A transfinite hierarchy of reals*

George Barmpalias
School of Mathematics
University of Leeds
Leeds LS2 9JT, U.K.
georgeb@amsta.leeds.ac.uk

July 26, 2002

Abstract

We extend the hierarchy defined in [5] to cover all hyperarithmetical reals. An intuitive idea is used for the definition, but a characterization of the related classes is obtained. A hierarchy theorem and two fixed point theorems (concerning computations related to the hierarchy) are presented.

1 Introduction

A real number x can be represented by its binary expansion, i.e. a set A such that $n \in A \iff$ the n-th binary digit of x is 1. In this case we write $x = x[A]$. Thus the classical hierarchies of computability theory (e.g. the arithmetical, the hyperarithmetical hierarchy) can be seen as hierarchies of reals. However, the classes of reals we get in this way are not natural from the point of view of computable analysis [1]; for, important classes like c.e. or co-c.e. reals (that is, left aproximable and right aproximable reals, see [5]) are not classes of these hierarchies. Weihrauch and Zheng [5] defined a natural hierarchy of length ω which is closely related to the (recursion theoritic) arithmetical one *and* classes as the c.e. and co-c.e. reals are classes of the hierarchy[2]. Moreover the definition reflects the difficulty of aproximating

*To appear in *Mathematical Logic Quarterly* **49** no 3 (2003)

[1] This is only one disadvantage of the binary representation of reals. In general this representation is not acceptable in computable analysis because it gives rise to peculiar situations.

[2] other, lower complexity hierarchies have been studied in Rettinger, Zheng, Gengler, von Braunmhl[2], Weihrauch and Zheng[4] and Zheng, Weihrauch, Ambos-Spies[6].

a real by a sequence of rationals. The purpose of this paper is to extend this hierarchy as far as possible, by using the same idea: reals are classified according to the 'order' of the prefix of $sup - inf$ alternations that is needed in front of a computable object, in order to get them (this statement will become more clear and precise in the following sections). In section 2 we give the intuitive idea behind the definition given in the next section. In section 4 we give a characterization of the reals of a class in the hierarchy which allows us to prove the hierarchy theorem in section 5. In section 4 we also prove the invariance of the hierarchy under the system of notation used in its definition. The relation of our hierarchy to the hyperarithmetical one is given in section 6. Finally a couple of fixed point theorems regarding a special kind of computations with a variable oracle (which arised from the definition of the hierarchy) is given in the last section.

2 The basic idea

We want to extend the hierarchy defined in [5] in terms of finitely many alternations of *sup* and *inf* in front of a computable function $f : \mathbb{N} \to \mathbb{Q}$; and we want to use the same idea. Intuitively, at infinite successor levels (which correspond to infinite successor ordinals) we want to have an infinite prefix (and in particular of the order type of the corresponding ordinal) of *sup-inf* alternations. But of course, a computable function has only finitely many arguments, and so it is impossible to apply this idea directly. However, we can do it indirectly; in terms and notation of [5] we define $\Sigma_\omega = \Pi_\omega = \Delta_\omega$ to be the set of all arithmetical real numbers, i.e. $\cup_{i\in\omega}\Delta_i$. Now a number x in $\Sigma_{\omega+1}$ is one that can be obtained as the supremum of a sequence $\{x_i\}_{i\in\omega}$ whose terms are of the form

$$x_i = \sup_{i_1} \inf_{i_2} \ldots \Theta_{m(i)} f_i(i_1, \ldots, i_{m(i)}) \tag{1}$$

where Θ_i is $\sup_i$ if i is odd and $\inf_i$ otherwise; $\{f_i\}_{i\in\omega}$ is a computable sequence of computable functions with rational values such that f_i has $m(i)$ variables (for natural numbers), m computable (say non-decreasing, unbounded) function.

We can picture the computable functions (with rational image) arranged in a hierarchy of height ω so that at level 0 we have constants and at level $m > 0$ we have the computable functions of m arguments; and we regard a function f to be 'higher' than one g with less arguments because in general,

f prefixed (with the usual $sup - inf$ prefix) can give more complex reals than g.

So, what we did in order to obtain a number in $\Sigma_{\omega+1}$ is to go *effectively* through the whole hierarchy of [5] (by choosing the 'rank' $m(i)$ and the program i of f and prefix it accordingly) and put a sup on the top. Intuitively, we picture x as

$$\sup \ldots \sup \inf F \tag{2}$$

where F is a 'higher-type' computable object, in this case a computable sequence of computable sequences of rationals; and speaking informally, the prefix has order-type $\omega + 1$ (reading it from right to left).

We can define $\Pi_{\omega+1}$ accordingly (by exchanging the occurences of inf and sup in the above). At the next successor levels, we just increase number of arguments of the higher-type object, and add the usual $sup-inf$ prefix for those arguments. In particular, to get a number in $\Pi_{\omega+2}$ we do what we did in the case of $\Sigma_{\omega+1}$ but using a computable double sequence of computable sequences and fixing the second argument; so we get a sequence which goes effectively through $\Sigma_{\omega+1}$ whose infimum (if it exists) is a number in $\Pi_{\omega+2}$ (one can show that if we take the supremum of such a sequence, we will never get out of $\Sigma_{\omega+1}$). We picture such a number x as follows:

$$x = \inf \sup \ldots \sup \inf F$$

where this time F is a 'higher-type' computable object of order $\omega + 2$. We can continue in the way described above, and define even higher classes. What we need if we are on a given level is to be able to select effectively objects of lower levels, i.e. to compute the level and the object of that level we want to select. This requirement restricts us in the initial segment of computable (i.e. recursive) ordinals and also forces us to employ a system of notations for these ordinals; Kleene's $\mathcal{O}$ is a straightforward choice.

In the following, we often use the term 'finite limit':

Definition 1. If we have an expression of the form $x(\boldsymbol{m}) = \lim_{i\to\infty} f(i, \boldsymbol{m})$ where $f : \mathbb{N}^{n+1} \to \mathbb{R}$, $n \geq 0$ and for all $\boldsymbol{m} \in \mathbb{N}^n$ $\lambda i.f(i, \boldsymbol{m})$ is eventually constant, then we say that x is a finite limit of f (or just a finite limit if it is clear which function is f or even that the limits in that expression are finite). We use the analogous expressions when we have sup or inf in place of lim.

In order to make things simpler, we use the following

Proposition 1. *From $n > 0$ and a function $f : \mathbb{N}^n \to \mathbb{Q}$ such that*

$$x := \sup_{i_1} \inf_{i_2} \ldots \Theta_{i_n} f(i_1, \ldots, i_n) \textit{exists},$$

we can go effecticely (in fact primitive recursively) to a fuction $g : \mathbb{N}^n \to \mathbb{Q}$ such that:

$$x = \sup_{i_1} \lim_{i_2} \ldots \lim_{i_n} g(i_1, \ldots, i_n)$$

where the limits are finite and the sequence $y_i = \lim_{i_2} \ldots \lim_{i_3} g(i_1, \ldots, i_n)$ is strictly increasing.

