is defined at all, i.e. if $z \in C^\Lambda$. As examples: for $n_k = 1$, $[(3, (5, k, (1, 1, \langle 1 \rangle)))] = P_k$ which is in M_1 and for $n_k = 0$, $[(5, k, \Lambda)] = P_k$ in M_0.

3. Derived rules

These take the form

$$(1) \qquad \frac{\cdots \beta_i \le \mathfrak{A}_i \cdots}{G \le \mathfrak{F}},$$

i.e. if each $\beta_i \le \mathfrak{A}_i$ is derivable by I–V then so is $G \le \mathfrak{F}$. To prove these it is shown how, given z_i with $\beta_i \le_{z_i} \mathfrak{A}_i$, to find z' with $G \le_{z'} \mathfrak{F}$. In all cases this is done by the primitive recursion theorem, so that z' may be chosen as a primitive recursive function of one or more of the z_i.

3.1. LEMMA: *The following are all derivable rules*:

(i) (Expansion) $\qquad \dfrac{G \le \mathfrak{F}}{G \le F, \mathfrak{F}}$

(ii) (Interchange) $\dfrac{G \le \mathfrak{F}}{G \le \pi_{i,j}(\mathfrak{F})}$ $\quad \pi_{i,j}$ *interchanges* F_i, F_j

(iii) (Permutation) $\dfrac{G \le \mathfrak{F}}{G \le \pi(\mathfrak{F})}$ $\quad \pi$ *any permutation*

(iv) (Identification) $$\frac{G \le F, F, \mathfrak{F}}{G \le F, \mathfrak{F}}.$$

3.2. Lemma (Abstraction—Extended Enumeration): *If*

$$\forall\, \alpha \in M_\rho[D(\alpha) \le_e \mathfrak{F}, \alpha \text{ and } G(\alpha) \le_{D(\alpha)} \mathfrak{F}, \alpha]$$

then

$$G \le \mathfrak{F}.$$

This is proved using Abstraction and Secondary Reference. It is first established for G of type $(\rho \to 0^{(i)})$ and then in general for types $(\rho \to \tau) = (\rho \to \tau_1 \to \cdots \to \tau_m \to 0^{(i)})$.

Note that Secondary Reference can be written in the form

(2) $$[[e]^{\mathfrak{F}}]^{\mathfrak{F}} = [e']^{\mathfrak{F}}$$

whenever $e \in C^{\mathfrak{F}}$, $[e]^{\mathfrak{F}} \in C_{0^{(i)}}^{\mathfrak{F}}$ and $e' = (4, e, i)$. It might also be called a *reflection principle.* With any τ in place of $0^{(i)}$ it is the following special case of 3.2.

3.3. Corollary: *We have a prim. rec. function* $T(z)$ *such that*

$$[[e]^{\mathfrak{F}}]^{\mathfrak{F}} = [T(e)]^{\mathfrak{F}}$$

whenever $e \in C^{\mathfrak{F}}$ and $[e]^{\mathfrak{F}} \in C^{\mathfrak{F}}$.

3.4. Theorem (Substitution, or Transitivity): *We have a prim. rec. function* $\mathrm{Sub}(w, z)$ *such that whenever*

$$F \le_f \mathfrak{F} \text{ and } G \le_e F, \mathfrak{F}$$

then

$$G \le_{\mathrm{Sub}(f,e)} \mathfrak{F}.$$

Proof: The definition of Sub (which must be used in §6 below) is by the primitive recursion theorem:

(0) $$\mathrm{Sub}(f, e) = P_k(f, e) = H(k, f, e),$$

for suitable k and prim. rec. H. This follows out the inductive generation of $G \le_e F, \mathfrak{F}$. Let $\mathrm{Typ}(F) = \sigma_0$, $\mathrm{Typ}(\mathfrak{F}) = \bar{\sigma} = \langle \sigma_1, \ldots, \sigma_n \rangle$, $\sigma_0^* \bar{\sigma} = \langle \sigma_0, \sigma_1, \ldots, \sigma_n \rangle$.

(i) (a) for $e = (1, 0, \sigma_{0*}\bar{\sigma})$ (which makes $G = F$) take $\mathrm{Sub}(f, e) = f$

(b) for $e = (1, i+1, \sigma_0^* \bar{\sigma})$ (which makes $G = F_i$) take $\mathrm{Sub}(f, e) = (1, i, \bar{\sigma})$

(ii) $\mathrm{Sub}(f, (2, e_1, e_2)) = (2, \mathrm{Sub}(f, e_1), \mathrm{Sub}(f, e_2))$
(iii) $\mathrm{Sub}(f, (3, e_1)) = (3, \mathrm{Sub}(f, e_1))$
(iv) (cf. below)
(v) $\mathrm{Sub}(f, (5, k, e_1, \ldots, e_{n_k})) = (5, k, \mathrm{Sub}(f, e_1), \ldots, \mathrm{Sub}(f, e_{n_k}))$.

In the case corresponding to Indirect Reference we will have $e = (4, e_0, i)$ and d where $d \leq_{e_0} F, \mathfrak{F}$ and $G \leq_d F, \mathfrak{F}$. (Here G is an object of type $0^{(i)}$.) Proceeding by induction we have for $e_1 = \mathrm{Sub}(f, e_0)$:

$$d \leq_{e_1} \mathfrak{F} \text{ and } G \leq_{\mathrm{Sub}(f,d)} \mathfrak{F}.$$

But $\mathrm{Sub}(f, d) = \mathrm{Sub}(f, [e_1]^{\mathfrak{F}}) = P_k(f, [e_1]^{\mathfrak{F}})$. Let H_1 be prim. rec. with $\ell \leq_{H_1(\ell)} \mathfrak{F}$ for any ℓ. Then from $f \leq_{H_1(f)} \mathfrak{F}$ and $[e_1]^{\mathfrak{F}} \leq_{e_1} \mathfrak{F}$ we get $P_k(f, [e_1]^{\mathfrak{F}} \leq_{e'} \mathfrak{F}$ for $e' = (5, k, H_1(f), e_1)$. Thus we take (iv) $\mathrm{Sub}(f, (4, e_0, i)) = (4, (5, k, H_1(f), \mathrm{Sub}(f, e_0)), i)$.

This completes the definition.

