THE 1-SECTION OF A COUNTABLE FUNCTIONAL

by

Dag Normann  and  Stan S. Wainer
University of Oslo  University of Leeds

PREPRINT SERIES - Matematisk institutt, Universitetet i Oslo
The 1-section of a countable functional

by

Dag Normann and Stan S. Wainer
University of Oslo University of Leeds

The continuous or countable functionals were independently defined by Kleene [9] and Kreisel [10]. They were intended as a suitable basis for constructive mathematics, and thus it is interesting to investigate various notions of recursion on the countable functionals.

There have been two main streams in this investigation, the study of countable recursion and the study of computability or Kleene-recursion.


This paper will mostly be concerned with Kleene-recursion on the countable functionals as defined in Kleene [8] and [9]. We assume some familiarity with the countable functionals and associates, as presented in Kleene [9], Bergstra [1] or any other paper on the subject.

Pioneering work with recursion in nonnormal objects was done by Grilliot [4], who proved that a functional \( F \) of type 2 is normal if and only if its 1-section (that is the set of functions recursive in \( F \)) is closed under ordinary jump, and if and only if \( F \) is continuous on 1-section (\( F \)).

Hinman [6] constructed a countable functional that is not re-
cursively equivalent to a function, and thereby showed that recursion in non-normal functionals is an extension of ordinary recursion in functions.

In [6], Hinman asked if there are functionals with topless 1-sections, i.e. with no maximal elements in the Semi-lattice of degrees. This was answered in the affirmative by Bergstra [1], using a spoiling construction. Thus the class of 1-sections of functionals extends the class of 1-sections of functions.

As a general approach to recursion in functionals of type 2, Wainer [16] constructed hierarchies for the 1-sections of all functionals of type 2. This had also been done by Moschovakis [12] and Hinman [5], but Wainer's hierarchy is based on primitive recursive diagonalization and primitive recursive limit's, so one never needs to ask $^{2E}$ whether or not a given subcomputation $\{e\}^a$ is defined - it always will be.

The hierarchy is modelled on Shoenfield's [15] for normal functionals. With each $F^2$ is associated a set of notations $0^F$, a well-founded ordering $<_F$ on $0^F$ and functions \{ $f_a : a \in 0^F$ \} such that:

- $f$ is recursive in $F$ iff $f$ is primitive recursive in some $f_a$. If $a <_F b$ then $f_a$ is primitive recursive in $f_b$. $0^F$, $<_F$ and \{ $f_a : a \in 0^F$ \} are all $\Pi^1_1(F)$.

This hierarchy generalizes the hierarchy of Shoenfield [15] which works only for normal functionals, but it lacks some of the hierarchical properties of Shoenfields hierarchy such as the uniqueness-property. The uniqueness-property says that if $\|a\|_0^F < \|b\|_0^F$, then $f_a$ is recursive in $f_b$. As we will see (corollary 1.2), this
uniquenes-property cannot hold for the hierarchy for a general non-normal functional.

Let $F$ be a functional. We say that Wainer's hierarchy collapses for $F$ if there is a notation $a$ such that for all $f \in \text{1-section}(F)$ there is a $b$ of norm less than the norm of $a$ such that $f$ is recursive in $f_b$. It is clear that if $F$ is normal, the hierarchy will not collapse, and if $F$ is recursively equivalent to a function, the hierarchy will collapse. Wainer then asked if there are continuous functionals with non-collapsing hierarchies, and Harrington noticed that it will be equivalent to ask for a $\Pi^1_1$ 1-section that is not $\Delta^1_1$.

The first serious approach to a solution of Wainer's problem was made in Bergstra-Wainer [2], and following ideas from that paper, Normann [13] constructed a functional recursive in $0'$ such that its 1-section is $\Pi^1_1$ but not $\Delta^1_1$.

As a basis for the construction, certain functionals $P^b_a$ of Bergstra [1] were used. They are improvements of some non-reducible functionals of Hyland. One of our main results is that this construction is general when we are only concerned with 1-sections. (Theorem 3)

A Construction of a 1-section

Let $T$ be Kleene's $T$-predicate with the following properties:

Each r.e. set is on the form $W_a = \{p: \exists q \ T(a,p,q)\}$.

For any $p,a$ there is at most one $q$ such that $T(a,p,q)$, and if $T(a,p,q)$ holds, then $q \geq 1$.

A computation-function $a$ for $W_a$ is a function which satisfies the following:
\[ p \in W_a \iff \exists q < a(p)T(a,p,q). \]

\( W_a \) will be uniformly primitive recursive in any computation-function for \( W_a \).

**Definition** (Bergstra [1])

\( a \), Let \( \sigma \) be a sequence number.

\[ R_\alpha(\sigma) \iff \exists p,q(1 \leq p,q \leq \text{lh}(\sigma) \land T(a,p,q) \land \sigma(p) < q) \]

\( b \)

\[ F^b_\alpha(a) = \begin{cases} \{ \tau \mid & T(b,\alpha(0),\tau) \land \neg R_\alpha(\tau) \} & \text{if such } \tau \text{ exists} \\ 0 & \text{otherwise} \end{cases} \]

**Remarks.** \( R_\alpha(\sigma) \) says that \( \sigma \) is not the beginning of a computation-function for \( W_a \) and that this can be checked by regarding \( W_a^{\text{lh}(\sigma)} \).

The idea of \( F^b_\alpha \) is that if \( \alpha \) is a computation-function for \( W_a \), then \( F^b_\alpha(\alpha) \) gives the value at \( \alpha(0) \) for a computation-function for \( W_b \). If \( \alpha \) is not a computation-function for \( W_a \), \( \tilde{\alpha}(t) \) is in \( R_\alpha \) for some \( t \), and then \( F^b_\alpha(\alpha) \) gives the computation-function for \( W_b^t \).

From the considerations above, the following lemma is easy to prove.

**Lemma 1** (Bergstra [1])

\( a \) \( W_{\alpha,n} \left[ R_\alpha(\tilde{\alpha}(n)) \Rightarrow R_\alpha(\tilde{\alpha}(n+1)) \right] . \)

\( b \) If \( W_a \) is not primitive recursive in \( \alpha \), then \( \exists n R_\alpha(\tilde{\alpha}(n)) \).