The proof of this can be obtained by iterating the procedure used in the proof of Lemma 3.1 in [5], and so it is omitted. With proposition 1 in mind we can think of expressions of the form $\sup\inf \ldots \Theta f$ as $\sup\lim \ldots \lim f$ and even expressions like (2) as $\sup \ldots \lim\lim F$ and so on; since when we have an effective sequence of $\sup\inf \ldots \Theta f$ expressions (as this was discussed above) we can transform it in a computable way to the $\sup\lim \ldots \lim f$ form.

3 The definition

After this informal discussion and in order to define the hierarchy of reals, we give a definition of the class of higher-type objects $\mathtt{HT}$ dependent on a system of ordinal notations $\mathcal{S}$. These computable objects ξ will be projected to $\mathbb{R}$ via suitable operators $\mathtt{Sup}, \mathtt{Inf}, \mathtt{Lim}$ and the expressions $\mathtt{Sup}\xi, \mathtt{Inf}\xi, \mathtt{Lim}\xi$ will represent what we informally called "prefix of an infinite order type over a higher type object".

Notation. Let $\mathcal{S}$ be a system of notation. We denote $|n|_{\mathcal{S}}$ the ordinal with notation n, and $p_{\mathcal{S}}$ the partial computable function which gives a notation for the predecessor of the ordinal represented by its argument (if this exists). We write $q_{\mathcal{S}}$ for the function which gives an index of a computable function whose succesive values are notations for an increasing sequence of ordinals converging to the ordinal represented by its argument. □

A system of notation $\mathcal{S}$ can be viewed as a well-founded tree: we say that $n <_{\mathcal{S}} m$ when $n, m \in \mathcal{S}$ and n can be obtained by applying successively the following operations starting from m:

- when we have $t \in \mathcal{S}$ which denotes a successor ordinal, we can aply $p_{\mathcal{S}}$ and get a new number

- when we have $s \in \mathcal{S}$ which denotes a limit ordinal, we can aply $q_{\mathcal{S}}$ and get any member of the resulting sequence.

One can see that $\langle \mathcal{S}, <_{\mathcal{S}} \rangle$ is a well-founded tree [3].

Lemma 1. *Let $\langle \mathcal{S}, <_{\mathcal{S}} \rangle$ be a notation system. There is a partial computable function f such that if $x \in \mathcal{S}$ then*

$$W_{f(x)} = \{y : y \leq_{\mathcal{S}} x\}$$

Proof. Enumerate in stages:

- *stage* 0: Enumerate x.
- *stage* $s+1$: Look at each member y of the finite set of elements already enumerated: if y is successor then enumerate $p_{\mathcal{S}}(y)$; and if limit, enumerate the first s elements of the sequence with index $q_{\mathcal{S}}(y)$.

It is not hard to prove that the algorithm works (if $y <_{\mathcal{S}} x$ is not enumerated, then prove by induction up to x that no z with $y <_{\mathcal{S}} z <_{\mathcal{S}} x$ is enumerated; a contradiction). □

Lemma 2. *Given a notation system $\langle \mathcal{S}, <_{\mathcal{S}} \rangle$ there is a partial computable function suc (called successor function) such that*

$$n <_{\mathcal{S}} m \Rightarrow suc(n,m) \leq_{\mathcal{S}} m \ \& \ p_{\mathcal{S}}(suc(n,m)) = n$$

Proof. To compute $suc(n,m)$ start enumerating the predecessors of m checking simulteneously the successor elements x whether $p_{\mathcal{S}}(x) = n$. When you find such number x, output x. □

Notation. We use the lower case Greek letters $\alpha, \beta, \gamma, \delta$ to denote ordinals and (in general) m, n, t, s, i, x for naturals. Also, we assume an effective enumeration of the partial computable functions $\{\lambda_i\}$, a pairing function $\langle ., . \rangle$ and its inverses $(.)_1, (.)_2$. We denote n-vectors by **bold face** letters. Finally in a prefix $\sup_{i_1} \inf_{i_2} \ldots \Theta_{i_m}$, Θ denotes the m-th term of this alternating sequence of $\sup, \inf$ (similarly for $\inf_{i_1} \sup_{i_2} \ldots \Theta_{i_m}$). □

[3] If one takes Kleene's $\mathcal{O}$ then the order relation described here is identical to $<_{\mathcal{O}}$.

Definition 2. Given a system of notation $\mathcal{S}$ we define a hierarchy of functions $\mathtt{HT} = \cup_{\gamma<\delta} K^\gamma$ (where δ is the first ordinal not having a notation under $\mathcal{S}$) and index them simulteneously by induction up to δ. Let $\{c_n\}_{n\in A}$ be an effective numbering of constants in $\mathbb{Q}$ (A is a computable set of natural numbers).

- *stage 0*:We set

$$\xi_{\langle n,t\rangle} = c_n$$

for all $n \in A$ and t with $|t|_{\mathcal{S}} = 0$. We denote the set of all such indices $\langle n, t\rangle$ by I^0 and $K^0 = \{\xi_i : i \in I^0\}$.

- *stage* $\beta + m$ (here β is limit or 0 and $m > 0$): We set

$$\xi_{\langle n,t\rangle} : \omega^m \to K^\beta$$
$$\xi_{\langle n,t\rangle}(\boldsymbol{x}) = \xi_{\lambda_n(\boldsymbol{x})}$$

for $t \in \mathcal{S}$ and all indices n of computable functions λ_n of m variables such that

1. $\forall \boldsymbol{x}\ \lambda_n(\boldsymbol{x}) \in I^\beta$
2. $\forall \boldsymbol{i}\ (\lambda_n(\boldsymbol{i}))_2 <_{\mathcal{S}} t$
3. $|t|_{\mathcal{S}} = \beta + m$

We denote the set of all such indices $\langle n, t\rangle$ by $I^{\beta+m}$ and $K^{\beta+m} = \{\xi_i : i \in I^{\beta+m}\}$.

- *stage* α (where α is limit): We set

$$\xi_{\langle n,t\rangle} : \omega \to \cup_{\gamma<\alpha} K^\gamma$$
$$\xi_{\langle n,t\rangle}(x) = \xi_{\lambda_n(x)}$$

for $t \in \mathcal{S}$ and all indices n of computable functions λ_n of one variable such that

1. $\forall x\ \lambda_n(x) \in \cup_{\gamma<\alpha} I^\gamma$ (and so, $\forall x\ |(\lambda_n(x))_2|_{\mathcal{S}} < \alpha$)

2. $\sup_i |(\lambda_n(i))_2|_{\mathcal{S}} = \alpha$
3. $\forall i\ (\lambda_n(i))_2 <_{\mathcal{S}} t$
4. $|t|_{\mathcal{S}} = \alpha$

We denote the set of all such indices $\langle n, t\rangle$ by I^α and $K^\alpha = \{\xi_i : i \in I^\alpha\}$.
Finally, we set $\mathtt{HT} = \cup_{\gamma<\delta} K^\gamma$, $I = \cup_{\gamma<\delta} I^\gamma$ and we say that $\xi \in \mathtt{HT}$ has *rank* $|\xi| = \gamma$, if it was generated at stage γ (i.e. it has an index in I^γ).