Substitution can be rewritten

$$[e]^{[f]^{\mathfrak{F}}, \mathfrak{F}} = [\mathrm{Sub}(f, e)]^{\mathfrak{F}}. \tag{3}$$

3.5. Corollary (Numerical parameters): *We have a prim. rec.* $S_0(z, x)$ *such that whenever* $e \in C^{m,\mathfrak{F}}$ *then* $S_0(e, m) \in C^{\mathfrak{F}}$ *and*

$$[e]^{m,\mathfrak{F}} = [S_0(e, m)]^{\mathfrak{F}}.$$

The basic rules II–V are *uniquely invertible.* For example, $G \leq_{(2, e_1, e_2)} \mathfrak{F}$ implies $e_1, e_2 \in C^{\mathfrak{F}}$ and $G = G_1(G_2)$ for $G_1 = [e_1]^{\mathfrak{F}}$, $G_2 = [e_2]^{\mathfrak{F}}$. As another example, $a \leq_{(4, e, i)} \mathfrak{F}$ implies $e \in C^{\mathfrak{F}}$ and $m = [e]^{\mathfrak{F}} \in C^{\mathfrak{F}}$ and $a \leq_m \mathfrak{F}$. Put another way: whenever $G \leq_e \mathfrak{F}$ there is a *unique derivation* of this by I–V.

3.6. Theorem (Enumeration): *For each* $\bar{\sigma}, \tau$ *we can find an* e *such that for any* $\mathfrak{F}$ *of type* $\bar{\sigma}$ *and any* z,

$$z \in C_\tau^{\mathfrak{F}} \Leftrightarrow e \in C_\tau^{z,\mathfrak{F}}$$

and

$$[z]^{\mathfrak{F}} = [e]^{z,\mathfrak{F}}$$

when $z \in C^{\mathfrak{F}}$.

The Recursion Theorem for the partial functionals given by Definition 2.5 follows directly from 3.5, 3.6. This is not needed below.

4. Relationship with Kleene's schemata

We take $p = 0$ in this section, i.e. N is the sole domain of individuals.

4.1. Definition: For G of type $m + 1$ and $\mathfrak{F}$ of type $\bar{\sigma}$ where the σ_i are pure, we say G is Kleene recursive in $\mathfrak{F}$ via e if

$$\forall \alpha \in M_m[G(\alpha) = \{e\}(\mathfrak{F}, \alpha)].$$

The main result of this section is that $G \leq \mathfrak{F} \Leftrightarrow G$ is Kleene recursive in $\mathfrak{F}$. The arguments used are analogous to certain ones given by Moschovakis [9] for functionals of type ≤ 3. His results concern recursion of type 1 objects G in $\mathfrak{F} = (F^3, F^2, F^1)$. Moschovakis defines by induction (for arbitrary $\mathfrak{F}$) a set $N(\mathfrak{F})$ of numbers and for each $e \in N(\mathfrak{F})$ a function $f_e^{\mathfrak{F}}$ of type 1. He shows that G (of type 1) is Kleene recursive in $\mathfrak{F} \Leftrightarrow G = f_e^{\mathfrak{F}}$ for some $e \in N(\mathfrak{F})$. If we write $G \leq_e \mathfrak{F}$ in place of $G = f_e^{\mathfrak{F}}$ then the clauses for the inductive definition of [9] may be considered to be special consequences of I–V for pure types of level ≤ 3. (In particular, abstraction is applied there only to types 0 and 1.)

If we follow the method of proof of Theorem 3 in [9] we obtain:

4.2. Lemma: *There is a prim. rec. function $q(z, s)$ such that whenever $\{e\}(\mathfrak{A} * \bar{x}) \simeq y$ where all α_i in $\mathfrak{A}$ are of pure type > 0, and $\bar{x} = \langle x_0, \ldots, x_{k-1}\rangle$, then $y \leq_{q(e, \bar{x})} \mathfrak{A}$.*

The following is then a corollary. However, we shall give a simpler direct proof using the machinery developed in §3 above.

4.3. Theorem: *There is a prim. rec. function $q_1(z)$ such that whenever $\{e\}(\mathfrak{A}) \simeq y$, then $y \leq_{q_1(e)} \mathfrak{A}$.*

Proof: $q_1(z)$ or—as we shall simply write it in the proof—$q(z)$, is found as a certain $P_k(z)$ by the prim. recursion theorem. k is chosen to make possible the following proof by induction on e corresponding to Kleene's schemata S1–S9. S1, S2, S3, and S6 are treated quite simply. As is familiar, S5 can be derived from S9 using ordinary primitive recursion, so it is also obtained directly. This leaves S4, S8, and S9.

S4. $\{e\}(\mathfrak{A}) \simeq \{f\}(\{h\}(\mathfrak{A}), \mathfrak{A})$.

Here $e = \langle 4, \langle r_0, \ldots, r_n \rangle, f, h \rangle$ in Kleene's indexing, so $f < e$, $h < e$. By induction we have for $\{h\}(\mathfrak{A}) \simeq x$ and $\{f\}(x, \mathfrak{A}) \simeq y$:

$$x \leq_{q(h)} \mathfrak{A} \quad \text{and} \quad y \leq_{q(f)} x, \mathfrak{A}.$$

Thus by our substitution Theorem 3.4, if we take $q(e) = \mathrm{Sub}\,(q(h), q(f))$ we have $y \leq_{q(e)} \mathfrak{A}$.

S8. $\{e\}(\alpha^j, \mathfrak{B}) \simeq \alpha^j(\lambda\beta^{j-2}\{f\}(\alpha^j, \beta^{j-2}, \mathfrak{B}))$.

Here again $f < e$ is found primitive recursively from the Kleene index e. We drop the superscripts j, $j-2$; β ranges over M_{j-2}. $\{f\}(\alpha, \beta, \mathfrak{B})$ is defined for each β if $\{e\}(\alpha, \mathfrak{B})$ is defined. By induction hypothesis for each $\beta \in M_{j-2}$

$$\{f\}(\alpha, \beta, \mathfrak{B}) \leq_{q(f)} \alpha, \beta, \mathfrak{B}.$$

By permutation we get prim. rec. H with $\{f\}(\alpha, \beta, \mathfrak{B}) \leq_{H(q(f))} \alpha, \mathfrak{B}, \beta$. Then by the Abstraction rule III,

$$\lambda\beta \cdot \{f\}(\alpha, \beta, \mathfrak{B}) \leq_{(3, H(q(f))} \alpha, \mathfrak{B}.$$

Choosing e_1 with $\alpha \leq_{e_1} \alpha, \mathfrak{B}$ by rule I we may take $q(e) = (2, e_1, (3, H(q(f))))$ in view of the Application rule II.

S9. $\{e\}(z, \mathfrak{A}, \mathfrak{B}) \simeq \{z\}(\mathfrak{A})$.