\( c \) There exists \( \alpha \) uniformly recursive in \( W_a \) such that \( \forall n \neg R_\alpha(\tilde{\alpha}(n)) \).

\( d \) \( W_b \) is uniformly recursive in \( W_a, F^b_\alpha, \alpha \) and \( b \).

\( e \) \( F^b_\alpha \) is uniformly recursive in \( W_b, \alpha \) and thus continuous.
Let $H^b_a$ be the partial recursive function defined by the following algorithm:

To compute $H^b_a$ find the least $t_0$ such that $R^a_a(\tilde{a}(t_0))$, and then, if there is a $t < t_0$ such that $T(b,a(0),t)$, let $H^b_a(a)$ be the one such $t$, otherwise let $H^b_a(a) = 0$. $H^b_a$ is a sub-function of $F^b_a$ defined at all $a$ in which $W^a_a$ is not primitive recursive.

Joining various $F^b_a$'s together, we may construct functionals with interesting 1-sections. This method was used in Bergstra [1], Bergstra-Wainer [2] and Normann [13].

We will now use it to prove the following.

**Theorem 1**

Let $A \subseteq N$ be a $\Pi^1_1$-set. Then there is a recursive 1-1-function $\rho$ and a functional $F$ of type 2 recursive in $0'$ such that for all $n$

$$W_{\rho(n)} \in 1\text{-section (}F\text{)} \iff n \in A$$

**Proof.** Let $B \subseteq N \times N$ be semi-recursive such that for no $n$ $B_n$ is recursive in $B_{-n}$, where

$B_n = \{m;(n,m) \in B\}$

$B_{-n} = \{(k,m) \in B;k \neq n\}$

$B$ is constructed by a standard priority construction.

Since $A$ is $\Pi^1_1$, there is a recursive family $<C_n>_{n \in N}$ of linear orderings such that

$$n \in A \iff C_n \text{ is a well-ordering}$$
If \( k \) field \((C_n)\), let

\[
X^n_k = \{ (<n,e>,a); e < C_n k \text{ and } (<n,e>,a) \in B \}
\]

\[
Y^n_k = \{ (<n,e>,a); e < C_n k \text{ and } (<n,e>,a) \in B \}
\]

\[
Z^n = \{ (<n,e>,a); e \text{ field } (C_n) \text{ and } (<n,e>,a) \in B \}
\]

Let \( \tau_1, \tau_2 \) and \( \rho \) be recursive such that

\[
X^n_k = W_{\tau_1(n,k)} \quad Y^n_k = W_{\tau_2(n,k)} \quad Z^n = W_{\rho(n)}
\]

Let \( F(n,k,a) = F_{\tau_2(n,k)}(a) \) whenever \( k \in \text{field } (C_n) \)

\[ F(n,k,a) = 0 \text{ otherwise.} \]

It follows by lemma 1 and the recursion theorem that if \( C_n \) is a well-ordering all \( X^n_k \) such that \( k \in \text{field } (C_n) \) will be uniformly recursive in \( F \), and thus \( Z^n = W_{\rho(n)} \in 1\text{-section } (F) \).

So now assume that \( C_n \) is not a well-ordering, and let \( \{k_i\}_{i\in \mathbb{N}} \) be a \( C_n \)-decreasing sequence. By induction on the computations in \( F \), we show that they are all computations in the partial functionals

\[
K_i(m,k,a) = \begin{cases} 
F(m,k,a) & \text{if } m \neq n \text{ or } m = n \text{ and } \forall(k_i < k) \\
H(m,k,a) & \text{if } m = n \text{ and } k_i \leq k 
\end{cases}
\]

where \( H(n,k,a) = H_{\tau_2(m,k)}(a) \).

\( K_i \) is recursive in \( B_{<n,k_i>} \), so \( B_{<n,k_i>} \) is not recursive in \( K_i \).

The only non-trivial part of the induction is when \( S8 \) is used,

\[
\{e\}^F(\hat{\alpha}) = F(\lambda x[e_1]^F(x,\hat{\alpha})).
\]

Let \( \alpha = \lambda x[e_1]^F(x,\hat{\alpha}) \).
By the induction hypothesis
\[ a = \lambda x(e_1^{K_{i+1}(x, \mathbf{a})}) \]
so \( a \) is recursive in \( B_{<n,k_{i+1}} \). It follows that \( B_{<n,k_{i+1}} \)
is not recursive in \( a \), and for \( k_{i+1} < C_n \) we have that \( W_{\tau_1}(n,k) \)is not recursive in \( a \). Then \( K_i(a) \) is defined and equals \( F(a) \), so
\[ \{e\}^F(a) = \{e\}^{K_i(a)}. \]

We may conclude that \( 1 \)-section \( (F) = 1 \)-section \( K_i \), so
\( 1 \)-section \( (F) \) \( \leq \) \( 1 \)-section \( (B_{<n,k_i}) \). Then \( Z^n = W_\rho(n) \) cannotbe an element of \( 1 \)-section \( (F) \). This ends the proof of the theorem.

Remarks.- The same type of argument is carried through in some more details in Normann [13].

In Theorem 5 we see that functionals of higher type with interesting \( 1 \)-sections are easier to find.

**Corollary 1.1** (Normann [13])

There is a functional \( F \) recursive in \( 0' \) such that
\( 1 \)-section \( (F) \) \( \in \) \( \Pi_1^1 \setminus \Delta_1^1 \).

**Proof.** Let \( A \in \Pi_1^1 \setminus \Delta_1^1 \) in Theorem 1.

**Corollary 1.2**

There is a functional \( F \) recursive in \( 0' \) such that for no \( \Pi_1^1 \)-linearily ordered set \( C \) of \( \Delta_2^0 \)-degrees, \( 1 \)-section \( (F) \) isgenerated by \( C \).

**Proof.** There are \( \Delta_1^1 \)-sets \( D_1, D_2 \) and \( D_3 \), mutually disjoint, such that \( \mathbb{N} = D_1 \cup D_2 \cup D_3 \), \( D_2 \) is infinite and all infinite arithmetic sets intersects both \( D_1 \) and \( D_3 \).
Let $A \in \Pi_1 \setminus \Delta_1$ be such that $D_1 \subseteq A$ and $D_3 \cap F$ be as in Theorem 1, and in order to obtain a, assume that $C$ is a linear $\Pi_1^1$-subset of $\Delta_2^0$-de-1-section $(F)$. Now $\{W_p(m); m \in D_1\}$ is $\Delta_1^1$, an-principle, there is a $\alpha \in C$ such that $W_p(m)$ for all $m \in D_1$.