Note that for any object ξ we generate in the above definition, we simultaneously code the stage it was generated into its index. This comes from our requirement to be able to select not only a program of an object of lower rank, but also the rank of that object.

Definition 3. We define operators $\mathtt{Sup}, \mathtt{Inf}, \mathtt{Lim}, :\subseteq \mathtt{HT} \to \mathbb{R}$.
If $|\xi| = 0$ then

$$\mathtt{Sup}\ \xi = \xi; \quad \mathtt{Inf}\ \xi = \xi; \quad \mathtt{Lim}\ \xi = \xi$$

If $|\xi| = \beta + m$ then

$$\mathtt{Sup}\xi = \sup_{i_1} \inf_{i_2} \ldots \Theta_{i_m} \overbrace{\mathtt{Lim}\underbrace{\xi(i)}_{\in K^\beta}}^{\in \mathbb{R}}$$

$$\mathtt{Inf}\xi = \inf_{i_1} \sup_{i_2} \ldots \Theta_{i_m} \overbrace{\mathtt{Lim}\underbrace{\xi(i)}_{\in K^\beta}}^{\in \mathbb{R}}$$

$$\mathtt{Lim}\xi = \lim_{i_1} \lim_{i_2} \ldots \lim_{i_m} \overbrace{\mathtt{Lim}\underbrace{\xi(i)}_{\in K^\beta}}^{\in \mathbb{R}}$$

And if $|\xi| = \alpha$ limit then

$$\mathtt{Sup}\xi = \sup_i \overbrace{\mathtt{Lim}\ \underbrace{\xi(i)}_{\in \cup_{\gamma<a} K^\gamma}}^{\in \mathbb{R}}$$

$$\mathtt{Inf}\xi = \inf_i \overbrace{\mathtt{Lim}\ \underbrace{\xi(i)}_{\in \cup_{\gamma<a} K^\gamma}}^{\in \mathbb{R}}$$

$$\mathtt{Lim}\xi = \lim_i \overbrace{\mathtt{Lim}\ \underbrace{\xi(i)}_{\in \cup_{\gamma<a} K^\gamma}}^{\in \mathbb{R}}$$

In the above equations (and more generally) we always assume that the *sup*, *inf*, *lim* on the right-hand side exist (and the function to which they apply is total). And finally we define the $\mathcal{S}$-hierarchy of reals:

$$\mathtt{Sup}^0 = \mathtt{Inf}^0 = \mathtt{Lim}^0 = \text{ The computable numbers}$$

$$\begin{aligned}
\mathtt{Sup}^{\beta+1} &= \{\mathtt{Sup}\xi : \xi \in K^\beta\} \\
\mathtt{Inf}^{\beta+1} &= \{\mathtt{Inf}\xi : \xi \in K^\beta\} \\
\mathtt{Lim}^{\beta+1} &= \{\mathtt{Lim}\xi : \xi \in K^\beta\}
\end{aligned}$$

$$\begin{aligned}
\mathtt{Sup}^\alpha &= \cup_{\gamma<\alpha} \mathtt{Sup}^\gamma \\
\mathtt{Inf}^\alpha &= \cup_{\gamma<\alpha} \mathtt{Inf}^\gamma \\
\mathtt{Lim}^\alpha &= \cup_{\gamma<\alpha} \mathtt{Lim}^\gamma
\end{aligned}$$

where α is limit.

In a similar way we can define the classes $\mathtt{Sup}^\gamma{}_n, \mathtt{Inf}^\gamma{}_n, \mathtt{Lim}^\gamma{}_n$ of sequences of n arguments for arbitrary n and the above definition would be a special case for $n = 0$. It is not difficult to verify then the following

Proposition 2. *A sequence* $\{x_m\}$ *is in* $\mathtt{Sup}_{\mathtt{n}}^{\gamma+\mathtt{t}}$ *if there is* $\{y_{mk}\} \in \mathtt{Sup}_{\mathtt{t+n}}^{\gamma}$ *(or* $\mathtt{Inf}_{\mathtt{t+n}}^{\gamma}$*; or* $\mathtt{Lim}_{\mathtt{t+n}}^{\gamma}$*) such that*

$$x_{\boldsymbol{m}} = \sup_{k_1} \inf_{k_2} \ldots \Theta_{k_t} y_{\boldsymbol{m}\boldsymbol{k}}$$

The cases $\mathtt{Inf}_{\mathtt{n}}^{\gamma+\mathtt{t}}$ *and* $\mathtt{Lim}_{\mathtt{n}}^{\gamma+\mathtt{t}}$ *are analogous.*

4 Normal form of the hierarchy

In this section we give a characterization of the reals belonging to a class K^β. Then we prove that the hierarchy of reals is in fact independent of the system of notation $\mathcal{S}$ used in its definition. By Φ_e we mean the e-th partial computable functional with rational values and $\{H(n)\}_{n\in\mathcal{S}}$ is the family of sets defined as in the hyperarithmetical hierarchy (see [3]) but with $\mathcal{S}$ in place of $\mathcal{O}$. For reference we give the following

Definition 4. For a system of notation $\mathcal{S}$ define the family of sets $\{H(n)\}_{n\in\mathcal{S}}$ as follows. For all $n \in \mathcal{S}$,

$$H(n) = \begin{cases} \emptyset & , |n|_{\mathcal{S}} = 0 \\ (H(p_{\mathcal{S}}(n)))' & , |n|_{\mathcal{S}} \text{ successor} \\ \{\langle i,j\rangle \mid j <_{\mathcal{S}} n \wedge i \in H(j)\} & , |n|_{\mathcal{S}} \text{ limit} \end{cases}$$

Theorem 1. *For any* $\xi_n \in \mathtt{HT}$ *we can find uniformly in its index* n*, programs* e_i, q_i *and computable functions* ν_i *such that*

1. if $|\xi_n|$ is limit:

$$\begin{aligned} \mathtt{Sup}\ \xi_n &= \sup_x \Phi_{e_1}(H(\nu_1(x)); x) \\ \mathtt{Inf}\ \xi_n &= \inf_x \Phi_{e_2}(H(\nu_2(x)); x) \\ \mathtt{Lim}\ \xi_n &= \lim_x \Phi_{e_3}(H(\nu_3(x)); x) \end{aligned} \tag{3}$$

and

- $\forall x\ \nu_i(x) <_{\mathcal{S}} (n)_2$
- $|\xi_n| = \sup_x |\nu_i(x)|_{\mathcal{S}}$

2. if $|\xi_n|$ successor

$$\begin{aligned} \mathtt{Sup}\ \xi_n &= \sup_x \Phi_{q_1}(H((n)_2); x) \\ \mathtt{Inf}\ \xi_n &= \inf_x \Phi_{q_2}(H((n)_2); x) \\ \mathtt{Lim}\ \xi_n &= \lim_x \Phi_{q_3}(H((n)_2); x) \end{aligned} \tag{4}$$

provided that the expressions on the left part of the equations are defined. If $|\xi_n| = 0$ then the index e_i is that of the constant ξ_n. Moreover in the above, the function under the sup is strictly increasing, and the function under the inf, strictly decreasing.

