Binary Enumerability of Real Numbers

(Extended Abstract)

Xizhong Zheng

Theoretische Informatik, FernUniversität Hagen
58084 Hagen, Germany
Email: *xizhong.zheng@fernuni-hagen.de*

Abstract. A real number x is called *binary enumerable*, if there is an effective way to enumerate all "1"-positions in the binary expansion of x. If at most k corrections for any position are allowed in the above enumerations, then x is called *binary k-enumerable*. Furthermore, if the number of the corrections is bounded by some computable function, then x is called *binary ω-enumerable*. This paper discusses some basic properties of binary enumerable real numbers. Especially, we show that there are two binary enumerable real numbers x and y such that their difference $x - y$ is not binary ω-enumerable (in fact we have shown that it is even of no "ω-r.e. Turing degree").

1 Introduction

There are many different ways to represent a real number by rational numbers. One of them is the binary expansion. In this way, any real number $x \in [0; 2]$ corresponds naturally to a subset $A \subseteq \mathbb{N}$ in the sense that $x = \sum_{n \in A} 2^{-n}$, i.e., A consists of all "1"-positions in the binary expansion of x. If we choose the finite set A to correspond to the rational number x, then such correspondence is one-to-one. We will call such set A a *binary set* of x and the real number x (which is usually denoted by x_A) is called a *binary real number* of A. It is well known that x_A is a computable real number, if and only if A is a recursive set. If a set A is recursively enumerable (r.e.), then its binary real number x_A is a limit of an increasing computable sequence of rational numbers, i.e., it is left computable (see [13]). As it is observed by Jockusch (see [9]), the converse is not true. That is, there is a left computable real number such that its binary set is not r.e.

It is well known that, a real number x is computable, if and only if its binary expansion is computable, namely, we can effectively write either "0" or "1" one bit after another. A correction in this procedure is not allowed. That is, if some bit "0" or "1" is written at some stage, then it cannot be changed later any more.

In the practice, the situation may be different. If we want to determine some real number x by giving its binary expansion, we will begin with an empty string (i.e., all bits are assumed to be "0" at the beginning) and write longer and longer strings of "0" and "1" to approximate x. In some stages, we may have to change some bits from "1" back to "0" and vice versa. According to the numbers of changes which are allowed in the this procedure, we get different type of the real numbers. Since additional 0's at the end of our binary expansions do not change its value, we can assume that, before a "1" is written first time to some position, there is already a "0" been written there. Then the first change at any position, if any, can only be from "0" to "1". If at most one change at any position is allowed, then the binary set of corresponding real number is r.e. and such real numbers are called *binary enumerable*. Therefore, x is binary enumerable, if and only if there is an r.e. set $A \subseteq \mathbb{N}$ such that $x = x_A$. Generally, we get a bigger class of real numbers which has less "effectivity", if more changes are allowed. For example, if k changes are allowed for any position, then the corresponding real numbers are called *binary k-enumerable*. More generally, if the number of such changes is bounded by some recursive function (respect to the positions) instead of some fixed number, then the corresponding real numbers are called *binary ω-enumerable*.

It is worth noting that the binary k-enumerability defined above is not symmetric with respect to "0" and "1" positions. Our definition corresponds to the (k-)enumerability of binary set A of the real number x_A. That is, x_A is binary k-enumerable iff A is k-r.e. (See Definition 1). Thus, the binary enumerability is not closed under the operation of "$-$" which similar to that the k-r.e.ness of subsets of $\mathbb{N}$ is not closed under the set operations of complement. In fact, our main result of this paper shows that there are binary enumerable real numbers x and y such that $x - y$ is not binary ω-enumerable.

W.l.o.g., we consider only the real numbers in the interval $[0; 2]$ in this paper. For other real numbers, say y, outside this interval, there is a natural number n and an $x \in [0; 1]$ such that $y = n + x$. Intuitively x and y have completely same type of computability in any reasonable sense.