By induction hypothesis, $\{z\}(\mathfrak{A}) \leq_{q(z)} \mathfrak{A}$ so $\{z\}(\mathfrak{A}) \leq_{H_1(q(z))} z, \mathfrak{A}, \mathfrak{B}$ for suitable prim. rec. H_1. Using $q(z) = P_k(z)$ we may find prim. rec. H_2 with $H_1(q(z)) \leq_{H_2(k)} z, \mathfrak{A}, \mathfrak{B}$ by rule V.

This permits us to take $q(e) = (4, H_2(k), 0)$ by secondary reference.

4.4. Corollary: *We have a prim. rec. function $q_2(z)$ such that whenever G is Kleene recursive in $\mathfrak{F}$ via e then $G \leq_{q_2(e)} \mathfrak{F}$.*

For a converse we must use a representation of finite type objects by those of pure type, more specifically an operation * which maps M_σ into $M_{\mathrm{lev}(\sigma)}$ for each σ such that * preserves application:

(1) $$\alpha(\beta) = \alpha^*(\beta^*), \quad \text{for } \alpha \in M_{\rho\to 0} \text{ and } \beta \in M_\rho.$$

More generally, if $\alpha \in M_{\rho_1 \to \cdots \to \rho_k \to 0}$ and $\beta_i \in M_{\rho_i}$ then we should have

(2) $$\alpha\beta_1 \cdots \beta_m = \alpha^*(\langle \beta_1^*, \ldots, \beta_k^* \rangle_m)$$

where $m + 1 = \text{level } \sigma$ and $\langle \gamma_1, \ldots, \gamma_k \rangle_m$ is an object of type m representing the k-tuple $\langle \gamma_1, \ldots, \gamma_k \rangle$ (for $\gamma_i \in M_{\ell_i}$, $\ell_i \leq m$). A definition of * is given in [5] §5.

4.5. THEOREM: *There is a primitive recursive function* $p(z)$ such that whenever $G \leq_e F_1, \ldots, F_n$ and G is of type $(\tau_1 \to \cdots \to \tau_k \to 0)$ then for all β_i of type τ_i

$$\{p(e)\}(F_1^*, \ldots, F_n^*, \beta_1^*, \ldots, \beta_k^*) = G\beta_1 \cdots \beta_k.$$

This is analogous to Theorem 4 of [9], slightly complicated by the larger type structure here. The proof is again by the primitive recursion theorem, following the inductive generation of $G \leq_e \mathfrak{F}$.

4.6. COROLLARY: *If* G *and* $F_1, \ldots, F_n$ *are of pure type and* $G \leq_e \mathfrak{F}$ *then* G *is Kleene recursive in* $\mathfrak{F}$ *via* $p(e)$.

N.B. These results do not establish that the partial functionals $\lambda \mathfrak{A} \cdot [z](\mathfrak{A})$ of Definition 2.5 are the same as the Kleene partial recursive functionals $\lambda \mathfrak{A} \cdot \{w\}(\mathfrak{A})$. The above statements only assert inclusions, e.g. that $\lambda \mathfrak{A} \cdot [e](\mathfrak{A})$ is a subfunction of $\lambda \mathfrak{A} \cdot \{p(e)\}(\mathfrak{A})$. But using invertibility of the rules I–V remarked at the end of §3 and corresponding invertibilities for S1–S9, it should be possible to refine the arguments for this section to show that the two theories of partial recursive functionals are indeed equivalent.

5. Connections with the use of infinite terms

Infinite terms of finite types have been used in proof theory, for example, those built up as follows.

(1) (*Variables*) Every variable x^σ, y^σ, $z^\sigma, \ldots$ is a term of type σ.

(2) (*Application*) If t_1 is of type $(\rho \to \tau)$ and t_2 is of type ρ then $t_1 t_2$ is of type τ.

(3) (*Abstraction*) If t is of type τ then $\lambda x^\rho \cdot t$ is of type $(\rho \to \tau)$.

(4) (*Sequencing*) If t_n is of type τ for each n and there are at most finitely many free variables in all t_n, then $\langle t_n \rangle_{n \in N}$ is of type $(0 \to \tau)$.

(5) (*Primitive recursion*) If $t_1, \ldots, t_{n_k}$ are of type 0 then $P_k(t_1, \ldots, t_{n_k})$ is of type 0.

Let *Tm* be the set of arbitrary terms generated by (1)–(5). In certain cases it is useful to fix some of the variables as parameters or *constants*; in this case we denote them instead by c^σ, $d^\sigma, \ldots$.

For each $t \in Tm$ and assignment $\mathfrak{F}$ to the free variables of t, let $[\![t]\!](\mathfrak{F})$ be the value of t at $\mathfrak{F}$ (defined in the natural way). When $\mathfrak{F}$ is assigned to constants and t is closed we write $[\![t]\!]^{\mathfrak{F}}$ for this. $[\![t]\!]$ denotes $[\![t]\!](\Lambda)$.

Each $t \in Tm$ has the structure of a coded well-founded tree in N. It may thus be represented in a canonical way by a function J_t of type 1. We call t recursive if J_t is recursive, similarly for prim. rec., etc.

Gödel [2] made use of a class PR of primitive recursive functionals (based on impredicative primitive recursion $R : R\alpha\beta 0 = \alpha$, $R\alpha\beta x' = \beta(R\alpha\beta x)x$ for any suitable combination of types) in his functional interpretation of (intuitionistic) number theory. One easily associates with each $G \in PR$ a (prim. rec.) term t_G such that $G = [\![t_G]\!]$. Tait [14] applied the Gentzen-Schütte method of normalization to these terms. In this way he could characterize the 1-section of PR (and thence recapture Kreisel's characterization of the provably recursive functions of arithmetic). My work [1] made use of a direct extension of Tait's for terms t with a constant c^2, to characterize the 1-section generated by $PR + F^2$ for any F^2.

It is inappropriate for general recursion theory to restrict the structure of terms in advance, e.g., to those which are prim. rec. Rather one wants to allow the set of terms to increase as the stock of functions which are defined by terms increases. In unpublished notes I introduced the following for any $\mathfrak{F}$: $Tm^{\mathfrak{F}}$ is the smallest set which contains the constants $c_1^{\sigma_1}, \ldots, c_n^{\sigma_n}$ and the variables of each type, is closed under application, abstraction, primitive recursion and:

(4′) (*Autonomous enumeration*) If $t_n \in Tm^{\mathfrak{F}}$ for each n and $t = \langle t_n \rangle_{n \in N}$ and $J_t = [\![s]\!]^{\mathfrak{F}}$ (under the assignment $c_i^{\sigma_i} \mapsto F_i$) for some closed s in $Tm^{\mathfrak{F}}$ then $t \in Tm^{\mathfrak{F}}$.