In the proof of the theorem we use freely the basic facts for computable functions (as the smn theorem) and ordinal notations, as well as proposition 1. We also use the following

Lemma 3. *Suppose that*

- $f(x, y) = \Phi_e(H(\lambda(x, y)); x, y)$
- $\forall x, y\ \lambda(x, y) <_{\mathcal{S}} \lambda_*(x)$

Then, we can find uniformly in λ, λ_*, e a program e_1 such that

$$f(x, y) = \Phi_{e_1}(H(\lambda_*(x)); x, y)$$

Proof. From the definition of $\{H(n) : n \in \mathcal{S}\}$ it follows that we can find uniformly in x, y an algorithm for the reduction $H(x) \leq_T H(y)$ (assuming $x <_{\mathcal{S}} y$). Now e_1 says: take x, y as input to the program e but to any questions which may occur during the computation answer in the way $H(\lambda(x, y))$ would answer (by consulting $H(\lambda_*(x))$). □

Proof of the theorem. First we give an algorithm which takes n and if $\mathtt{Lim}\xi_n(\boldsymbol{x})$ exists for all $\boldsymbol{x}$, then it outputs a program e and a computable function λ such that

$$\mathtt{Lim}\ \xi_n(\boldsymbol{x}) = \lim_y \Phi_e(H(\lambda(\boldsymbol{x})); \boldsymbol{x}, y)$$

and

1. *if* $|\xi_n|$ *is limit then*

   - $|\xi_n| = \sup_x |\lambda(x)|_{\mathcal{S}}$
   - $\forall x\ \lambda(x) <_{\mathcal{S}} (n)_2$

2. *and if* $|\xi_n|$ *is successor then*

   - $|\xi_n| = |b|_{\mathcal{S}} + m$
   - $\forall \boldsymbol{x}\ \lambda(\boldsymbol{x}) = b$

   for $m > 0$ and $b <_{\mathcal{S}} (n)_2$ ($|b|_{\mathcal{S}}$ is the maximum limit less than $|\xi_n|$).

Given $n \in I^\beta$ the algorithm at some point will call itself but at an $m \in I$ which is generated at an earlier stage according to definition (2), i.e. $m \in I^\gamma$ for some $\gamma < \beta$. Hence, after finitely many calls it will reach some $m \in I^0$, in which case it is very easy to give an answer.

The algorithm

Given n we see whether $\xi_n \in K^0$. If yes, then we find the desired e, λ. Otherwise, we check whether it belongs to a limit level K^α of HT or to a successor level $K^{\beta+m}$.

- *Case* K^α*:* By definition we have $\mathtt{Lim}\xi_n(x) = \mathtt{Lim}\xi_{\lambda_{(n)_1}(x)}$. Now we can find uniformly in n functions τ, m such that given x, $|(\lambda_{(n)_1}(x))_2|_{\mathcal{S}} = |\tau(x)|_{\mathcal{S}} + m(x)$ where $\tau(x) <_{\mathcal{S}} (\lambda_{(n)_1}(x))_2$ and $|\tau(x)|_{\mathcal{S}}$ limit, $m(x) \in \omega$. So we have

  $$\mathtt{Lim}\xi_n(x) = \lim_{i_1} \ldots \lim_{i_m} \mathtt{Lim}\xi_{\lambda_{(n)_1}(x)}(\boldsymbol{i})$$

  where $m = m(x)$. By applying the algorithm itself on the index $\lambda_{(n)_1}(x)$ we get a program e and a function ν_1 with the property

  $$\mathtt{Lim}\ \xi_{\lambda_{(n)_1}(x)}(\boldsymbol{i}) = \lim_t \Phi_e(H(\nu(\boldsymbol{i}, x)); x, \boldsymbol{i}, t)$$

  and

  - if $m(x) = 0$: $\forall \boldsymbol{i}\ \nu(\boldsymbol{i}, x) <_{\mathcal{S}} \tau(x)$ (and $\sup_{\boldsymbol{i}} |\nu(\boldsymbol{i}, x)|_{\mathcal{S}} = \tau(x)$)
  - if $m(x) > 0$: $\forall \boldsymbol{i}\ \nu(\boldsymbol{i}, x) = \tau(x)$

From e, τ and ν we can find (by lemma 3) a program e_1 such that

$$\Phi_e(H(\nu(\boldsymbol{i}, x)); x, \boldsymbol{i}, t) = \Phi_{e_1}(H(\tau(x)); x, \boldsymbol{i}, t)$$

and so we have

$$\mathtt{Lim}\xi_n(x) = \lim_{i_1} \ldots \lim_{i_m} \lim_t \Phi_{e_1}(H(\tau(x)); x, \boldsymbol{i}, t)$$

and we can assume that all the limits except the first one are finite (because by proposition 1 we can find from e_1 another program which does the same job *and* this requirement is fulfilled). Now we know from the proof of Shoenfield's lemma that from e_1 we can find e_2 such that

$$\lim_{i_1} \ldots \lim_{i_m} \lim_t \Phi_{e_1}(H(\tau(x)); x, \boldsymbol{i}, t) = \lim_i \Phi_{e_2}(H(\lambda(x)); x, i)$$

where λ takes x and applies $m = m(x)$ times the successor function along the path of $(n)_2$. We output e_2 and λ.