We conclude this section by introducing some notations. Let $\Sigma := \{0, 1\}$ be an alphabet and $\mathbb{N}$ be the set of natural numbers. Denote by Σ^* and Σ^ω the set of binary strings and the set of infinite sequences over Σ, respectively. Strings are denoted by lower case letters u, v, w. The concatenation of two strings x and y is denoted by xy; $|u|$ denotes the length of the string u; λ is the empty string which has the length 0; $<$ is the length-lexicographical ordering on Σ^*. The ith bit of the string u is denoted by $u(i)$, so $u := u(0)u(1)\cdots u(|x|-1)$. For any set $A \subseteq \mathbb{N}$, its characteristic sequence is also denoted by A. For $A \subseteq \mathbb{N}$ and $n \in \mathbb{N}$, let $A \upharpoonright n$ denote the finite initial segment of A below n, i.e. $A \upharpoonright n := \{i : i < n \;\&\; i \in A\}$. We identify this initial segment whith its characteristic string, i.e. $A \upharpoonright n = A(0)A(1)\cdots A(n-1) \in \Sigma^*$. Similarly, we define the initial segment of $u \in \Sigma^*$ below $n \in \mathbb{N}$ by $u \upharpoonright n := u(0)u(1)\cdots u(n-1)$ if $n \leq |u|$ and $u \upharpoonright n := u$ otherwise. If $u = A \upharpoonright n$ for some $n \in \mathbb{N}$, then we denote that $u \sqsubset A$. Thus the length-lexicographical order on Σ^* can be extended to $\Sigma^* \cup \Sigma^\omega$ by $u < A \iff \exists v\, (v \sqsubset A \;\&\; u < v)$ and $A < u \iff \exists v\, (v \sqsubset A \;\&\; v < u)$ for any $x \in \Sigma^*$ and $A \in \Sigma^\omega$. For $A \subseteq \mathbb{N}$, we define A_L to be the set of all finite binary strings which are "left" the infinite binary sequence A, i.e. $A_L := \{u \in \Sigma^* : u < A\}$.

2 Ershov Hierarchy and Weak Computable Real Numbers

In this section we will recall at first the definition of Ershov Hierarchy on subsets of natural numbers and discuss some relationships between this hierarchy and weak computability of real numbers.

Definition 1 (Putnam, [7] Gold [4] and Ershov [3]).

1. Let $h : \mathbb{N} \to \mathbb{N}$ be any function. A set $A \subseteq \mathbb{N}$ is called *h-r.e.*, if there is a computable sequence $(A_s)_{s\in\mathbb{N}}$ of finite subsets of $\mathbb{N}$ such that
   (a) $A_0 = \emptyset$;
   (b) $A = \lim\limits_{s\to\infty} A_s := \bigcup\limits_{n=0}^{\infty} \bigcap\limits_{s=n}^{\infty} A_s$; and
   (c) $\forall n \in \mathbb{N}(|\{s \in \mathbb{N} : n \in A_{s+1} \triangle A_s\}| \leq h(n))$.
   where Δ is defined by $B\Delta C := (B\backslash C) \cup (C\backslash B)$. The sequence $(A_s)_{s\in\mathbb{N}}$ is called an *effective h-enumeration* of A.
2. Set $A \subseteq \mathbb{N}$ is called ω-r.e., if A is h-r.e. for some recursive function h. In this case, the sequence $(A_s)_{s\in\mathbb{N}}$ is called an *effective ω-enumeration* of A and h is a *bounding function* of this enumeration.
3. If h is a constant function with $h(x) := k$, then the h-r.e. set $A \subseteq \mathbb{N}$ is called *k-r.e.* The sequence $(A_s)_{s\in\mathbb{N}}$ is called an *effective k-enumeration* of set A.

So, in particular, the empty set is the unique 0-r.e. set and the 1-r.e. sets are the r.e. sets. The 2-r.e. are usually called *d-r.e.* as they are the differences of the r.e. sets, i.e., A is d-r.e., iff $A = B\backslash C$ for some r.e. sets B and C. Similarly, A is $(k+1)$-r.e., iff there are r.e. set B and k-r.e. set C such that $B\backslash C$. A Turing degree is called *(k-) r.e.* if it contains at least one (k-) r.e. set.

Definition 2. For any $\alpha \in \mathbb{N}^+$ or $\alpha = \omega$, a real number x is called *binary α-enumerable*, if there is an integer n and an α-r.e. set A such that $x = n + x_A$. The class of all binary α-enumerable real numbers is denoted by $\mathbb{B}_\alpha$

Weihrauch and Zheng [13] call a real number x *left (right), computable* if there is an increasing (decreasing) sequence of rational numbers which converges to x. Left and right computable real numbers are called *semi-computable.* The class of semi-computable real numbers is denoted by $\mathbb{C}_1$.