Now $\{m; W_p(m)\}$ is recursive in $\alpha$ is arithmetic, thus infinite and intersects $D_3$. But if $m \in D_3$, $W_p(m) \in 1$-section $(F)$, so $W_p(m)$ cannot be rec-$\alpha \in C \subseteq 1$-sc $(F)$. This gives the contradiction

Remark. Let $G$ be any countable functional: $\exists$ function $\beta$ such that 1-section $(G) = 1$-section natural hierarchy for 1-section $(\beta)$ will be a archy for 1-section $(G)$ with the uniqueness-principle there will be a function $\beta$ in this hierarchy every element of 1-section $G$ is recursive in 1-section $(G) = 1$-section $(\beta)$.

The functional we constructed in corollary collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsing hierarchy, but no hierarchy with the collapsi
reducing defined computations \( \{e\}(\varphi) \) to "countable recursions" \( \{e'\}(a_\varphi) \) on associates of \( \varphi \). There are however, more countable recursions than there are computations - the fan functional is countably recursive but not computable. We will approach the problem more directly by analyzing the computations themselves, assuming that higher-type objects \( \varphi \) as "given".

A constructive object \( x \) of type \( k+1 \) must in some way be determined as the (pointwise) limit of a sequence \( \langle x_n \rangle_{n<\omega} \) of clearly calculable (eg. primitive recursive) approximations. The complexity of \( x \) is then reflected by its modulus function \( M_x \) such that for all \( \varphi \) of type \( k \), \( M_x(\varphi) = \mu m(\forall n>m)(x_n(\varphi) = x_m(\varphi)) \).

Thus we would like to associate with each defined computation \( x = \{e\}(\varphi) \), a sequence \( \langle x_n \rangle \) uniformly primitive recursive in \( x \), such that \( x = \lim x_n \) and the modulus \( M_x \) is computable uniformly in \( x \). This is clearly possible for \( \varphi \) of type \( \leq 1 \), taking \( x_n \) to be the result (if any) after \( n \) steps in the computation. It is also possible for \( \varphi \) of type-2, as was shown in Wainer [17], but a much more detailed analysis of computations is required in this case. A direct result of this analysis is that the 1-section of every non-normal type-2 object is generated by its "r.e." elements, and so each such type-2 object can be viewed as an "r.e. set construction". This line is developed further in the next section. We now generalize [17] to all finite types.

Let \( \varphi = <\varphi_1, \ldots, \varphi_r> \) denote any list of countable objects of types \( \leq k \), encoded as a single type-\( k \) object. With each possible computation \( \{e\}(\varphi) \) associate the sequence \( \lambda n.h(e,\varphi,n) \) of approximations as follows:

If \( n = 0 \) or if \( e \) is not of the correct form for an index, set \( h(e,\varphi,n) = 0 \). Otherwise:
If \( \{e\}(\varphi) \) is defined by an outright computation \( S1, S2, S3 \) or \( S7 \) then for every \( n > 0 \) set \( h(e, \varphi, n) = \{e\}(\varphi) \).

If \( \{e\}(\varphi) \approx \{e_1\}\{e_2\}(\varphi, \varphi) \) by \( S4 \), set
\[
h(e, \varphi, n) = h(e, <h(e_2, \varphi, n-1), \varphi> , n-1).
\]

If \( \{e\}(\varphi, \kappa) \) is defined by primitive recursion \( S5 \) from \( \{e_1\} \) and \( \{e_2\} \), set
\[
h(e, <\varphi, \kappa>, n) = \begin{cases} h(e_1, \varphi, n-1) & \text{if } \kappa = 0 \\ h(e_2, <h(e, <\varphi, \kappa-1>, n-1), \varphi, \kappa>, n-1) & \text{ow.} \end{cases}
\]

If \( \{e\}(\varphi) \approx \{e_1\}(\varphi') \) by \( S6 \), where \( \varphi' \) is a permutation of \( \varphi \), set
\( h(e, \varphi, n) = h(e_1, \varphi', n-1) \).

If \( \{e\}(\varphi) \approx \varphi_i(\lambda \beta. \{e_1\}(\varphi, \beta)) \) by \( S8 \), set
\[
h(e, \varphi, n) = \varphi_i(\lambda \beta. h(e_1, <\varphi, \beta>, n-1)).
\]

Finally, if \( \{e\}(z, \varphi) \approx \{z\}(\varphi) \) by \( S9 \), set
\[
h(e, <z, \varphi>, n) = h(z, \varphi, n-1).
\]

Since \( h \) is defined by simple induction on \( n \), it is total and primitive recursive.

**Theorem 2**

(i) If \( \{e\}(\varphi) \) is defined, then \( \lim_{n} h(e, \varphi, n) = \{e\}(\varphi) \).

(ii) There is a partial recursive functional \( M \) such that if \( \{e\}(\varphi) \) is defined, then
\[
M(e, \varphi) = \mu m(\forall n \geq m)(h(e, \varphi, n) = \{e\}(\varphi)).
\]

**Proof.** We must prove (i) and (ii) together, by induction over computations \( \{e\}(\varphi) \), using the Recursion Theorem to define \( M \).

All cases except \( S8 \) are straightforward. For example suppose
\[ h(e,\varphi, n) = h(e_1, \langle h(e_2, \varphi, n-1), \varphi \rangle, n-1) \]. Inductively, we can compute \( m_2 = M(e_2, \varphi) \) such that for all \( n > m_2 \), \( h(e_2, \varphi, n-1) = \{e_2\}(\varphi) \), and then compute \( m_1 = M(e_1, \{e_2\}(\varphi), \varphi) \) such that for all \( n > m_1 \), \( h(e_1, \langle e_2 \rangle(\varphi), \varphi, n-1) = \{e\}(\varphi) \). Then for all \( n > \max(m_1, m_2) \), \( h(e, \varphi, n) = \{e\}(\varphi) \), and \( M(e, \varphi) = \mu m (\forall n)[m \leq n \leq \max(m_1, m_2) + 1 \rightarrow h(e, \varphi, n) = h(e, \varphi, m)] \).