I showed that
(‡) for $\mathfrak{F}$ of level ≤ 2, the functionals $[\![t]\!]^{\mathfrak{F}}$ are the same as the functionals Kleene recursive in $\mathfrak{F}$.

This was re-established by Schwichtenberg and Wainer [16] by a different method. They also showed that (‡) is in general false if we replace the type level 2 by a level $m > 2$.

The relation $G = [\![t]\!]^{\mathfrak{F}}$ has certain analogies to $G = [e]^{\mathfrak{F}}$ above. The closure conditions (1)–(3) and (5) correspond to the rules I–III and V resp., while as we have seen IV can be used to give a form of enumeration which is analogous to (4)′. The analogy can be made even closer by generating a class of numerical codes for terms, rather than the terms themselves. But there still remains an essential difference, namely: in the inductive generation of the relation $G = [\![t]\!]^{\mathfrak{F}}$, *$\mathfrak{F}$ is kept fixed*, but in that of $G = [e]^{\mathfrak{F}}$, *$\mathfrak{F}$ is variable.* We can associate with each term t a code e_t such that for any $\mathfrak{F}$, $[e_t]^{\mathfrak{F}} = [\![t]\!]^{\mathfrak{F}}$, but an attempt to go in the reverse direction breaks down at Abstraction (III).

Returning to the subject of normalization, if a term t contains a sub-term of the form $(\lambda x^{\rho}\cdot t_1)t_2$ it is said to be *reducible.* Otherwise t is said to be *irreducible* or in *normal form* (n.f.). Tait's method provides a suitable order of attack on sub-terms so that repetition of the *reduction step*

$$(\lambda x^{\rho}\cdot t_1)t_2 \mathrel{=\!\!|} \mathrm{Sub}\,(^{x}_{t_2})t_1$$

leads eventually from each t to a term t^* in n.f. with $[\![t^*]\!](\mathfrak{F}) = [\![t]\!](\mathfrak{F})$ for any $\mathfrak{F}$. The method of proof in the next section is similar, except that the choice of e^* depends also on $\mathfrak{F}$, i.e. we must work with pairs $(e;\mathfrak{F})$ for $e\in C^{\mathfrak{F}}$.

6. Normalization

It is convenient here to use the following abbreviations.

6.1. Definition:

(i) $\mathrm{Id}\,(e)\Leftrightarrow e$ is of the form $(1, i, \bar{\sigma})$ for $1\le i\le \ell h(\bar{\sigma})$

(ii) $e_1e_2 = (2, e_1, e_2)$

(iii) $\hat{e} = (3, e)$

(iv) $\bar{e} = (4, e)$

(v) $p_k(e_1,\ldots, e_{n_k}) = (5, k, e_1,\ldots, e_{n_k})$.

We write $e_1e_2\cdots e_m$ for $(\cdots(e_1e_2)\cdots)e_m$ (association to the left).

6.2. Definition: For $e\in C^{\mathfrak{F}}$, *the direct predecessors* $(z;\mathfrak{A})$ of $(e;\mathfrak{F})$ are determined as follows:

(i) If $\mathrm{Id}\,(e)$ then $(e;\mathfrak{F})$ has no direct preds

(ii) $(e_1e_2;\mathfrak{F})$ has $(e_1;\mathfrak{F})$ and $(e_2;\mathfrak{F})$ as its direct preds

(iii) $(\hat{e};\mathfrak{F})$ has $(e;\mathfrak{F},\alpha)$ for each $\alpha\in M_{\rho}$ as its direct preds, where $\mathrm{Ty}\,(\hat{e}) = (\bar{\sigma};\rho\to\tau)$

(iv) $(\bar{e};\mathfrak{F})$ has $(e;\mathfrak{F})$ and $([e]^{\mathfrak{F}};\mathfrak{F})$ as its direct preds

(v) $(p_k(e_1,\ldots, e_{n_k});\mathfrak{F})$ has $(e_i;\mathfrak{F})$ $(1\le i\le n_k)$ as its direct preds.

By unique invertibility, if $(z;\mathfrak{A})$ is a direct pred. of $(e;\mathfrak{F})$ then $z\in C^{\mathfrak{A}}$.

6.3. Definition (Transitive closure): For $e\in C^{\mathfrak{F}}$, $(z;\mathfrak{A})\le(e;\mathfrak{F})$ if there is a sequence $(z;\mathfrak{A}) = (z_0;\mathfrak{A}_i),\ldots,(z_i;\mathfrak{A}_i),\ldots,(z_n;\mathfrak{A}_n) = (e;\mathfrak{F})$ such that each $(z_i;\mathfrak{A}_i)$ is a direct pred. of $(z_{i+1};\mathfrak{A}_{i+1})$. $(z;\mathfrak{A}) < (e;\mathfrak{F})$ if this holds for $n>0$.

6.4. LEMMA: $<$ *is well founded.*

6.5. DEFINITION: $(e;\mathfrak{F})$ is called *reducible* if there exist z_1, z_2 and $\mathfrak{A}$ with $(\hat{z}_1 z_2;\mathfrak{A}) \leq (e;\mathfrak{F})$. Otherwise, $(e;\mathfrak{F})$ is called *irreducible* or *normal.*

6.6. DEFINITION:
(i) $\mathrm{rnk}\,(\hat{e}_1 e_2) = \mathrm{lev}\,(\rho \to \tau)$ where $\mathrm{Ty}\,(\hat{e}_1) = (\bar{\sigma};\rho \to \tau)$.
(ii) For $e \in C^{\mathfrak{F}}$, with $(e;\mathfrak{F})$ reducible

$$\mathrm{rnk}\,(e;\mathfrak{F}) = \sup\{\mathrm{rnk}\,(\hat{z}_1 z_2)|\text{ for some } \mathfrak{A},\ (\hat{z}_1 z_2;\mathfrak{A}) \leq (e;\mathfrak{F})\},$$
$$\mathrm{rnk}\,(e;\mathfrak{F}) = 0 \text{ for } (e;\mathfrak{F}) \text{ normal.}$$

The function rnk is analogous to the *cut-rank* for derivations or terms; it measures the complexity of reducible sub-pairs $(z;\mathfrak{A})$.