Theorem 1 (Weihrauch and Zheng [13]). *For any $k \in \mathbb{N}$.*

1. *There is a real number $x \in \mathbb{B}_{k+1}\backslash\mathbb{B}_k$ such that x is left computable.*
2. *There is a real number $x \in \mathbb{B}_\omega$ with $x \notin \mathbb{B}_k$ for any $k \in \mathbb{N}$ such that x is left computable.*
3. *There is real number $x \in \mathbb{B}_2$ such that x is not left computable.*

By Theorem 1, a left computable real number can have a $(k+1)$-r.e. binary set which is not k-r.e. or even have an ω-r.e. binary set which is not k-r.e. for any $k \in \mathbb{N}$. But next theorem shows that they must *tt*-equivalent to some r.e. set.

Theorem 2 (Ambos-Spies [1], Jockusch [5]). *For any set* $A \subseteq \mathbb{N}$, x_A *is left computable, if and only if* A_L *is r.e. Therefore, if* x_A *is semi-computable, then* A *has an r.e. tt-degree.*

The converse of Theorem 2 is not true because it is shown in [13] (Theorem 5) that there are r.e. sets $B, C \subseteq \mathbb{N}$ such that $B \oplus \overline{C}$, which has an r.e. *tt*-degree, has a non-semi-computable binary real number.

Theorem 3. *A real number* x_A *is semi-computable, if and only if* A_L *is* k*-r.e. for some* $k \in \mathbb{N}$.

Proof. The part of "⇒" is obvious. We prove here the part of "⇐" and only for the case of $k = 2$. For other $k > 2$, the idea is similar.

Let $A_L = B \backslash C$ and $B, C \subseteq \Sigma^*$ are r.e. sets. If C has a least element u_0, then $A_L = B \cap (\Sigma^* \upharpoonright u_0) := \{v \in \Sigma^* : v < u_0 \& v \in B\}$ is an r.e. set. This implies that x_A is left computable.

Suppose that C has no least element. We can define a computable sequence $\{u_s\}_{s \in \mathbb{N}}$ of finite strings such that $u_s \in C$ and $u_{s+1} < u_s$ for any $s \in \mathbb{N}$. That is, $\{u_s\}_{s \in \mathbb{N}}$ is a decreasing sequence which has the limit, say, $x_{A'} =: \lim_{s \to \infty} u_s$. If $A = A'$, then A is right computable. Suppose that $A \neq A'$, hence that $x_A < x_{A'}$. There is an $n \in \mathbb{N}$ such that $x_{A'} - x_A > 2^{-n}$.

If $A'(m) = 0$ for almost all $m \in \mathbb{N}$, hence $(\forall n > m_0)\,(A'(m) = 0)$ for some $m_0 \in \mathbb{N}$, then $u \geq (A' \upharpoonright m_0)$ for all $u \in C$. This implies that A_L is r.e. again.

Suppose that there are infinite many m such that $A'(m) = 1$. Choose an $m > n$ such that $A'(m) = 1$, let $v =: (A' \upharpoonright m)0$. Then we have $x_A < v < x_{A'}$ since $(x_{A'} - v) \leq 2^{-n}$. It follows that $A_L = B \cap (\Sigma^* \upharpoonright v)$ is an r.e. set again. Thus x_A is left-computable.

As an natural extension of semi-computable real numbers, Weihrauch and Zheng [13] introduced the class of weakly computable real numbers.

Definition 3. A real number x is called *weakly computable* (w.c. for short), if there are two semi-computable real numbers y, z such that $x = y + z$. The set of all weakly computable real numbers is denoted by $\mathbb{C}_2$.

For example, if $A := B \setminus C$ is a d-r.e. set with r.e. sets B and C, then $B \cup C$ is also r.e. set. So x_A is weakly computable because $x_A = x_{B \cup C} - x_C$. More generally, we can show by an easy induction on k, that if A is a k-r.e set, $k \geq 1$ then x_A is a weakly computable real number. Furthermore, The class $\mathbb{C}_2$ is a closed field generated by $\mathbb{C}_1$ as shown in [13]. The class of weakly computable reals has another characterization as follows.

Theorem 4 (Weihrauch and Zheng [13]). *A real number* x *is weakly computable, iff there is a computable sequence* $(x_n)_{n \in \mathbb{N}}$ *of rational numbers which converges to* x *weakly effectively, i.e., the sum of its jumps* $\sum_{n=0}^{\infty} |x_{n+1} - x_n|$ *is bounded.*

It follows immediately from Theorem 1 that the class of w.c. real numbers extends the class of semi computable real numbers properly, since x_A is w.c. if A is d-r.e. This implies also that there are two semi-computable real numbers such that their sum is not semi-computable.