Now suppose \( \{e\}(\varphi) = \varphi_1(\lambda \beta. \{e_1\}(\varphi, \beta)) \) by S8, to that \( h(e, \varphi, n) = \varphi_1(\lambda \beta. h(e_1, \langle \varphi, \beta \rangle, n-1)) \). Inductively, we can assume that for every \( \beta \) and every \( n > M(e_1, \varphi, \beta) \)

\[ h(e_1, \langle \varphi, \beta \rangle, n-1) = \{e_1\}(\varphi, \beta) \].

Let \( \alpha_\varphi \) be a fixed list of associates for \( \varphi \), and for each \( \beta \) let \( \alpha_\beta \) be any associate. By Kleene [9], since \( \lambda \beta. M(e_1, \varphi, \beta) \) and \( \lambda \beta. h(e_1, \langle \varphi, \beta \rangle, M(e_1, \varphi, \beta)) = \lambda \beta. \{e_1\}(\varphi, \beta) \) are computable in \( \varphi \), they have associates \( A_M \) and \( A_{hoM} \), recursive in \( \alpha_\varphi \). Clearly \( A_M \) and \( A_{hoM} \) can be chosen so that for every \( \alpha_\beta \) there is an \( x \) such that for every finite sequence \( \sigma \) extending \( \bar{\alpha}_\beta(x) \),

\[ A_M(\sigma) = M(e_1, \varphi, \beta) + 1 \]

and \( A_{hoM}(\sigma) = h(e_1, \langle \varphi, \beta \rangle, M(e_1, \varphi, \beta)) + 1 = \{e_1\}(\varphi, \beta) + 1 \).

Thus we can define an associate \( A_{e_1} \) for \( \lambda \beta. \{e_1\}(\varphi, \beta) \) by

\[ A_{e_1}(\sigma) = \begin{cases} A_{hoM}(\sigma) & \text{if } A_M(\sigma) > 0 \\ 0 & \text{otherwise.} \end{cases} \]

The value of \( \{e\}(\varphi) = \varphi_1(\lambda \beta. \{e_1\}(\varphi, \beta)) \) is therefore determined (with respect to \( \alpha_{\varphi_1} \)) by some finite initial segment of \( A_{e_1} \), say \( \bar{A}_{e_1}(m) \). Let \( N = \max\{A_M(\sigma) | \sigma \leq m\} \).

Now for any \( n > N \) and every \( \bar{\alpha}_\beta(x) \leq m \) such that \( A_{e_1}(\bar{\alpha}_\beta(x)) > 0 \) we have \( n \geq A_M(\bar{\alpha}_\beta(x)) = M(e_1, \varphi, \beta) + 1 \), and so
h(e_1, <\phi, \beta>, n-1) = \{e_1\}(\phi, \beta) = \tilde{A}_{e_1}(\tilde{\alpha}_\beta(x)) - 1. Therefore there is an associate of \lambda \beta. h(e_1, <\phi, \beta>, n-1) which extends \tilde{A}_{e_1}(m), and hence

h(e, \phi, n) = \phi_1(\lambda \beta. h(e_1, <\phi, \beta>, n-1)) = \phi_1(\lambda \beta. \{e_1\}(\phi, \beta)) = \{e\}(\phi).

Thus h(e, \phi, n) = \{e\}(\phi) for every n \geq N, and (i) is proved.

For part (ii) we must show how to compute

M(e, \phi) = \mu m(\forall n > m)(h(e, \phi, n) = h(e, \phi, m)).

Clearly it will be sufficient to show how to decide for each n, whether or not the following holds:

\exists m > n(h(e, \phi, m) \neq h(e, \phi, n))

i.e. \exists m > n(\phi_1(\lambda \beta. h(e_1, <\phi, \beta>, m-1)) \neq \phi_1(\lambda \beta. h(e_1, <\phi, \beta>, n-1))).

To do this, first define a functional \gamma_n as follows:

\gamma_n(\beta) = \begin{cases} h(e_1, <\phi, \beta>, m_0-1) & \text{if } m_0 = \mu m(n < m \leq M(e_1, \phi, \beta) \land h(e, \phi, m) \neq h(e, \phi, n)) \\ h(e_1, <\phi, \beta>, M(e_1, \phi, \beta)) & \text{if there is no such } m_0. \end{cases}

Clearly, \gamma_n is uniformly recursive in n, \phi, and it is easy to see that

\gamma_n = \begin{cases} \lambda \beta. h(e_1, <\phi, \beta>, m_0-1) & \text{if } m_0 = \mu m > n(h(e, \phi, m) \neq h(e, \phi, n)) \\ \lambda \beta. \{e_1\}(\phi, \beta) & \text{if } \forall m > n(h(e, \phi, m) = h(e, \phi, n)). \end{cases}

Therefore \exists m > n(h(e, \phi, m) \neq h(e, \phi, n)) if and only if

h(e, \phi, n) \neq \phi_1(\gamma_n).

This completes the proof of (ii).

Corollary 2.1

For every computable type \( k+1 \) functional \( \lambda \phi. \{e\}(\phi) \) over the type structure of countable functionals, there is a computable \( \Sigma^0_1 \) set

\( D^{k+1}_e = \{<\phi, n>|\exists m > n(h(e, \phi, m) \neq h(e, \phi, n))\} \)
such that for every \( \varphi \),
\[
\{e\}(\varphi) = h(e, \varphi, \mu m(<\varphi, m> \in D_{e}^{k+1}))
\]

**Corollary 2.2**

For each countable functional \( \varphi \) of type \( > k \) define
\[
h_{\varphi}^{k}(e, \beta, n) = h(e, <\varphi, \beta>, n)
\]
and
\[
D_{e, \varphi}^{k} = \{<\beta, n> | \exists m > n(h_{\varphi}^{k}(e, \beta, m) * h_{\varphi}^{k}(e, \beta, n))\}.
\]

Then for every \( \lambda \beta, \{e\}(\varphi, \beta) \in k-\text{sc}(\varphi) \) we have
(i) \( D_{e, \varphi}^{k} \in k-\text{sc}(\varphi) \), since \( <\beta, n> \in D_{e, \varphi}^{k} \) if \( n < M(e, \varphi, \beta) \).
(ii) \( \lambda \beta, \{e\}(\varphi, \beta) \) is \( \mu \)-recursive in \( h_{\varphi}^{k} \) and \( D_{e, \varphi}^{k} \).