- *Case $K^{\beta+m}$:* By definition we have

$$\mathtt{Lim}\xi_n(\boldsymbol{i}) = \mathtt{Lim}\xi_{\lambda_{(n)_1}(\boldsymbol{i})} = \lim_x \mathtt{Lim}\ \xi_{\lambda_{(n)_1}(\boldsymbol{i})}(x)$$

Now we call the algorithm for $\lambda_{(n)_1}(\boldsymbol{i})$ and (uniformly in n) we get a program e and a function ν with properties as in case K^α and

$$\mathtt{Lim}\ \xi_{\lambda_{(n)_1}(\boldsymbol{i})}(x) = \lim_y \Phi_e(H(\nu(\boldsymbol{i}, x)); x, y, \boldsymbol{i})$$

Now find e_1, ν_* such that

$$\mathtt{Lim}\ \xi_n(\boldsymbol{i}) = \lim_x \lim_y \Phi_{e_1}(H(\nu_*(\boldsymbol{i}, x)); x, y, \boldsymbol{i}) = \lim_x \Phi_{e_1}(H(\nu_*(\boldsymbol{i}, x)); x, \boldsymbol{i})$$

and $\forall x, \boldsymbol{i}\ \nu_*(\boldsymbol{i}, x) <_{\mathcal{S}} (\lambda_{(n)_1})_2(\boldsymbol{i})$ (we can do this because the operator of proposition 1 is primitive recursive). In the same way as above, from $\lambda_{(n)_1}$, e_1 and ν_* we can find e_2 such that

$$\Phi_{e_1}(H(\nu_*(\boldsymbol{i}, x)); x, \boldsymbol{i}) = \Phi_{e_2}(H((\lambda_{(n)_1}(\boldsymbol{i}))_2); x, \boldsymbol{i})$$

and so

$$\mathtt{Lim}\ \xi_n(\boldsymbol{i}) = \lim_x \Phi_{e_2}(H((\lambda_{(n)_1}(\boldsymbol{i}))_2); x, \boldsymbol{i})$$

We output $e_2, (\lambda_{(n)_1})_2$.

One could think of the above algorithm as a recursion over the well-founded tree $\langle I, <_* \rangle$ where $n <_* m \iff (n)_2 <_{\mathcal{S}} (m)_2$. It is not difficult to prove by induction up to the least ordinal which does not receive an $\mathcal{S}$-notation, that the algorithm does its job.

To prove the theorem, given $n \in I$ we check whether $|(n)_2|_{\mathcal{S}}$ is 0, successor $\beta + m$ or limit α. The case 0 is trivial.

- *Case $\beta + m$:*

$$\mathtt{Sup}\xi_n = \sup_{x_1} \inf_{x_2} \ldots \Theta_{x_m} \mathtt{Lim}\xi_n(\boldsymbol{x}) = \sup_{x_1} \inf_{x_2} \ldots \Theta_{x_m} \lim_y \Phi_e(H(b); \boldsymbol{x}, y)$$

Now, as usual, we can assume that the sup, inf, lim are finite (except for the first one) and find e_1 such that

$$\sup_{x_1} \inf_{x_2} \ldots \Theta_{x_m} \lim_y \Phi_e(H(b); \boldsymbol{x}, y) = \sup_x \Phi_{e_1}(H((n)_2); x)$$

because from b applying m times the successor along $(n)_2$ we get $(n)_2$. We output e_1.

- *Case α:*

$$\mathtt{Sup}\xi_n = \sup_x \mathtt{Lim}\xi_n(x) = \sup_x \lim_y \Phi_e(H(\lambda(x)); x, y) =$$
$$\sup_x \Phi_{e_1}(H(suc(\lambda(x), (n)_2)); x)$$

We output e_1 and λ_* (where $\lambda_*(x) = suc(\lambda(x), (n)_2)$).

The cases $\mathtt{Inf}, \mathtt{Lim}$ are similar. $\square$

Note that any number of the form $\lim_x \Phi_q(H((n)_2); x)$ can be written as $\lim_x \Phi_e(H(\nu(x)); x)$ (with $\nu(x) <_{\mathcal{S}} (n)_2$), and similarly with $\sup, \inf$.

Theorem 2. *The converse of theorem 1 holds, i.e. given a real of the form*

- $\sup_x \Phi_e(H(\nu(x)); x)$
- $|m| = \sup_x |\nu(x)|_{\mathcal{S}} =$ *limit*
- $\forall x\ \nu(x) <_{\mathcal{S}} m$

we can find uniformly in e *,* ν*, a program* n *such that*

$$\sup_x \Phi_e(H(\nu(x)); x) = \mathtt{Sup}\xi_{\langle n,m\rangle}$$

and given a real of the form

- $\sup_x \Phi_e(H(m); x)$
- $m \in \mathcal{S}$
- $|m|_{\mathcal{S}}$ *successor*

we can find uniformly in e *and* m *an index* n *such that*

$$\sup_x \Phi_e(H(m); x) = \mathtt{Sup}\xi_n$$

An analogous result holds for the cases of $\mathtt{Inf}, \mathtt{Lim}$.

Proof. An algorithm is needed similar to the one of the proof of theorem 1 but doing the converse job. The details are omitted. $\square$

The above two theorems give a characterization of the real numbers which belong to a class (e.g. $\mathtt{Sup}^\alpha$) of the hierarchy in terms of $\sup, \inf, \lim$ of a function $f : \mathbb{N} \to \mathbb{Q}$.

When we say that a $\xi \in \mathtt{HT}$ is on a particular branch (of $\langle I, <_*\rangle$) we mean that its index lies on that branch. The following question arises: suppose we are given two branches of $\langle I, <_*\rangle$ of the same length. Then, would the corresponding classes of $\sup \xi$ for ξ on the one or the other branch differ? The following theorem says no.

Theorem 3. *Suppose we are given n and m lying on $\langle I, <_* \rangle$ such that $|(n)_2|_{\mathcal{S}} = |(m)_2|_{\mathcal{S}}$. We can find uniformly in n, m a program e and a function ν such that if* $\mathtt{Sup}\xi_n$ *is defined, then*

1. *if $|\xi_n|$ limit*

   - $\mathtt{Sup}\xi_n = \sup_x \Phi_e(H(\nu(x)); x)$
   - $\forall x\ \nu(x) <_{\mathcal{S}} (m)_2$

2. *if $|\xi_n|$ successor*

$$\mathtt{Sup}\xi_n = \sup_x \Phi_e(H((m)_2); x)$$

Similarly for $\mathtt{Inf}, \mathtt{Lim}$.

In the proof we use implicitly some lemmas which where originally proved (by Spector) for the case of $\mathcal{S} = \mathcal{O}$ (see [3]) but the same proofs work for an arbitrary system $\mathcal{S}$ (by replacing $<_{\mathcal{O}}$ with $<_{\mathcal{S}}$).

Lemma 4.

$$[x \in \mathcal{S}\ \&\ y \in \mathcal{S}\ \&\ |x|_{\mathcal{S}} = |y|_{\mathcal{S}}] \Longrightarrow H(x) \leq_T H(y),$$

uniformly in x and y. And

$$[x \in \mathcal{S}\ \&\ y \in \mathcal{S}\ \&\ |x|_{\mathcal{S}} = |y|_{\mathcal{S}} = successor] \Longrightarrow H(x) \equiv H(y),$$

uniformly in x and y.

Lemma 5. *For $x \in \mathcal{S}$*

$$\{u : u \in \mathcal{S}\ \&\ |u|_{\mathcal{S}} = |x|_{\mathcal{S}}\}$$

is recursive in $H(x)''$.