Now we will prove a result about Turing degrees. By Definition, the recursive Turing degree $\mathbf{0}$ consists of only recursive sets. We will show that any nonrecursive r.e. degree contains a set whose binary real number is w.c. but not semi-computable. We will also show that the binary sets of weakly computable real numbers do not exhaust all ω-r.e. sets. These results follow immediately flowing observation of Ambos-Spies [1].

Theorem 5 (Ambos-Spies [1]).

1. *If B, C are r.e. sets such that $B|_T C$, then $x_{B\oplus\overline{C}}$ is not semi-computable.*
2. *Assume that $x_{A\oplus\emptyset}$ is weakly computable. Then A is f-r.e. for $f(n) = 2^{3n}$.*

Theorem 6. *For any nonrecursive r.e. Turing degree $\mathbf{a}$, there is a set $A \in \mathbf{a}$ such that x_A is weakly computable but not semi-computable.*

Proof. Let $\mathbf{a}$ be a nonrecursive r.e. Turing degree. By Sacks' Splitting Theorem [8] there exist two incomparable r.e. degrees $\mathbf{b_0}, \mathbf{b_1}$ such that $\mathbf{a} = \mathbf{b_0} \vee \mathbf{b_1}$. Choose r.e. sets $B_0 \in \mathbf{b_0}$ and $B_1 \in \mathbf{b_1}$ and define set $A =: B_0 \oplus \overline{B_1}$. Obviously, A is a d-r.e set. So x_A is weakly computable. On the other hand, it is not semi-computable by (1) of Theorem 5, since $B_0|_T B_1$.

Theorem 7. *There exists a binary ω-enumerable real number x such that x is not weakly computable, i.e., $\mathbb{B}_\omega \subsetneq \mathbb{C}_2$.*

Proof. Let $f(n) =: 2^{3n}$ and $g(n) =: 2^{4n}$. Then $g(n) > f(n)$ for all $n \in \mathbb{N}$. By a result of Ershov [3], there is a g-r.e. set A' which is not f-r.e. Let $A =: A' \oplus \emptyset$. By (2) of the Theorem 5, x_A is not weakly computable. But A', hence A is ω-r.e. since g is a recursive function.

3 More Complicated Weakly Computable real numbers

In this paper, we will construct two binary enumerable real numbers x and y such that their difference $z := x - y$ is not binary ω-enumerable. In fact we prove even more that the binary set of z has no ω-r.e. Turing degree. Our proof here is a sophisticated finite injury priority construction. In the proof, we will use the following three kinds of the restriction of a subset of $\mathbb{N}$:

$$A \upharpoonright n := \{x \in A : x < n\};$$
$$A \downharpoonright n := \{x \in A : x > n\};$$
$$A \upharpoonright (n; m) := \{x \in A : n < x < m\}.$$

We show at first a technical lemma whose proof is straightforward.

Lemma 1. *Let $A, B, C \subset \mathbb{N}$ be finite sets such that $x_A = x_B - x_C$ and n, m and y be any natural numbers. Then the following hold.*

1. $\max A \leq \max(B \cup C)$;
2. *If* $B \downharpoonright n = C \downharpoonright n$, *then* $\max A \leq n$ *and* $n \in A \iff n \in B \backslash C$,
3. *If* $n, m \in B \backslash C$, $n < y < m$ *and* $(B \upharpoonright (n; m)) \backslash \{y\} = (C \upharpoonright (n; m)) \backslash \{y\}$, *then* $n \notin A \iff y \in C \backslash B$;
4. *If* $x_{A_1} = x_{B_1} - x_{C_1}$, $(B \cup C) \upharpoonright (n+1) = (B_1 \cup C_1) \upharpoonright (n+1)$ *and* $n \in B \backslash C$, *then* $A \upharpoonright n = A_1 \upharpoonright n$.
5. *Suppose that* $m < y < n$, $n \in A$.
   - (5.a) *If* $m \in A$, $x_{A'} = x_A - 2^{-y}$ *and* $A \upharpoonright (m; n) = \emptyset$, *then* $A' := (A \backslash \{m\}) \cup \{m+1, m+1, \ldots, y\}$.
   - (5.b) *If* $m \notin A$, $x_{A'} = x_A + 2^{-y}$ *and* $A \upharpoonright (m; n) = \{m+1, m+2, \ldots, y\}$, *then* $A' := (A \cup \{m\}) \backslash \{m+1, \ldots, y\}$.