Hence \( k-\text{sc}(\varphi) \) is generated by its \( \Sigma_{1}^{0}(h_{\varphi}^{k}) \) elements.

**Corollary 2.3**

For every countable functional \( \varphi \) of type \( k+2 \), \( 1-\text{sc}(\varphi) \) is a \( \Pi_{1}^{1}(h_{\varphi}^{1}) \) \( (\Pi_{1}^{1}(h_{\varphi}^{1}) \) if \( k = 0 \)) set of reals generated by its \( \text{r.e.-in-} h_{\varphi}^{1} \) elements.

More precisely, there is a \( \Pi_{1}^{1}(h_{\varphi}^{1}) \) set \( B \) of \( \text{r.e.-in-} h_{\varphi}^{1} \) indices such that
\[
1-\text{sc}(\varphi) = \{f|f \leq T^{h_{\varphi}^{1}}_{i} \text{ for some } i \in B\}.
\]

**Proof.** By Corollary 2.2, we need only to take
\[
B = \{i|\exists e(W_{i}^{h_{\varphi}^{1}} = h_{\varphi}^{1} \cup D_{e, \varphi}^{1} \wedge \lambda x.\{e\}(\varphi, x) \text{ is total})\}.
\]
By results of Bergstra [1] on \( 2-\text{en}(\varphi) \), \( \{e|\lambda x.\{e\}(\varphi, x) \text{ is total}\} \) is \( \Pi_{1}^{1}(k+1) \) in \( h_{\varphi}^{1} \), and therefore so is \( B \).
The Enumeration Operator of a Non-Normal Type-2 Object

Let $F$ be an arbitrary fixed non-normal type-2 object. Since $1\text{-sc}(F)$ is generated entirely "from within", and since $F$ is continuous on its 1-section, all the results of the previous section carry through unchanged for recursions in $F$ (alternatively see Wainer [17]). Thus we have a total primitive-recursive-in-$F$ function

$$h_F(e,a,n) = h(e,<F,a>,n)$$

and a partial-recursive-in-$F$ function

$$M_F(e,a) = M(e,F,a)$$

such that for every $e$ and every list $a = <a_1 \ldots a_r>$ of integers:

$$\{e\}^F(a) \text{ defined } \rightarrow M_F(e,a) \text{ defined and equal to }$$

$$\forall m \forall n \geq m[h_F(e,a,n) = \{e\}^F(a)].$$

Therefore with each $\{e\}^F \in 1\text{-sc}(F)$ is associated a r.e.-in-$h_F$ set

$$D_e,F = \{<a,n> | \exists m > n(h_F(e,a,m) \neq h_F(e,a,n))\}$$

such that

(i) $D_e,F \in 1\text{-sc}(F)$, since $<a,n> \in D_e,F$ iff $n < M_F(e,a)$.

(ii) $\{e\}^F \subseteq_T h_F,D_e,F$.

Henceforth we will usually omit the subscripts $F$.

Now there is a primitive recursive function $\lambda e.e'$ such that $\{e'\}^F(a) = F(\lambda x.\{e\}^F(a,x))$. Thus the action of $F$ can be regarded as a "jump" from the h-r.e. set $D_e$ to the h-r.e. set $D_{e'}$. As was remarked in [17], this begs the question whether it might be possible to replace $F$, in the generation of its 1-section, by a continuous-everywhere Bergstra-type functional $F_{e'}^e(a)$. We now show that this is indeed the case. Thus from the point of view of 1-sections, the Bergstra-type functionals are the only ones.
Definition

(a) For finite sequences $\sigma$ let

\[ \text{Mod}(e, a, \sigma) \leftrightarrow \forall x, j < \text{lh}(\sigma)[j \sigma_x \rightarrow h(e, <a, x>, j) = h(e, <a, x>, \sigma_x)]. \]

(b) Then associate with $F$ the enumeration operator

\[ J_F(a) = \{<e, a, n> | \exists m > n(h(e', a, m) \ast h(e', a, n) \land \text{Mod}(e, a, \tilde{a}(m))) \} \]

Note (cf. Lemma 1)

(i) To compute $J_F(a)(<e, a, n>)$ from $h$ first see if

\[ n \in D_{e', a} = \{n | \exists m > n(h(e', a, m) \ast h(e', a, n)) \}. \]

If so, find the first stage $m$ which witnesses its membership and then for each $a$ give value 1 if $\text{Mod}(e, a, \tilde{a}(m))$. Otherwise give value 0.

Hence $J_F$ is continuous and of Kalmar rank $\leq \omega$.

(ii) $\text{Mod}(e, a, \tilde{a}(m))$ says that up to stage $m$, $a$ looks like a correct modulus function for the sequence $\lambda x. h(e, <a, x>, j)$ approximating $\lambda x. \{e\}(a, x)$. Thus

\[ \forall m \text{ Mod}(e, a, \tilde{a}(m)) \leftrightarrow \forall x(M(e, a, x) \leq a(x)). \]

Therefore if $\forall m \text{ Mod}(e, a, \tilde{a}(m))$ then the set

\[ D_{e, a} = \{<x, n> | \exists m > n(h(e, <a, x>, m) \ast h(e, <a, x>, n)) \} \]

is primitive recursive in $a$.

Conversely if $\lambda x. \{e\}(a, x)$ is total, then from $D_{e, a}$ we can compute $M(e, a, x) = \min(<x, n> \in D_{e, a})$, so that

\[ n \in D_{e', a} \leftrightarrow <e, a, n> \in J_F(\lambda x. M(e, a, x)). \]

Thus $J_F(a)(<e, a, n>) \approx \frac{F_{e', a}(n{\ast}a)}{e, a}$. 

Theorem 3

\[ 1\text{-sc}(F) = 1\text{-sc}(h_F,J_F) \].

Corollary 3.1

For every non-normal type-2 object there is a continuous type-2 object, of Kalmar rank \( \leq \omega \), with the same 1-section.

Theorem 3 is proved by the two following lemmas.

Lemma 2

There is a recursive function \( d \) such that

\[ \{e\}^F(a) \text{ defined} \Rightarrow D_{e,a} = \{d(e,a)\}^h_j \].

Hence if \( \{e\}^F \) is total, then \( \{e\}^F \preceq_T h,D_e \in 1\text{-sc}(h,J) \).