Proof of the theorem. If $|\xi_n|$ limit,
define $\nu(x) = suc(suc(\nu_1(x)))$ (ν_1 is from the theorem 1) and the program e says:

start enumerating the $<_{\mathcal{S}}$ - predecessors of $(m)_2$ until you find the one x with $|x|_{\mathcal{S}} = |\nu_1(x)|_{\mathcal{S}}$. Now run the program e_1 of theorem 1 and to any questions that may occur, answer them in the way $H(\nu_1(x))$ would answer them (by consulting $H(\nu(x))$, as we have $H(\nu_1(x)) \leq_T H(\nu(x))$ uniformly in x).

If $|\xi_n|$ successor,

we can find uniformly in n, m a computable isomorphism for the equivalence $H((n)_2) \equiv H((m)_2)$. The program e now says: take x and apply e_1 of theorem 1. To any questions that may occur, answer the way $H((n)_2)$ would answer by consulting $H((m)_2)$.
The cases $\mathtt{Inf}, \mathtt{Lim}$ are similar. □

Note that in the above theorem it is enough to have $|(n)_2|_{\mathcal{S}} \leq |(m)_2|_{\mathcal{S}}$ instead of $|(n)_2|_{\mathcal{S}} = |(m)_2|_{\mathcal{S}}$.

So far our hierarchy of reals is dependent on a fixed system of notation $\mathcal{S}$. And we saw that this hierarchy remains the same if we take as system of notation any branch of $\mathcal{S}$. So we only need to consider univalent systems of notation. [4] But it is well known that any univalent system of notation (i.e. a branch of a system of notation) is computably isomorphic to a branch of Kleene's $\mathcal{O}$. So, by combining the above results we get

Corolarry 1. *The hierarchy of reals defined in definition 3 is independent of the system of notation $\mathcal{S}$ used in the sense that if $\mathcal{S}$ assigns notations to the ordinals up to $\beta \in On$ and $\mathcal{S}'$ up to $\alpha \in On$ with $\beta \leq \alpha$ then the first hierarchy is an initial segment of the second.*

5 The hierarchy theorem

Thus we have defined a unique hierarchy of reals which we get if we take $\mathcal{S}$ to be a maximal system of notation (i.e. one which assigns notations to all ordinals up to ω_1^{CK}). The next theorem asserts that the hierarchy never collapses.

Theorem 4. *For all $\alpha, \beta < \omega_1^{CK}$*

$$\alpha < \beta \Longrightarrow \begin{array}{l} \mathtt{Sup}^\alpha \subsetneq \mathtt{Sup}^\beta,\ \mathtt{Inf}^\beta,\ \mathtt{Lim}^\alpha \\ \mathtt{Inf}^\alpha \subsetneq \mathtt{Inf}^\beta,\ \mathtt{Sup}^\beta,\ \mathtt{Lim}^\alpha \\ \mathtt{Lim}^\alpha \subsetneq \mathtt{Lim}^\beta,\ \mathtt{Inf}^\beta,\ \mathtt{Sup}^\beta \end{array}$$

[4] a univalent system of notation is one that assigns exactly one notation to every ordinal lying on an initial segment of the ordinals.

Definition 5. In the following we write $f \leq_{wtt^*} \emptyset^\alpha$ if α is limit and

- $f(x) = \Phi(H(\lambda(x)); x)$
- $\sup_x |\lambda(x)|_{\mathcal{S}} = \alpha$
- $\forall x$ $\lambda(x)$ lie on a branch of a system of notation $\mathcal{S}$

or if α is successor and $f \leq_T H(y)$ for $|y|_{\mathcal{S}} = \alpha$ ($\mathcal{S}$ a system of notation).

We know from the above that this definition is independent of $\mathcal{S}$.

Proof of theorem 4. The inclusions $\subseteq$ follow from theorem 1. We want to prove e.g. $\mathtt{Sup}^\alpha \subset \mathtt{Sup}^{\alpha+1}$ when α is limit. Assume that λ is a computable function whose successive values form a sequence of notations for ordinals tending to α. Its enough to define a diagonal sequence $\{x_s\} \leq_{wtt^*} \emptyset^\alpha$ which fulfills the requirements:

$$R_{\langle e,i\rangle}: \quad \left.\begin{array}{l} \Phi_e(H(\lambda(i)); s) \text{ total} \\ \& \text{ strictly decreasing} \end{array}\right\} \Longrightarrow x = \sup_s x_s \neq \inf_s \Phi_e(H(\lambda(i)); s)$$

and in particular we can make them differ at their $\langle e, i\rangle$-th digit. But such a number $\inf_s \Phi_e(H(\lambda(i)); s)$ will be of the form $y = 0.y_1 y_2 \ldots$ in binary expansion where

- $y_k = \lim_s \Phi_{e_*}(H(\lambda(i)); s, k)$
- the limit is finite with modulus of convergence not more than $k + 2$
- all values of $\lambda s.\Phi_{e_*}(H(\lambda(i)); s, k)$ equal 0 or 1

So it is enough to find a sequence $\{t_s\} \leq_{wtt^*} \emptyset^\alpha$ such that

$$\left.\begin{array}{l} \forall s\ f_{\langle e,i\rangle}(s) := \Phi_e(H(\lambda(i)); s, \langle e, i\rangle) \in \{0, 1\} \\ \forall s \geq \langle e, i\rangle + 2\ f_{\langle e,i\rangle}(s) = f_{\langle e,i\rangle}(\langle e, i\rangle + 2) \end{array}\right\} \Longrightarrow$$
$$t_{\langle e,i\rangle} = 1 - \lim_s \Phi_e(H(\lambda(i)); s, \langle e, i\rangle)$$

In that case our diagonal real would be $x = 0.t_1 t_2 \ldots$ (as binary expansion). To find $t_{\langle e,i\rangle}$ check whether $\Phi_e(H(\lambda(i)); \langle e, i\rangle + 2, \langle e, i\rangle)$ is defined. If not, put $t_{\langle e,i\rangle} = 0$. Otherwise put

$$t_{\langle e,i\rangle} = 1 - \Phi_e(H(\lambda(i)); \langle e, i\rangle + 2, \langle e, i\rangle)$$

It is now easy to see that our requirements are fulfilled and also that $\{t_s\} \leq_{wtt^*} \emptyset^\alpha$.

In the case of a successor ordinal $\beta + 1$ we have to diagonalize over all $\sup_x \Phi_e(H(t_0); x)$ $(|t_0|_{\mathcal{S}} = \beta)$ within $\mathtt{Sup}^{\beta+1}$ and the proof is similar (even easier).