For the proof of our next result, we recall some standard recursion-theoretical notations. (More detail explanations of these notations can be found in, e.g., [10, 6].) Let $(M_e^A : e \in \mathbb{N})$ be an effective enumeration of all (type one) Turing machines (with oracle A) and $(\varphi_e^A : e \in \mathbb{N})$ be the corresponding effective enumeration of all A-computable functions from $\mathbb{N}$ to $\mathbb{N}$ such that φ_e^A is computed by the e-th Turing machine M_e^A with oracle A. $\varphi_{e,s}^A$ is the s-th approximation of φ_e^A. The use-function $u_{e,s}^A(x)$ of the computation $M_e^A(x)$ is defined to be the length of the initial segment of the oracle which is really used in the computation $u_{e,s}^A(x)$. It is important to note that, if $u_{e,s}^A(x) = t$, and $A \upharpoonright t = B \upharpoonright t$, then the computations of $M_e^A(x)$ and $M_e^B(x)$ are completely the same. As usual Turing machine model, we assume always that $u_{e,s}^A(x) \leq s$. This is a very useful estimation because we need only to preserve simply the initial segment $A \upharpoonright s$ so that the computation $M_{e,s}^A(x)$ will not be destroyed. Thus, any change of the membership of the elements $a \geq s$ to A do not destroy this computation. As usual, we always assume that the use-function is nondecreasing about s. As the partial recursive functionals, $M_e^A(x)$ is often denoted by uppercase Greek letters $\mathit{\Gamma}$, $\mathit{\Delta}$, $\mathit{\Lambda}$, etc. and the corresponding lowercase Greek letters γ, δ, λ are their use-functions, respectively. To simplify the notation, instead of pointing out the subscript s, we will often use the expression like $\mathit{\Gamma}^W(x) \upharpoonright \gamma(x)[s]$ to denote the current value of this expression $\mathit{\Gamma}^W(x) \upharpoonright \gamma(x)$ at the stage s. Then we have, e.g., $A_s = A[s]$, $B_s(x_s) = B(x)[s]$, etc.

As usual, superscript A is always omitted if $A = \emptyset$. We define also $W_e := \mathrm{dom}(\varphi_e)$ and $W_{e,s} := \mathrm{dom}(\varphi_{e,s})$. Then $(W_e : e \in \mathbb{N})$ enumerates all r.e. subsets of $\mathbb{N}$ and $(W_{e,s} : s \in \mathbb{N})$ is a recursive enumeration of W_e.

We usually do not distinguish between a subset of $\mathbb{N}$ and its characteristic function. That is, we have always that $x \in A \iff A(x) = 1$ and $x \notin A \iff A(x) = 0$.

Theorem 8. *There are r.e. sets B and C such that the set A which satisfies $x_A = x_B - x_C$ is not of ω-r.e. Turing degree.*

Proof. (Sketch) We will construct effectively the computable sequences $(A_s : s \in \mathbb{N})$, $(B_s : s \in \mathbb{N})$ and $(C_s : s \in \mathbb{N})$ of finite subsets of $\mathbb{N}$ which satisfy the following conditions:

1. $x_{A_n} = x_{B_n} - x_{C_n}$ for all $n \in \mathbb{N}$;
2. The limits $A := \lim_{n\to\infty} A_n$, $B := \lim_{n\to\infty} B_n$ and $C := \lim_{n\to\infty} C_n$ exist and $x_A = x_B - x_C$ holds;
3. $(B_s : s \in \mathbb{N})$ and $(C_s : s \in \mathbb{N})$ are the recursive enumerations of sets B and C, respectively. Hence B and C are r.e. sets and x_A is weakly computable.
4. The degree $\deg_T(A)$ is not ω-r.e.

The first condition is easy to satisfy. At any stage $s+1$, we will define directly the finite sets B_{s+1} and C_{s+1} in such a way that $B_s \subseteq B_{s+1}$ and $C_s \subseteq C_{s+1}$ and let A_{s+1} simply to be the unique finite set satisfying $x_{A_{s+1}} = x_{B_{s+1}} - x_{C_{s+1}}$. This satisfy automatically the second and third conditions too.