Proof. By induction over computations \( \{e\}^F(a) \), using the Recursion Theorem to define \( d \).

For example suppose \( \{e\}^F(a) \cong \{e_1\}^F(\{e_2\}^F(a),a) \) by S4.

Inductively we can use \( d(e_2,a) \) to compute \( k_2 = \mu k \in D_{e_2,a} \), and then \( k_1 = \mu k \in D_{e_1,u,a} \) where \( u = h(e_2,a,k) = \{e_2\}^F(a) \). Then

\[ D_{e,a} = \{n|\exists m>n(h(e,a,m) \ast h(e,a,n))\} \text{ where, for every } m > \max(k_1,k_2), \]

\[ h(e,a,m) = h(e_1,(h(e_2,a,m-1),a\rangle,m-1) = h(e_1,<u,a>,m-1) = \{e\}^F(a) \].

So \( D_{e,a} = \{n|\exists m>n \leq \max(k_1,k_2) \wedge h(e,a,n) \ast h(e,a,m))\} \), and its index \( d(e,a) \) is clearly given by a primitive recursive function of \( d(e_2,a) \) and \( d(e_1,<u,a>) \).

If \( \{e'\}^F(a) \cong F(\lambda x.\{e\}^F(a,x)) \) by S8, then inductively we have, for each \( x \), \( D_{e,a,x} = \{d(e,<a,x>)\}^h_j \). Therefore

\[ D_{e,a} = \{<x,n>\mid n \in D_{e,a,x}\} \text{ is recursive in } h,J \text{ with index given by a primitive recursive function of } e,a, \text{ and an index for } d \]. Note (ii) above then shows how to obtain the required \( J \)-index for \( D_{e',a} \).
Lemma 3

There is a partial recursive $\psi$ such that if $F$ is continuous on $1$-$sc(a,h)$, then for all $e,a,n$,

$$J_F(a)(<e,a,n>) = \psi(F,a,e,a,n).$$

Hence $1$-$sc(h,J) \subseteq 1$-$sc(F)$.

Proof. The parameters $a$ will be deleted from the following argument, since they remain inactive throughout. Recalling the definition of $J_F(a)$ and of $h(e',m)$, we simply have to decide (recursively in $F$) the following:

$$\exists m>n(F(\lambda x.h(e,x,m-1)) \neq F(\lambda x.h(e,x,n-1)) \land \text{Mod}(e,\bar{a}(m))).$$

The procedure is a refinement of that used for part (ii) of Theorem 2.

Define $g_n$ recursively in $h,a,e,n$ as follows: Given $x$, look for the least $m_0$ such that $n < m_0 \leq a(x) \land h(e',n) \neq h(e',m_0) \land \text{Mod}(e,\bar{a}(m_0))$. If such an $m_0$ is found, set $g_n(x) = h(e,x,m_0-1)$. If no such $m_0$ is found, set $g_n(x) = h(e,x,a(x))$.

Suppose $\exists m>n(h(e',n) \neq h(e',m) \land \text{Mod}(e,\bar{a}(m)))$ and let $m_0$ be the least such. Clearly if $m_0 \leq a(x)$ then $g_n(x) = h(e,x,m_0-1)$. If $a(x) < m_0$ then since $\text{Mod}(e,\bar{a}(m_0))$ holds, we have (putting $j = m_0-1$)

$$g_n(x) = h(e,x,a(x)) = h(e,x,m_0-1).$$

Therefore

$$g_n = \begin{cases} 
\lambda x.h(e,x,m_0-1) & \text{if } <e,n> \notin J_F(a) \\
\lambda x.h(e,x,a(x)) & \text{if } <e,n> \in J_F(a). 
\end{cases}$$

With the aid of $g_n$, we can now compute $J_F(a)(<e,n>)$ recursively in $F$ as follows:
(1) First see if \( h(e',n) = F(\lambda x. h(e,x,a(x))) \). If so, then
\[ <e,n> \in J_F(\alpha) \iff h(e',n) \neq J(g_n). \]
because if \( <e,n> \in J_F(\alpha) \) then the \( m_0 \) above exists and
\[ F(g_n) = F(\lambda x. h(e,x,m_0-1)) = h(e',m_0). \]

(2) Now suppose \( h(e',n) \neq F(\lambda x. h(e,x,a(x))) \). Define
\[ \beta_m(x) = \begin{cases} h(e,x,a(x)) & \text{if } x,a(x) < m \\ h(e,x,m-1) & \text{otherwise.} \end{cases} \]
Then \( \lambda x. \beta_m(x) \in 1-sc(\alpha,h) \) and \( \lambda x. h(e,x,a(x)) = \lim \beta_m \).
Therefore \( F(\lambda x. h(e,x,a(x))) = \lim F(\beta_m) \) since \( F \) is continuous on \( 1-sc(\alpha,h) \), so we can compute
\[ m_1 = \mu m > n(h(e',n) \neq F(\beta_m)). \]
Now if \( <e,n> \in J_F(\alpha) \) let
\[ m_0 = \mu m > n(h(e',n) \neq h(e',m) \land \text{Mod}(e,\bar{a}(m))). \]
Since \( \text{Mod}(e,\bar{a}(m_0)) \) holds, we have for every \( j \leq m \),
\[ \beta_j = \lambda x. h(e,x,j-1) \]
and hence \( F(\beta_j) = h(e',j) \).
Therefore \( m_0 = m_1 \) and so \( \text{Mod}(e,\bar{a}(m_1)) \) holds.
Conversely if \( \text{Mod}(e,\bar{a}(m_1)) \) holds, then for every \( j \leq m \),
\( F(\beta_j) = h(e',j) \) and so \( m_1 \) is the first witness to the fact that
\( <e,n> \in J_F(\alpha) \). Hence
\[ <e,n> \in J_F(\alpha) \iff \text{Mod}(e,\bar{a}(m_1)). \]
This completes case (2) and the proof of Lemma 3.

Corollary 3.2.

If \( F \) is everywhere-continuous, then \( J_F \) is recursive in \( F \).
A natural question to ask of \( J_F \) is whether or not it is a uniform enumeration operator (i.e. whether or not there is a recursive function \( j(e) \) such that if \( a_1 \) is recursive in \( a_2 \) with index \( e \) then \( J_F(a_1) \) is recursive in \( J_F(a_2) \) with index \( j(e) \)). If this were the case, then one would hope to be able to give a notation-free degree-theoretic hierarchy for \( 1\text{-}sc(F) \).