We shall have to examine codes of the form *ef*, so that $\mathrm{Ty}\,(e)$ is of the form $(\bar{\sigma};\rho \to \tau)$. It is seen by induction that e has the form $e = e_0 e_1, \ldots, e_k$ where $\mathrm{Id}\,(e_0)$ or the form $e = \hat{e}_0 e_1, \ldots, e_k$, in either case with $k \geq 0$.

6.7. LEMMA: *If* $ef \in C^{\mathfrak{F}}$ *and* $\mathrm{rnk}\,(ef;\mathfrak{F}) \leq m$ *and* $\mathrm{lev}\,(e) \geq m+1$ *then for some* $k \geq 0$, $e = e_0 e_1, \ldots, e_k$ *and* $\mathrm{Id}\,(e_0)$.

This is analogous to [14], Lemma 3.

6.8. LEMMA: *If* $e \in C^{\mathfrak{F}}$ *and* $\pi_{ij}(e)$ *is given by* 3.1(ii) *so that* $\pi_{ij}(e) \in C^{\pi_{ij}(\mathfrak{F})}$, $[\pi_{ij}(e)]^{\pi_{ij}(\mathfrak{F})} = [e]^{\mathfrak{F}}$ *then* $\mathrm{rnk}\,(e) = \mathrm{rnk}\,(\pi_{ij}(e))$.

6.9. LEMMA: *Suppose*
(i) $F \leq_f \mathfrak{F}$ *and* $G \leq_e F, \mathfrak{F}$
(ii) $\mathrm{Ty}\,(f) = (\bar{\sigma};\rho)$, $\mathrm{Ty}\,(e) = (\rho, \bar{\sigma};\tau)$, $\mathrm{lev}\,(\rho) = m$
(iii) $\mathrm{rnk}\,(f;\mathfrak{F}) \leq m$ *and* $\mathrm{rnk}\,(e;F,\mathfrak{F}) \leq m$.
Then $\mathrm{rnk}\,(\mathrm{Sub}\,(f,e);\mathfrak{F}) \leq m$.

The proof of this proceeds by induction on e, using the definition of $\mathrm{Sub}\,(f,e)$ in 3.4.

This result is analogous to [14], Lemma 2. It is the main lemma for lowering cut-rank. For if $\hat{e} \in C^{\mathfrak{F}}$, $f \in C^{\mathfrak{F}}$, let $F = [f]^{\mathfrak{F}}$ and $G = [e]^{\mathfrak{F}}, F$, so also $G = [\hat{e}f]^{\mathfrak{F}}$. Suppose $\mathrm{Ty}\,(\hat{e}) = (\bar{\sigma};\rho \to \tau)$, $\mathrm{lev}\,(\rho \to \tau) = \max\,(\mathrm{lev}\,(\rho)+1, \mathrm{lev}\,(\tau)) = m+1$. When $\mathrm{rnk}\,(e;\mathfrak{F},F) \leq m$, $\mathrm{rnk}\,(f;\mathfrak{F}) \leq m$ then $\mathrm{rnk}\,(\hat{e}f;\mathfrak{F}) = m+1$. Then $G = [\mathrm{Sub}\,(f,\pi_{1,n+1}(e))]^{\mathfrak{F}}$ and $\mathrm{rnk}\,(\mathrm{Sub}\,(f,\pi_{1,n+1}(e));\mathfrak{F}) \leq m$ by 6.8, 6.9.

6.10. Theorem: *We have a prim. rec. function* $N_0(m, z)$ *such that whenever a code* $e \in C^{\mathfrak{F}}$ *and* $\text{rnk}\,(e; \mathfrak{F}) \leq m+1$ *then* $e' = N_0(m, e) \in C^{\mathfrak{F}}$, $[e']^{\mathfrak{F}} = [e]^{\mathfrak{F}}$ *and* $\text{rnk}\,(e'; \mathfrak{F}) \leq m$.

Proof: By the primitive recursion theorem, to be set up for a proof by induction on e, the following are the two cases which need attention:

(i) The code given is of the form ef. Let $\text{Ty}\,(e) = (\bar{\sigma}; \rho \to \tau)$.

  (a) $\text{Lev}\,(\rho \to \tau) \geq m+2$. In this case for $e = e_0 e_1 \cdots e_k$ with $k \geq 0$ and $\text{Id}\,(e_0)$, take $(ef)' = e_0 e_1' \cdots e_k' f'$. This gives the desired result by 6.7.

  (b) $\text{Lev}\,(\rho \to \tau) \leq m+1$. Form $e'f'$. If $e' \neq \hat{e}_0'$, take $(ef)' = e'f'$. If $e' = \hat{e}_0'$ take $(ef)' = \text{Sub}\,(f', \pi_{1,n+1}(e_0'))$. By the preceding, this gives the desired result.

(ii) The code given is of the form $\bar{e}$. Let k be a prim. rec. index for $\lambda z \cdot N_0(m, z)$. Take $(\bar{e})' = \overline{(5, k, e')}$. To show this gives the desired result, suppose $\bar{e} \in C^{\mathfrak{F}}$. The direct predecessors of $(\bar{e}; \mathfrak{F})$ are $(e; \mathfrak{F})$ and $(d; \mathfrak{F})$ where $d = [e]^{\mathfrak{F}}$. By induction, $[e']^{\mathfrak{F}} = [e]^{\mathfrak{F}}$, $[d']^{\mathfrak{F}} = [d]^{\mathfrak{F}}$, and $\text{rnk}\,(e'; \mathfrak{F}) \leq m$, $\text{rnk}\,(d'; \mathfrak{F}) \leq m$. Then for $e_1 = (5, k, e')$, $e_1 \in C^{\mathfrak{F}}$, $[e]^{\mathfrak{F}} = P_k([e']^{\mathfrak{F}}) = P_k([e]^{\mathfrak{F}}) = P_k(d) = d'$, and also $\text{rnk}\,(e_1; \mathfrak{F}) \leq m$.

6.11. Corollary: *We have a prim. rec. function* $N_1(m, z)$ *such that whenever* $e \in C^{\mathfrak{F}}$ *and* $m = \text{rnk}\,(e; \mathfrak{F}) < \omega$ *then* $e' = N_1(m, e) \in C^{\mathfrak{F}}$, $[e']^{\mathfrak{F}} = [e]^{\mathfrak{F}}$ *and* $\text{rnk}\,(e'; \mathfrak{F}) = 0$.