Also, the rest of the cases are proved in the same way (for the case of $\mathtt{Lim}^\alpha \subsetneq \mathtt{Inf}^\alpha$ take the diagonal sequence which starts with $0.1111\ldots$ and its n-th term is $0.t_1 t_2 \ldots t_n 111\ldots$). $\square$

Theorem 5. *If $m > 0$ and β limit and computable then*

- $\mathtt{Sup}^{\beta+\mathtt{m}+1} \cap \mathtt{Inf}^{\beta+\mathtt{m}+1} = \mathtt{Lim}^{\beta+\mathtt{m}}$
- $\mathtt{Sup}^{\beta+1} \cap \mathtt{Inf}^{\beta+1} \supsetneq \mathtt{Lim}^\beta$
- $\mathtt{Sup}^\beta = \mathtt{Inf}^\beta = \mathtt{Lim}^\beta$

and for any computable ordinal α

$$\mathtt{Sup}^\alpha \cup \mathtt{Inf}^\alpha \subsetneq \mathtt{Sup}^{\alpha+1} \cap \mathtt{Inf}^{\alpha+1}$$

Proof. This proof is in a sense a relativization of the proof of Lemma 3.3 in [5]. It is not difficult to see that a number x in $\mathtt{Sup}^{\beta+\mathtt{m}+1}$ can be written as

$$x = \sup_i \inf_j f_1(i,j)$$

with

- $f_1 \leq_{wtt*} \emptyset^{\beta+m-1}$
- $f_1(i,j) < f_1(i,j+1)$
- $\sup_j f_1(i,j) > \sup_j f_1(i+1,j)$

A dual statement holds for a number x in $\mathtt{Inf}^{\beta+\mathtt{m}+1}$:

$$x = \inf_i \sup_j f_2(i,j)$$

with

- $f_2 \leq_{wtt*} \emptyset^{\beta+m-1}$
- $f_2(i,j) > f_2(i,j+1)$
- $\inf_j f_2(i,j) < \inf_j f_2(i+1,j)$

We have $\inf_j f_2(i,j) < x < \sup_j f_1(i,j)$. We define the following function:

$$e(i) = \mu j[f_2(i,j) < f_1(i,j)]$$

It is $e \leq_{wtt*} \emptyset^{\beta+m-1}$. Define

$$f(i) = f_2(i, e(i))$$

and we have $f \leq_{wtt*} \emptyset^{\beta+m-1}$. It is

$$\inf_j f_2(i,j) \leq f_2(i,e(i)) = f(i) < f_1(i,e(i)) \leq \sup_j f_1(i,j)$$

So $\lim_i f(i) = x$ which means that $x \in \mathtt{Lim}^{\beta+\mathtt{m}}$. To prove that $\mathtt{Sup}^{\beta+1} \cap \mathtt{Inf}^{\beta+1} \supsetneq \mathtt{Lim}^{\beta}$ it is enough to define a diagonal real as in the proof of theorem 4 (in the case of limit ordinal α). The rest of the theorem follows easily. □

6 Relation to the hyperarithmetical hierarchy

As mentioned in the introduction, a real number x can be seen as a function $\mathbb{N} \to \mathbb{N}$, e.g. the characteristic function of the set A which corresponds to its binary expansion ($n \in A \iff$ the n-th digit of x is 1). So, the hyperarithmetical hierarchy of classical computability theory also applies to reals and the question is how do these hierarchies relate to each other (they have the same height). The following theorem shows that the hierarchy defined in this paper is wider than the hyperarithmetical hierarchy (but the two hierarchies contain the same class of reals).

Definition 6 (Kleene). Define the hyperarithmetical hierarchy as follows [5]. Let $\beta = \gamma + m$ where γ is limit or 0 and $\gamma = |y|_{\mathcal{O}}$ for $y \in \mathcal{O}$. Define

[5] usually the classes which correspond to limit levels below are not considered and are replaced by the corresponing classes of the next level. However we include them, as we did in the definition of our hierarchy, because they are important classes.

$$\Sigma^0_\beta = \Sigma^{H(y)}_m$$
$$\Pi^0_\beta = \Pi^{H(y)}_m$$
$$\Delta^0_\beta = \Sigma^0_\beta \cap \Pi^0_\beta$$

Lemma 6. *If a set is H-c.e. where H is a hyperarithmetical set of a limit level (i.e. $H = H(y)$, for $y \in \mathcal{O}, |y|_\mathcal{O}$ limit), then it is also enumerated by a function of the form:*

$$f(n) = \Phi(H(\lambda(n)); n)$$

where λ is any computable function such that

- $\forall n\ \lambda(n) <_\mathcal{S} y_1$
- $\sup_i |\lambda(i)|_\mathcal{S} = |y_1|_\mathcal{O}$

and $\mathcal{S}$ any system of notation which assigns notation to $|y|_\mathcal{O}$ and $|y_1|_\mathcal{S} = |y|_\mathcal{O}$.

Proof. Let $m_0 \in H$. Without loss of generality we assume $\mathcal{S} = \mathcal{O}$ and $\lambda <_\mathcal{O}$ - increasing. Suppose that A is enumerated by $\Phi(H; n)$. Note that $H = \{\langle i, j\rangle : i \in H(j) \wedge j <_\mathcal{O} y\}$. Now at stage s we enumerate the $<_\mathcal{O}$ - predecessors (which appear by the s-th stage of the particular enumeration) of $\lambda(0), \lambda(1), \ldots, \lambda(s)$, in a set D_s. Then we run the computation $\Phi(H; n)$ for each $n < s$ and if some question "$\langle i, j\rangle \in H$? " occurs we do the following: check whether $j \in D_s$. If not, output m_0 and forget this (unfinished) computation for this stage. If yes, then answer the question (which is really about whether "$i \in H(j)$?") by consulting $H(\lambda(s))$, and continue the computation doing the same thing, until the computation is finished (so you output the result) or canceled because we are unable to answer a question (this is the case when $j \notin D_s$). After cancel or finish all computations for $n < s$, go to stage $s+1$. It is not difficult to see that the program we defined does its job (if equipped with the "variable-oracle" $H(\lambda(n))$). □

Proposition 3. *(i) If β is successor then*

$$A \in \Sigma^0_\beta(\Pi^0_\beta) \underset{\not\Leftarrow}{\Rightarrow} x[A] \in \mathtt{Sup}^\beta(\mathtt{Inf}^\beta)$$

and if α is limit

$$A \in \Sigma^0_\alpha (= \Pi^0_\alpha) \iff x[A] \in \mathtt{Sup}^\alpha (= \mathtt{Inf}^\alpha)$$

and for any $\gamma < \omega_1^{CK}$

$$A \in \Delta^0_\gamma \iff x[A] \in \mathtt{Lim}^\gamma$$

(ii)

$$A \in \Delta^1_1 \iff x[A] \in \cup_{\gamma < \omega_1^{CK}} \mathtt{Lim}^\gamma$$