**Theorem 4**

If \( 1\text{-}sc(F) \) is topless, then \( J_F \) is not uniform.

**Proof.** If \( 1\text{-}sc(F) \) is topless, then \( h \cup J_F(h) \prec_T h' \), for otherwise \( 1\text{-}sc(F) = 1\text{-}sc(h') \). But then by Theorem 3 of Lachlan [11], \( J_F(h \cup J_F(h)) \prec_T h \cup J_F(h) \). So if \( J_F \) were uniform, a straightforward induction on computations would give \( 1\text{-}sc(F) = 1\text{-}sc(h \cup J_F(h)) \), again contradicting the fact that \( 1\text{-}sc(F) \) is topless.

**Remark**

Theorem 3 shows that every non-normal 1-section is an ideal in the degrees, generated from a real \( h \) by iterating a certain "r.e. set construction" \( J \) along (simultaneously generated) well-orderings.

By an "r.e. set construction" we mean a procedure of the following kind:

An arbitrary \( \Delta^0_2 \)-set \( A (\prec_T 0') \) is presented in the form of a pair \((e,a)\) where \( e \) is the index of a primitive recursive sequence \( \lambda s.x.[e](x,s) \) such that

\[
A(x) = i \iff \forall s \leq a(x) ([e](x,s) = i) \quad i = 0 \text{ or } 1.
\]

Then a (recursive) sequence \( \lambda m.h(e,a,m) \) is defined so that for each \( a \), \( h(e,a,m) = 0 \) unless at some stage \( m \) it is decided, on the basis of the finite set \( \{[e](x,s) | x,s < m\} \) of approximations
to A "available" at stage $m$, to enumerate $a$ into the r.e. set being constructed, in which case $h(e,a,m) = 1$. For the decision to have been a vital one, it is therefore only necessary that, up to $m$, $[e]$ looks like a correct sequence for $A$. This can be expressed by the relation

$$\text{mod}(e,\bar{a}(m)) \Leftrightarrow \forall x, s < m(s > a(x) \Rightarrow [e](x,s) = [e](x,a(x))).$$

Thus the construction can be regarded as an enumeration operator $J$ of the following familiar sort:

$$J(a) = \{<e,a,n> | \exists m(h(e,a,m) * h(e,a,n) * \text{mod}(e,\bar{a}(m)))\},$$

and the set constructed from $A$ as above is then

$$A_1 = \{a | \exists m(h(e,a,m) = 1)\} = \{a | <e,a,0> \in J(a)\}.$$ 

From $e$ and $J(a)$ we can then compute a presentation $(e_1, a_1)$ of $A_1$, and repeat the construction in order to obtain a new r.e. set

$$A_2 = \{a | <e_1,a,0> \in J(a_1)\},$$

and so on.

It therefore makes perfectly good sense to talk about the 1-section of an r.e. set construction, i.e. $1\text{-sc}(h,J)$, and it is to be hoped that many interesting 1-sections will be generated directly by appropriate combinations of priority constructions.

**The 1-section of a type $k+2$ functional ($k \geq 1$)**

In the previous section we described a standard procedure for creating 1-sections of countable type-2 functionals. In this section we will give a general 1-section construction for higher type functionals. We will show that the necessary conditions for a class of sets to be a 1-section given in corollary 2.3 will in fact
be sufficient.

In Normann [14], a method of constructing higher type functionals with interesting 1-sections were developed. We will show that all 1-sections of functionals of type >2 may be obtained using that method. We need the following lemma:

**Lemma 4** (Normann [14])

There is a primitive recursive list \( \phi_n \) of primitive recursive functionals of type \( k \) such that:

For all \( \Pi_k^1 \)-sets \( B \subseteq \mathbb{N} \) there is a recursive relation \( R \) such that

\[
m \in B \iff \exists \psi \in \mathcal{C}(k+1) \forall n R(m, \psi(\phi_0), \ldots, \psi(\phi_{n-1})).
\]

Moreover, if \( m \in B \), we may choose \( \psi \) recursive uniformly in \( m \).

Now, given any \( \Pi_k^1 \)-set \( B \) of indices, we define

\[
\phi(e, k, \psi) = \begin{cases} 1 & \text{if } \exists s(T(e, k, s) \land \forall n \leq s R(e, \psi(\phi_0), \ldots, \psi(\phi_{n-1}))) \\ 0 & \text{otherwise} \end{cases}
\]

If \( e \in B \) and \( \forall n R(e, \psi(\phi_0), \ldots, \psi(\phi_{n-1})) \) we see that \( \lambda k \phi(e, k, \psi) \) is the characteristic function of \( W_e \), so \( \{W_e; e \in B\} \) will be a subset of 1-section (\( \phi \)).

It is also easily seen that \( \phi \) is recursive in \( 0' \).

We prove the unrelativized version of the theorem:

**Theorem 5** \((k \geq 1)\)

Let \( A \subseteq \mathcal{P}(\omega) \) be \( \Pi_k^1 \), closed under recursion in finite lists and recursively generated by it's r.e. elements.

Then there is a continuous functional \( \phi \) of type \( k+2 \) recursive in \( 0' \) such that

\[
A = 1\text{-section } (\phi).
\]
Proof. We prove the theorem in detail for \( k = 1 \). With some modifications, the same argument works for the general case.

Let \( C \in \Delta^1_1 \) be a subset of \( \omega \) such that both \( C \) and its complement intersects all infinite arithmetic sets.

Let \( B_0 = \{ e; W_e \in A \} \), \( B = B_0 \cap C \). \( B \) will be \( \Pi^1_1 \).

If \( W \) is an r.e. set, \( \{ e; W = W_e \} \) is an infinite arithmetic set, so each r.e. set in \( A \) has an index in \( C \), and thus in \( B \). On the other hand all arithmetic subsets of \( B \) are finite. So, if we construct \( \phi \) as above, the first property gives us that

\[
A \subseteq 1\text{-section}(\phi).
\]

We will see that the other property gives us equality.

Definition.

Let \( \alpha_{e,k} \) be the canonical associate for \( \lambda \phi \lambda (e,k,\phi) \).

Let \( \alpha = \langle \alpha_{e,k} \rangle, e,k \in \omega \).