6.12. Theorem: *We have a prim. rec. function* $N(z)$ *such that whenever* $e \in C^{\mathfrak{F}}$ *then* $e^* = N(e) \in C^{\mathfrak{F}}$, $[e^*]^{\mathfrak{F}} = [e]^{\mathfrak{F}}$ *and* $(e^*; \mathfrak{F})$ *is normal.*

Proof: By the primitive recursion theorem. In the case that e is a product $e_1 e_2$, and e_1^* is not an $\hat{e}_0$, take $e^* = e_1^* e_2^*$. Otherwise take $e^* = N_1(m, e_1^* e_2^*)$ where $\text{Ty}\,(e_1) = (\bar{\sigma}; \rho \to \tau)$ and $m = \text{lev}\,(\rho \to \tau)$.

7. Consequences of normalization

7.1. Lemma: *Suppose* $e \in C^{\mathfrak{F}}$. *Then* $(e; \mathfrak{F})$ *is normal if and only if one of the following holds*:

(i) $\text{Id}\,(e_0)$

(ii) $e = e_0 e_1 \cdots e_k$ *where* $k \geq 1$, $\text{Id}\,(e_0)$ *and each* $(e_i, \mathfrak{F})$ *is normal* $(1 \leq i \leq k)$

(iii) $e = \hat{e}_1$ *where* $(e_1; \mathfrak{F}, \alpha)$ *is normal for all* $\alpha \in M_\rho$ $(\mathrm{Ty}\,(e) = (\bar{\sigma}; \rho \to \tau))$
(iv) $e = \bar{e}_1$ *where* $(e_1; \mathfrak{F})$ *and* $([e_1]^{\mathfrak{F}}; \mathfrak{F})$ *are normal*
(v) $e = p_k(e_1, \ldots, e_{n_k})$ *where each* $(e_i; \mathfrak{F})$ *is normal.*

When $(e; \mathfrak{F})$ is normal and $(z; \mathfrak{A}) \leq (e; \mathfrak{F})$ we can establish some *sub-type relationships* analogous to sub-formula properties for normal derivations. We do not state the most general form, but only some consequences for type levels.

7.2. LEMMA: *Suppose* $\mathrm{lev}\,(\mathfrak{F}) \leq m + 2$ *and* $G \leq_e \mathfrak{F}$ *with* $\mathrm{lev}\,(G) \leq m + 1$ *and* $(e; \mathfrak{F})$ *normal. Then each* $(z; \mathfrak{A}) \leq (e; \mathfrak{F})$ *is of the form* $(z; \mathfrak{F}, \mathfrak{B})$ *where* $\mathrm{lev}\,(\mathfrak{B}) \leq m$ *and* $\mathrm{lev}\,([z]^{\mathfrak{F},\mathfrak{B}}) \leq m + 1$.

This is direct from 7.1. In words, under the given hypotheses, $G \leq_e \mathfrak{F}$ is generated entirely from $H \leq \mathfrak{F}, \mathfrak{B}$ where the level of H is also $\leq m + 1$ and level $\mathfrak{B}$ is $\leq m$. Abstraction is applied only to types ρ of levels $\leq m$.

We get particularly nice relationships for pure types (defined for any number p of domains in §1). Call a sequence of types pure if it consists of pure types.

7.3. LEMMA: *Suppose* $(e; \mathfrak{F})$ *is normal and suppose* $\mathrm{Ty}\,(e) = (\bar{\sigma}; \tau)$ *is pure. Then each* $(z; \mathfrak{A}) \leq (e; \mathfrak{F})$ *has* $\mathrm{Ty}\,(z)$ *of the form* $(\bar{\sigma}, \rho; \mu)$ *and* $\mathrm{Ty}\,(z)$ *pure.*

If $[e]^{\mathfrak{F}}$ is of pure type $(\rho \to \nu)$ (thus $\mathrm{lev}\,(\nu) = 0$) we write $[e]^{\mathfrak{F}} = \lambda\beta \cdot [e]^{\mathfrak{F}}(\beta)$, thus expressing it by abstraction in terms of a level 0 object. This suggests the following schemata R1–R5 for a system of coding partially defined $\langle e\rangle(\mathfrak{A})$ of type level 0 ('R' for reduced).

In these schemata, $\mathfrak{A} = \langle \alpha_1, \ldots, \alpha_n \rangle$ where σ_i is of *pure type* σ_i; $\bar{\sigma} = \langle \sigma_1, \ldots, \sigma_n \rangle$ and $\mathrm{Typ}\,(\langle e\rangle(\mathfrak{A})) = \nu$ where ν is a type of level 0.

Scheme	Code	Type Restrictions
R1 $\langle e\rangle(\mathfrak{A}) = \alpha_j$	$e = (1, \bar{\sigma}, j)$	$\mathrm{lev}\,(\sigma_j) = 0$
R2 $\langle e\rangle(\mathfrak{A}) \simeq \alpha_j(\langle e_1\rangle(\mathfrak{A}))$	$e = (2, e_1, j)$	$\begin{cases} \sigma_j = (\nu_1 \to \nu_2) \\ \mathrm{Typ}\,(\langle e_1\rangle(\mathfrak{A})) = \nu_1 \end{cases}$
R3 $\langle e\rangle(\mathfrak{A}) \simeq \alpha_j(\lambda\beta^\rho \cdot \langle e_1\rangle(\mathfrak{A}, \beta))$	$e = (3, e_1)$	$\begin{cases} \sigma_j = ((\rho \to \nu_1) \to \nu_2) \\ \mathrm{Typ}\,(\langle e_1\rangle(\mathfrak{A}, \beta)) = \nu_1 \end{cases}$
R4 $\langle e\rangle(\mathfrak{A}) \simeq \langle\langle e_1\rangle(\mathfrak{A})\rangle(\mathfrak{A})$	$e = (4, e_1, \nu)$	$\begin{cases} \mathrm{Typ}\,(\langle e_1\rangle(\mathfrak{A})) = 0 \\ \mathrm{Typ}\,(\langle e\rangle(\mathfrak{A})) = \nu \end{cases}$
R5 $\langle e\rangle(\mathfrak{A}) \simeq P_k(\langle e_1\rangle(\mathfrak{A}), \ldots, \langle e_{n_k}\rangle(\mathfrak{A}))$	$e = (5, k, e_1, \ldots, e_k)$	$\begin{cases} \mathrm{Typ}\,(\langle e_i\rangle(\mathfrak{A})) = 0 \\ \cdot \end{cases}$

We can define prim. rec. $\mathrm{Ty}_N(z)$ so that $\mathrm{Typ}(\langle e\rangle(\mathfrak{A})) = \nu$ when $\mathrm{Ty}_N(e) = (\bar{\sigma}; \nu)$.