Proof. The only interesting part is to find a set $A \notin \Sigma^0_\beta$ such that $x = x[A] \in \mathtt{Sup}^\beta$ (we can similarly do the dual case). Let $\beta = \gamma + 1$. Our requirements are:

$$R_e : x \neq 0.W_e^H$$

where $0.A = x[A]$ and H is a set of the hyperarithmetical hierarchy of level γ (i.e. $H = H(y)$ for some $y \in \mathcal{O}$ with $|y|_\mathcal{O} = \gamma$). This is because $A \in \Sigma^0_\beta \iff A$ is c.e. in H. Now we keep the $2e+1$ - th place in the decimal expansion of x, for the e-th requirement. We start with the rational $0.001010101\ldots$. At stage s we have an enumeration of W_i^H for $i < s$ and we output the rational we had in the previous stage with the following changes: We check for each $2i+1 < s$ whether it belongs to the set of elements of $0.W_i$ so far enumerated. If yes, we put 0 in the $2i+1$-th position and 1 in the $2i$ position. Our sequence is increasing, and its supremum x will satisfy all of our requirements. Moreover the sequence is of the form $\Phi(H(y); n)$, if γ successor, or $\Phi(H(\lambda(n)); n)$ (with $\forall n\ \lambda(n) <_\mathcal{O} y$ and $\sup_n |\lambda(n)|_\mathcal{O} = \gamma$) if γ limit (the last due to lemma 6) [6].
To prove $\Delta^0_\gamma = \mathtt{Lim}^\gamma$ use Shoenfield's lemma and note that for any function of the form $f(n) = \Phi(H(y); n)$ where $|y|_\mathcal{O} = \gamma$ limit, there exists $f_* \leq_{wtt*} \emptyset^\gamma$ such that $\lim_n f(n) = \lim_i f_*(i)$. □

[6] the idea for this diagonalization is from Downey[1].

7 Fixed point theorems

In this section we prove two fixed point theorems regarding computation with a non-fixed oracle, i.e. of the form $f(x) = \Phi(H(\lambda(x)); x)$. Here, Φ denotes a computable functional with rational values.

Lemma 7. *(i) If A is a computable set and $\mathbb{N} - A$ does not contain a collection of indices for all partial computable functions , then every computable function f has a fixed point in A. Moreover, given A we can find it uniformly in f.* [7]

(ii) If $A \leq_T B$ and $\mathbb{N} - A$ does not contain a collection of indices for all partial computable functions , then every B- computable function f has a fixed point in A. Moreover, given A we can find it uniformly in B and f.

(iii) The above (i), (ii) hold for functionals instead of functions: If $A \leq_T B$ and $\mathbb{N} - A$ does not contain a collection of indices for all partial computable functionals, then for every B- computable function f, given A we can find uniformly in B and f an $e \in A$ such that $\Phi_{\lambda(e)} \simeq \Phi_e$.

Proof. (i) Take

$$f_*(x) = \begin{cases} f(x), & x \in A \\ x_0, & \text{otherwise} \end{cases}$$

where $\forall x \notin A \, \neg[\lambda_x \simeq \lambda_{x_0}]$ (and so $x_0 \in A$). Then choose b such that

$$\lambda_{f_*(\lambda_x(x))} \simeq \lambda_{\lambda_b(x)}$$

Now the fixed point $\lambda_b(b)$ is in A.

(ii) This is a relativization of (i).

(iii) The proof is similar to the above.

□

[7] From this, it follows easily that every partial computable function ψ with $Dom\psi$ containing the set of indices of a partial computable function, has a fixed point.

Theorem 6. *Suppose that $f(s,n) \simeq \Phi_e(H(\lambda_d(s)); s, n)$ for some indices e, d of partial computable functions. Then we can find $(\emptyset' \oplus <_{\mathcal{S}})'$ - uniformly in e, d an index e_* such that:*

$$f(s,n) \simeq \Phi_{e_*}(H(\lambda_{e_*}(s)); s, n)$$

Proof. Take

$$A = \{x : \forall n[\lambda_d(n) \downarrow \Longrightarrow \lambda_d(n) <_{\mathcal{S}} \lambda_x(n)]\}$$

A can be decided by a $(\emptyset' \oplus <_{\mathcal{S}})'$ -oracle. Obviously $\forall x \notin A,\ \neg[\lambda_d \simeq \lambda_x]$. Define (uniformly in e, d) a total function g such that:

$$\Phi_{g(x)}(H(\lambda_x(s)); s, n) \simeq \Phi_e(H(\lambda_d(s)); s, n)$$

whenever $\lambda_d(s) <_{\mathcal{S}} \lambda_x(s)$. By lemma 7 we can find $(\emptyset' \oplus <_{\mathcal{S}})'$- effectively an index $e_* \in A$ such that $\Phi_{g(e_*)} \simeq \Phi_{e_*}$.

So we have

$$\Phi_{e_*}(H(\lambda_e(s)); s, n) \simeq \Phi_e(H(\lambda_d(s)); s, n) \tag{5}$$

when $\lambda_d(s) <_{\mathcal{S}} \lambda_{e_*}(s)$. But since $e \in A$, (5) is true for all s. □

Theorem 7. *Suppose that $f(s,n) \simeq \Phi_{\lambda_e(s)}(H(\lambda_d(s)); n)$ for some indices e, d of partial computable functions. Then we can find $(\emptyset' \oplus <_{\mathcal{S}})'$ - uniformly in e, d an index e_* such that:*

$$f(s,n) \simeq \Phi_{\lambda_{e_*}(s)}(H(\lambda_{e_*}(s)); n)$$

Proof. The only difference from the previous proof is that now we find g such that

$$\Phi_{\lambda_{g(x)}(s)}(H(\lambda_x(s)); n) \simeq \Phi_{\lambda_e(s)}(H(\lambda_d(s)); n)$$

□

References

[1] R. G. Downey, Some computability-theoretical aspects of reals and Randomness, Notes based upon a short course of lectures given in the fall of 2000 at the University of Notre Dame, September 10, 2001 (see http://www.mcs.vuw.ac.nz/research/maths-pubs.shtml).

[2] R. Rettinger, X. Zheng, R. Gengler and B. von Braunmhl. Monotonically computable real numbers. *Mathematical Logic Quarterly*, **48**(2002), no 3, 459 - 479

[3] H.J. Rogers, *Theory of recursive functions and effective computability,* McGraw-Hill, 1967

[4] K. Weihrauch and X. Zheng. A finite hierarchy of recursively enumerable real numbers. MFCS'98, Brno, Czech Republic. Aug. 24 - 28, 1998, LNCS 1450, pp798–806, Springer 1998.

[5] K. Weihrauch and X. Zheng, *The Arithmetical hierarchy of real numbers, Mathematical Logic Quarterly*, **47** Issue 1 (2001) 51-65

[6] X. Zheng, K. Weihrauch and K. Ambos-Spies. Weakly computable real numbers. *Journal of Complexity* **16**(2000), 676-690.