Now we can discuss the fourth condition. The degree $\deg_T(A)$ is not ω-r.e. means that A is not Turing equivalent to any ω-r.e. set. Then, for meeting the fourth condition, it suffices to make sure that, for all ω-r.e. set V (with the effective ω-enumeration $(V_s : s \in \mathbb{N})$ and corresponding recursive bounding function h), partial recursive functionals Γ and Δ, the following requirements are satisfied:

$$R_{V,\Gamma,\Delta} : A \neq \Gamma^V \text{ or } V \neq \Delta^A.$$

Since the classes of all ω-r.e. sets and all partial recursive functionals are effectively enumerable (see [2, 10]), respectively. So these requirements can be effectively enumerated too. We fix $(R_e : e \in \mathbb{N})$ as one of such kind of effective enumeration. Thus all requirements can be given different priorities according to this enumeration, i.e., R_i is of the higher priority than R_j if and only if $i < j$.

The strategy for satisfying a single requirement $R_{V,\Gamma,\Delta}$ is as follows:

First phase: At first we choose arbitrarily a witness $x \in \mathbb{N}$. Then wait for the stage s such that:

$$A_s(x) = \Gamma^V(x)[s] \ \& \ V \upharpoonright \gamma(x)[s] = \Delta^A \upharpoonright \gamma(x)[s]. \tag{1}$$

(if this never happens, then x is a witness to the success of $R_{V,\Gamma,\Delta}$.) Define $B_{s+1} := B_s \cup \{x\}$, $C_{s+1} := C_s$. It can be shown that $A_{s+1} = A_s \cup \{x\}$. In a later stage, we might hope to remove x from A when it is necessary. Since B should be a r.e. set, we can not simply remove it by taking out x from B. Instead, we achieve that by putting some bigger element to C. Besides, we hope to preserve the computations of $\Delta^A \upharpoonright \gamma(x)[s]$, so we define here a supplementary element y big enough so that $y > \max\{\delta\gamma(x)[s], x\}$. Then into next phase.

Second phase: Wait for some new stage $s' > s$ at which the following hold:

$$A_{s'}(x) = \Gamma^{V\upharpoonright\gamma(x)}(x)[s'] \ \& \ V \upharpoonright \gamma(x)[s'] = \Delta^{A\upharpoonright\delta\gamma(x)} \upharpoonright \gamma(x)[s']. \tag{2}$$

(if this never happens, then x is also a witness to the success of $R_{V,\Gamma,\Delta}$.) In this case, we hope to remove x from A to force the initial segment $V \upharpoonright \gamma(x)[s']$ to be changed whenever this condition is satisfied later again. This can be achieved by putting y into C, i.e., define $B_{s'+1} := B_{s'}$, $C_{s'+1} := C_{s'} \cup \{y\}$. It can be shown that $x \notin A_{s'+1}$. Then into next phase.

Third phase: Wait for some new stage $s'' > s'$ at which the following hold:

$$A_{s''}(x) = \Gamma^{V \upharpoonright \gamma(x)}(x)[s''] \;\&\; V \upharpoonright \gamma(x)[s''] = \Delta^{A \upharpoonright \delta\gamma(x)} \upharpoonright \gamma(x)[s'']. \tag{3}$$

(if this never happens, then x is a witness to the success of $R_{V,\Gamma,\Delta}$ again.) Define $B_{s''+1} := B_{s''} \cup \{y\}$, $C_{s''+1} := C_{s''}$ and let $A_{s''+1}$ similar as before. It can be shown that $x \in A_{s''+1}$ again. Now the supplementary element y is used two times and can not be used again. Then choose a new one $y := y + 1$ and go to the second phase.

Because the action in any of above phases changes the membership of x to A, whenever we go from phase X to another phase Y, the value of $A(x)$, hence also the initial segment $V \upharpoonright \gamma(x)$ must be changed. But if we go from phase Y back to phase X later, $A \upharpoonright \delta\gamma(x)$ is recovered to that of last appearance of phase X. So, the initial segment $\Delta^{A \upharpoonright \delta\gamma(x)} \upharpoonright \gamma(x)$, hence also the initial segment $V \upharpoonright \gamma(x)$, is recovered. Because of the ω-r.e.ness of the set V, this can happen at most finitely often. Therefore, after some stages, (2) or (3) will never hold again. Thus the requirement $R_{V,\Gamma,\Delta}$ is finally satisfied by the witness x.