7.4. THEOREM: *Suppose* $\bar{\sigma}$, τ, ρ *are pure*, $\mathrm{lev}(\nu) = 0$.

(i) *Whenever* $e \in C^{\mathfrak{F}}$ *and* $(e; \mathfrak{F})$ *is normal and* $\mathrm{Ty}(e) = (\bar{\sigma}; \rho \to \nu)$ *then we can find* e' *such that*

$$\forall \beta \in M^{\rho}[\langle e'\rangle(\mathfrak{F}, \beta)\downarrow \text{ and } \langle e'\rangle(\mathfrak{F}, \beta) = [e]^{\mathfrak{F}}(\beta)].$$

When $\mathrm{Ty}(e) = (\bar{\sigma}; \nu)$ *we can find* e' *such that*

$$\langle e'\rangle(\mathfrak{F})\downarrow \text{ and } \langle e'\rangle(\mathfrak{F}) = [e]^{\mathfrak{F}}.$$

(ii) *For each* e *we can find* e' *such that for all* $\mathfrak{A}$,

$$\langle e\rangle(\mathfrak{A})\downarrow \text{ implies } e' \in C^{\mathfrak{A}} \text{ and } [e']^{\mathfrak{A}} = \langle e\rangle(\mathfrak{A}).$$

Thus R1–R5 can serve to replace I–V for recursion restricted to pure types. The proof of 7.4 is direct from 7.1–7.3.

The schemata R1–R5 are of a character similar to Kleene's S1–S9. In the case that $p = 0$ they are equivalent to them by 7.4 and the results of §4. Of course the equivalence can be established directly by the same methods as §4.

8. Directions of further work

We start out with a definition and some likely steps which may be taken and then become progressively more speculative.

(a) *Ordinals, selection theorem, hierarchies.* If the generating rules I–V are used for a development *ab initio* of recursion in higher type objects over N, then the normalization theorem seems essential for obtaining the selection theorems of Gandy, Moschovakis, and Platek (refs. below).

8.1. DEFINITION: For $e \in C^{\mathfrak{F}}$, let $|e|^{\mathfrak{F}}$ be the ordinal length of the well-founded relation $<$ on $\{(z; \mathfrak{A}) : (z; \mathfrak{A}) < (e; \mathfrak{F})\}$. Let κ be the supremum of all such $|e|^{\mathfrak{F}} + 1$.

It can be shown that $|e|^{\mathfrak{F}}$ is also the ordinal of first appearance of $[e]^{\mathfrak{F}} \leq_e \mathfrak{F}$ in the inductive generation of the $\leq$ relation. Thence

(i) $|e|^{\mathfrak{F}} = 1$ when $\mathrm{Id}(e_0)$

(ii) $|e_1 e_2|^{\mathfrak{F}} = \max(|e_1|^{\mathfrak{F}}, |e_2|^{\mathfrak{F}}) + 1$

(iii) $|\hat{e}|^{\mathfrak{F}} = \sup_{\alpha \in M_\rho}(|e|^{\mathfrak{F},\alpha} + 1)$ for $\mathrm{Ty}(\hat{e}) = (\bar{\sigma}; \rho \to \tau)$

(iv) $|\bar{e}| = \max(|e|^{\mathfrak{F}}, |[e]^{\mathfrak{F}}|^{\mathfrak{F}}) + 1$

(v) $|p_k(e_1, \ldots, e_{n_k})|^{\mathfrak{F}} = \max_{1 \leq i \leq n_k}(|e_i|^{\mathfrak{F}} + 1)$.

One defines $|e|^{\mathfrak{F}} = \kappa$ when $e \notin C^{\mathfrak{F}}$.

For economy, one will work with normal $(e; \mathfrak{F})$. Then if $\mathfrak{F}$ consists of functions F_i where

(1) $$m + 1 \leq \operatorname{lev}(F_i) \leq m + 2$$

$\mathfrak{F}$ can be kept fixed and one considers ordinals

(2) $$|e|^{\mathfrak{F},\mathfrak{A}} \text{ for } \operatorname{lev}(\mathfrak{A}) \leq m.$$

Now let $p = 0$. It should be possible to define a functional ϕ, partial recursive in $\mathfrak{F}$, ${}^{m+1}E$, giving ordinal comparison relationships for $|z|^{\mathfrak{F},\mathfrak{A}}$ and $|w|^{\mathfrak{F},\mathfrak{B}}$ analogous to [9], Theorem 6 ($m = 1$). The prim. rec. function N of 6.12 can be used to reduce to the case of normal pairs (when $z \in C^{\mathfrak{F},\mathfrak{A}}$ or $w \in C^{\mathfrak{F},\mathfrak{B}}$). The general methods of Grilliot [3] should then apply for the selection of type 0 objects.[1]

It should also be possible to generalize Shoenfield's treatment [13] of hierarchies recursive in $\mathfrak{F}$, ${}^2E$ where $m = 0$, and that of Moschovakis in [9], §3 for $\mathfrak{F}$, ${}^3E$ where $m = 1$; cf. also [15].

(b) *Recursion theory over arbitrary structures.* A 1st order structure over ground domains $M_0 = N$, $M_1, \ldots, M_p$ is given by any $\mathfrak{F}$ of level ≤ 1. (Equality on M_i is not assumed; it can be supplied by placing its characteristic function in $\mathfrak{F}$.) More generally, any $\mathfrak{F}$ may be thought of as specifying a higher type structure over these domains.

One thing to do is compare the present I–V with other proposals for recursion in $\mathfrak{F}$ over arbitrary domains, particularly those of Moschovakis [10] for $\mathfrak{F}$ of type level ≤ 2 and Platek [11] for arbitrary $\mathfrak{F}$.

In the first case one will compare recursion over N, M in the present sense with Moschovakis' over the domain M^* = the closure of $M \cup \{0\}$ under pairing. The schemata R1–R5 are of the same character as those for Moschovakis' notion of *prime computable.* I expect these give equivalent theories under a suitable match up of the type structures. In addition, being *recursive in* ${}^2E_M$ should be equivalent to being *hyperprojective* in Moschovakis' sense.[2]

It is also not unreasonable to conjecture that the recursion theory

[1] According to MacQueen [8] there is an essential error in the proof of [3] for selection at higher types; he and Harrington give a correct proof there as well as more abstractly in [4].