Remark

\( \alpha_{e,k} \) will be uniformly recursive in \( W_e \) and \( k \). Moreover, if \( e_1 \) is an index for a Kleene-computation, \( \hat{f} \) a list of functions and \( \{e_1\}(\phi,\hat{f}) \), the value of this computation is uniformly recursive in \( \alpha, \hat{f} \).

Now assume that \( \lambda x\{e_1\}(\phi,x) \) is a total function. We call a subcomputation \( \{e_2\}(\phi,\hat{f}) \) of \( \{e_1\}(\phi,x) \) essential if the list \( \hat{f} \) is from \( \omega \cup \{\phi_n; n \in \omega \} \).

Claim. The set of essential subcomputations of \( \lambda x\{e_1\}(\phi,x) \) is arithmetic with an arithmetic enumeration.

Proof. The essential subcomputations may be defined by a \( \Sigma^0_1(\alpha) \)-positive inductive definition. We give two of the clauses:
iv If \( \{e_2\}(\phi,\vec{e},\vec{f}) = \{e_3\}(\phi,\{e_4\}(\phi,\vec{e},\vec{f}),\vec{f}) \) is an essential sub-computation, then \( \{e_4\}(\phi,\vec{e}) = k \) is an essential sub-computation, and \( \{e_3\}(\phi,k,\vec{f}) \) is an essential sub-computation. \( k \) is found using \( a \).

viii If \( \{e_2\}(\phi,e,k,\vec{f}) = \phi(e,k,\lambda g\{e_3\}(\phi,g,\vec{f})) \) is an essential sub-computation, then \( \{e_3\}(\phi,\varphi_n,\vec{f}) \) are essential sub-computations for all \( n \).

Starting with \( \{e_1\}(\phi,x) \), we will then get to all essential sub-computations, so this is a \( \Sigma_1^0(\alpha) \)-class and by the effective enumeration of the \( \varphi_n \)'s, it is arithmetically enumerable.

We say that \( \phi \) is used non-effectively at \( e \) if there is an essential sub-computation
\[
\{e_2\}(\phi,e,k,\vec{f}) = \phi(e,k,\lambda g\{e_3\}(\phi,g,\vec{f}))
\]
such that
\[
\forall n R(e,<\{e_3\}(\phi,\varphi_0,\vec{f}),\ldots,\{e_3\}(\phi,\varphi_{n-1},\vec{f})>).
\]

If \( \phi \) is used non-effectively at \( e \), then \( e \) must be in \( B \), since \( \forall = \lambda g\{e_3\}(\phi,g,\vec{f}) \) is total and for \( e \in B \) there is an \( n \) such that
\[
\forall R(e,<\forall(\varphi_0),\ldots,\forall(\varphi_{n-1})>).
\]

Moreover \( D = \{e; \phi \text{ is used non-effectively at } e \} \) is arithmetic, so \( D \) is a finite subset of \( B \).

Let \( a_D = \langle a_e, k \rangle e \in D \).

\( a_D \) will be recursive in \( \langle \forall e \rangle e \in D \), so \( a_D \) is recursive in some element in \( A \).

By induction on the length of the essential computations we
prove that they will be uniformly recursive in $\alpha_D$. The only non-trivial case is

$$\{e_2\}(\phi,e,k,f) = \phi(e,k,\lambda g\{e_3\}(\phi,\Lambda_f))$$

where we split the instruction in two cases:

**i** $e \in D$ Then $k \in W_e \leftrightarrow e_k(<>) = 1$. If $k \in W_e$, we give out value 0. If $k \in W_e$, find recursively the $s$ such that $T(e,k,s)$. If

$$\forall n \leq s \in R(e,<\{e_3\}(\phi,\Lambda_{0_f}),\ldots,\{e_3\}(\phi,\Lambda_{n-1_f})))$$

we give out value 1, otherwise we give value 0.

By the induction hypothesis we may recursively in $\alpha_D$ decide the statement above.

**ii** $e \notin D$ Then for some least $n$

$$\exists R(e,<\{e_3\}(\phi,\Lambda_{0_f}),\ldots,\{e_3\}(\phi,\Lambda_{n-1_f})))$$

which we find recursively in $\alpha_D$.

Then check if for some $s < n$, $T(e,k,s)$. If there is one, give out value 1, otherwise give out value 0.

It follows that $\lambda x\{e\}(\phi,x)$ is recursive in $\alpha_D$, and thus recursive in some element of $A$. This means that 1-section $(\phi) \subseteq A$, and the theorem is established for $k = 1$.

For $k > 1$, we let $C$ be $\Sigma^1_k$ such that both $C$ and the complement of $C$ intersect all infinite $\Sigma^1_{k-1}$-sets. We construct $\phi$ in the analogue way of the construction above. Again $A \subseteq 1$-section $(\phi)$ is trivial. Assume that $\lambda x\{e\}(\phi,x)$ is total. We let the essential subcomputations be all subcomputations where arguments of type-$k$ are from the list $\{\omega_n\}_{n \in \omega}$. Replacing functionals of type-$k$ by arbitrary associates for them, we see that
the set of essential subcomputations will be \( \Sigma_{k-1}^1 \). Then \( \{ e; \phi \text{ is used non-effectively at } e \} \) will be a \( \Sigma_{k-1}^1 \)-subset of \( B \), and thus finite. The rest of the argument goes exactly as above, replacing functionals of type \( \leq k \) by arbitrary associates for them.

**Corollary 5.1**

Every 1-section of a countable functional is the 1-section of a 1-obtainable functional of the same type.

**Remark.** Corollary 5.1 is an analogue of corollary 3.1. The notion of Kalmar-Rank for higher-type functionals is meaningless, but for our constructed functional \( \phi \), when \( e \) and \( k \) are given, there will be a finite, fixed list of functionals \( \phi_0, \ldots, \phi_{n-1} \) such that \( D(e,k,\psi) \) is decided by \( \psi(\phi_0), \ldots, \psi(\phi_{n-1}) \). So these functionals really have well-defined rank \( \omega \).

**Footnote**

1) While preparing this paper, the second author was partly supported by a grant from the Norwegian Research Council for Science and the Humanities (NAVF).

**References**


5. P. Hinman, Hierarchies of Effective Descriptive Set Theory, T.A.M.S. 142 (1969) 111-140.