To satisfy all the requirements, we apply the finite injury priority construction. A witness x_e and a supplementary element y_e ($> x_e$) are appointed to the requirement R_e at any stage. Instead of trying to satisfy all requirements in the given order we will attack those requirements first to which the above strategy can be applied. This action should be preserved from injury by lower priority requirements. Therefore whenever the actions of any above phases are made for the requirement R_e, all requirements R_i with $i > e$ will be *initialized*. That is, we redefine the witnesses x_i and supplementary elements y_i of R_i (for $i > e$) bigger than all elements enumerated into B or C so far and also bigger than the $\delta\gamma(x_e)$ to preserve the computations of $\Delta^{A \upharpoonright \delta\gamma(x)} \upharpoonright \gamma(x_e)$ from the injury by lower priority requirements. Still it is not enough, because the witness $x_e[s]$ may be removered from A or put into A not only by putting y_e into C or B as we hoped, but also by putting y_i in to C or B through the actions for R_i with some $i > e$. Then the condition (2) or (3) can be satisfied again without any changes of $V \upharpoonright \gamma(x_e)$, and hence in such a way R_e can requires attentions infinitely often. To avoid this bad situation, we will build a “firewall” between the all supplementary elements y_e's of R_e and the x_i's, y_i's for R_i with $i > e$. Namely, we put another element z_e, which satisfies that $\forall i > e\,(y_e < z_e < x_i, y_i)$, into $B \backslash C$ as well. By (4) of the Lemma 1, it succeeds. Therefore, at any stage s, we will define a witness x_e and two supplementary elements y_e, z_e such that $x_e < y_e < z_e$ and $\forall i > e\,(z_e < x_i)$.

Of course, a requirement could be injured by some higher priority requirements. Since finite many attacks suffices for any given requirement as we saw in the above strategy, every requirement R_e can be injured at most finitely often because there are only finite many requirements which have the higher priority than R_e. So every requirement have enough chances to be satisfied by the above strategy.

We give the exact construction as follows:

Stage 0: Define $A_0 = B_0 = C_0 := \emptyset$. For any $e \in \mathbb{N}$, we define $x_e[0] := 3e + 1$, $y_e[0] := 3e + 2$ and $z_e[0] := 3e + 3$. All requirements R_e are said to be *initialized* at this stage.

Stage $s+1$: Given $A_s, B_s, C_s, x_e[s], y_e[s]$ and $z_e[s]$ for all $e \in \mathbb{N}$. A requirement $R_e (= R_{V,\Gamma,\Delta})$, for some ω-r.e. set V, and partial recursive functionals Γ and Δ, *requires attention* if the following condition holds:

$$A(x_e)[s] = \Gamma^{V \upharpoonright \gamma(x_e)}(x_e)[s] \ \& \ V \upharpoonright \gamma(x_e)[s] = \Delta^{A \upharpoonright \delta\gamma(x_e)} \upharpoonright \gamma(x_e)[s]. \tag{4}$$

Choose the minimal $e \le s$, if any, such that the requirement $R_e (= R_{V,\Gamma,\Lambda})$ requires attention at this stage. Then do the following:

In all the following cases we will always define the set A_{s+1} as the unique finite subset of $\mathbb{N}$ which satisfies the condition that $x_{A_{s+1}} = x_{B_{s+1}} - x_{C_{s+1}}$. We define also $x_e[s + 1] := x_e[s]$.

Case 1. $x_e[s] \notin B_s$. Let $g(x_e)[s] := \sum_{i=0}^{\gamma(x_e)[s]} h(i)$. Then we define in this case all the following:

$$\begin{cases} y_e[s + 1] := \max\{\delta\gamma(x_e)[s], \max(B_s \cup C_s)\} + 1; \\ z_e[s + 1] := y_e[s + 1] + g(x_e)[s] + 1; \\ x_i[s + 1] := z_e[s + 1] + 3i + 1, \text{ for all } i > e; \\ y_i[s + 1] := z_e[s + 1] + 3i + 2, \text{ for all } i > e; \\ z_i[s + 1] := z_e[s + 1] + 3i + 3, \text{ for all } i > e; \\ B_{s+1} := B_s \cup \{x_e[s], z_e[s + 1]\} \cup (C_s \downharpoonright x_e[s]); \\ C_{s+1} := C_s \cup (B_s \downharpoonright x_e[s]). \end{cases}$$

Notice that we have that $x_e[s], z_e[s+1] \in B_{s+1}$, $x_e[s] \in A_{s+1}$, by (2) of Lemma 1, and that

$$\begin{aligned} &x_e[s] = x_e[s + 1] < \max\{\delta\gamma(x_e)[s], \max(B_s \cup C_s)\} < y_e[s + 1] \\ &< y_e[s + 1] + g(x_e)[s] < z_e[s + 1] < x_i[s + 1] < y_i[s + 1] < z_i[s + 1], \end{aligned}$$

for all $i > e$.