[2] By this I mean the restricted sense without scheme C9, i.e., what is elsewhere called *prime computable in* E_{M^*}. It is likely, as the referee of this paper has suggested, that the closely related notion of *prime recursion in a list* $\mathfrak{F}$ used by Fenstad and Moldestad in their recent development of abstract recursion theory is equivalent to being recursive in $\mathfrak{F}$ in the present sense, at least when $\mathfrak{F}$ is total (and of type level ≤ 2). It is possible that use of R1–R5 could then simplify their exposition.

using I–V over any M with N is equivalent to Platek's theory for M within which N is suitably represented.

The next thing to do would be to consider what rules might be reasonably added for stronger recursion theories. One such has been given by Moschovakis [10], namely the schema C9 for a *search operator.* However, this differs from S1–S9, I–V, R1–R5, all of which have a *deterministic* character in the sense that they are uniquely invertible. The following is a deterministic rule which should be adjoined to I–V, but seems not as strong as the search schema. It is based on the idea that if a relation is recursive in $\mathfrak{F}$, and the relation is the graph of a function (or functional) then the function is recursive in $\mathfrak{F}$. Since the unicity condition on a relation to be a function (from M_ρ) into $M_0(i)$ requires the identity relation on $M_0(i)$, it is appropriate only if that relation is recursive in given $\mathfrak{F}$.

VI. Unique Selection

$$\frac{a \in M^{(i)}, 0 \leq_e \mathfrak{F}, a \text{ and } \forall\, b \in M^{(i)}(b \neq a \Rightarrow 1 \leq_e \mathfrak{F}, b)}{a \leq_{e'} \mathfrak{F}}\; e' = (b, e, i).$$

The following is then derivable from VI and abstraction.
(*Functionals from graphs*) For G of type $(\rho \rightarrow 0^{(i)})$,

$$\frac{\forall\, \alpha \in M_\rho[0 \leq_e \mathfrak{F}, \alpha, G(\alpha) \text{ and } \forall\, b \in M^{(i)}(b \neq G(\alpha) \Rightarrow 1 \leq_e \mathfrak{F}, \alpha, b)]}{G \leq \mathfrak{F}}.$$

VI is dispensable in the case that N is the only ground domain, since there we can take

$$a = \mu x \cdot ([e]^{\mathfrak{F}, x} = 0).$$

which is recaptured by use of the recursion theorem. This makes special use of the particular structure of the natural numbers.

All the work of §§3, 6, 7 on derived rules and normalization can be extended directly to include rule VI. Thus a recursion theory based on I–VI should also have good properties. Again, one would want to compare this with other proposals.

(c) *Significance of the schemata.* One would hope that there is a clear informal notion of relative recursiveness for which the kind of schemata here considered are correct and complete, analogous to the situation with the notion of mechanically computable function and the familiar schemata for recursive functions. This is the main gap in the foundations of the subject of recursion theory of finite type objects. It is possible this gap cannot be filled, in particular as suggested by Kreisel [7], pp. 175ff (cf. also back references) there may be something incoherent about assuming the *maximal* type structure and looking for a class of definitions for this structure with a certain constructive character. Nothing done so far deals with such objections.

It should be possible though to isolate some mathematically general features of the schemata considered and to see what rules would be correct and complete for them. One feature, of being *deterministic*, ought to make sense for quite arbitrary structures.

(1) Is there a general notion of a deterministic recursion theory over any domain?

(2) If so, are there finitely many schemata correct and complete for this notion?

A second feature applies particularly to typed structures satisfying very strong forms of the *comprehension schema*, in particular the maximal structure. For these, any *operation* $G: M_\rho \to M_\tau$ which is already defined is thereby recognized to be an *element* $G \in M_{(\rho\to\tau)}$.

(3) Is there a way of subsuming recursion in functionals of finite type under a general recursion theory for arbitrary structures, by building in abstraction as a basic operation?

Since abstraction acts on *syntactic* objects, it would seem for this that one would have to deal with such as members of some new kind of basic domain. Syntactic objects in general are certain kinds of coded well-founded trees. In the end it may be possible to denote these by natural numbers, but that should not be the starting point.

REFERENCES

[1] S. FEFERMAN: *Ordinals and functionals in proof theory*. Proc. Int'l. Cong. of Mathematicians (Nice, 1970) 1, 229–233.

[2] K. GÖDEL: *Über einer bisher noch nicht benützten Erweiterung des finiten Standpunktes. Dialectica 12* (1958) 280–287.

[3] T. J. GRILLIOT: *Selection functions for recursive functionals. Notre Dame J. Formal Logic 10* (1969) 225–234.

[4] L. A. HARRINGTON and D. B. MACQUEEN: Selection in abstract recursion theory. *J. Symbolic Logic 41* (1976) 153–158.

[5] S. C. KLEENE: *Countable functionals*, in *Constructivity in Mathematics*, N-H, Amsterdam (1959) 81–100.

[6] S. C. KLEENE: *Recursive functionals and quantifiers of finite types. Trans. Amer. Math. Soc. 91* (1959) 1–52.

[7] G. KREISEL: *Some reasons for generalizing recursion theory*, in *Logic Colloquium '69*, N-H, Amsterdam (1971) 139–198.

[8] D. B. MACQUEEN: *Post's problem for recursion in higher types*. Dissertation, M.I.T. (1972).

[9] Y. N. MOSCHOVAKIS: *Hyperanalytic predicates. Trans. Amer. Math. Soc. 129* (1967) 249–282.

[10] Y. N. MOSCHOVAKIS: *Abstract first order computability. Trans. Amer. Math. Soc. 138* (1969) I. 427–464, II. 465–504.

[11] R. PLATEK: *Foundations of recursion theory.* Dissertation, Stanford Univ. (1966).

[12] G. SACKS: *The* 1-*section of a type n object*, in *Generalized Recursion Theory* (eds. Fenstad, Hinman), Amsterdam (1974) 81–93.

[13] J. SHOENFIELD: *A hierarchy based on a type* 2 *object. Trans. Amer. Math. Soc. 134* (1968) 103–108.

[14] W. W. TAIT: *Infinitely long terms of transfinite type*, in *Formal Systems and Recursive Functions*, N-H, Amsterdam (1968) 465–475.

[15] S. S. WAINER: *A hierarchy for the* 1-*section of any type two object, J. Symbolic Logic 39* (1974) 88–94.

[16] H. SCHWICHTENBERG and S. S. WAINER: Infinite terms and recursion in higher types. Kiel Proof Theory Symposion, 1974. *Lecture Notes in Mathematics V. 500* (1975) (Springer, Berlin) 341–364.

(Oblatum 27–XI–1974 & 6–III–1975)

Department of Mathematics
Stanford University
Stanford, California 94305
U.S.A.