Case 2. $x_e[s] \in B_s$. There are still four possibilities as follows:

Case 2.1. $x_e[s] \in A_s$ & $y_e[s] \notin C_s$. This means that the supplementary element $y_e[s]$ is not yet used and hence not in $(B \cup C)[s]$. Then define

$$\begin{cases} B_{s+1} := B_s \ \& \ C_{s+1} := C_s \cup \{y_e[s]\}; \\ y_e[s + 1] := y_e[s] \ \& \ z_e[s + 1] := z_e[s]. \end{cases}$$

Case 2.2. $x_e[s] \in A_s$ & $y_e[s] \in C_s$. Here $x_e[s] \in A_s$ means that if a supplementary element y_e is put into C, then it must be put into B too. In this situation, a new supplementary element is defined which is still not in C (see case 2.3.) So this case can not happen in fact.

Case 2.3. $x_e[s] \notin A_s$ & $y_e[s] \in C_s$. In this case, $y_e[s]$ is already put into C but not yet enumerated into B. We need only put $y_e[s]$ into B and define a new y_e. That is, we define

$$\begin{cases} B_{s+1} := B_s \cup \{y_e[s]\} \ \& \ C_{s+1} := C_s; \\ y_e[s + 1] := y_e[s] + 1 \ \& \ z_e[s + 1] := z_e[s]. \end{cases}$$

Case 2.4. $x_e[s] \notin A_s$ & $y_e[s] \notin C_s$. This can also never happen in fact.

In all of these cases, we define $x_i[s+1] := x_i[s]$; $z_i[s+1] := z_i[s]$ and $y_i[s+1] := y_i[s]$ for all $i < e$. Then *initialize* all lower priority requirements R_i (for $i > e$) by defining

$$x_i[s+1] := 3w[s]+1; y_i[s+1] := 3w[s]+2 \text{ and } z_i[s+1] := 3w[s]+3 \qquad (5)$$

where $w[s] := \max(B_{s+1} \cup C_{s+1})$. If the requirements R_i $(i > e)$ has received attention before stage $s+1$ and was not yet been initialized thereafter, then it is called that R_i is *injured* at this stage by the requirement R_e. We say that the requirement R_e *receives attention* at this stage.

If no requirements require attention at this stage, then every thing remains unchanged and go directly to the next stage.

This ends the construction. We can show that the sets $A := \lim_{s\to\infty} A_s$, $B := \lim_{s\to\infty} B_s$ and $C := \lim_{s\to\infty} C_s$ satisfy the theorem.

References

1. K. Ambos-Spies. A note on recursively approximatable real numbers, *Forschungsberichte Mathematische Logik,* Mathematisches Institut, Universität Heidelberg. Nr. 38, 1998.
2. M. M. Arslanov. Degree structures in local degree theory, in *Complexity, Logic, and Recursion Theory,* Andrea Sorbi (ed.), *Lecture Notes in Pure and Applied Mathematics*, Vol. 187, Marcel Dekker, New York, Basel, Hong Kong, 1997, pp. 49–74.
3. Y. Ershov. On a hierarchy of sets I; II and III, *Algebra i Logika*, 7(1968), no. 1, 47 – 73; 7(1968), no. 4, 15 – 47 and 9(1970), no. 1, 34 – 51.
4. E. M. Gold. Limiting recursion *J. of Symbolic Logic,* 30(1965), 28–48.
5. C. G. Jockusch Semirecursive sets and positive reducibility. Trans. Amer. Math. Soc. 131(1968), 420–436.
6. P. Odifreddi. *Classical Recursion Theory*, vol. 129 of Studies in Logic and the Foundations of Mathematics, North-Holland, Amsterdan, 1989.
7. H. Putnam. Trial and error predicates and solution to a problem of Mostowski, *J. of Symbolic Logic,* 30(1965), 49–57.
8. G. E. Sacks On the degrees less than $\mathbf{0}'$, *Ann. of Math.* 77(1963), 211–231.
9. R. Soare Recursion theory and Dedekind cuts, *Trans, Amer. Math. Soc.* 140(1969), 271–294.
10. R. Soare. *Recursively Enumerable Sets and Degrees*, Springer-Verlag, Berlin, Heidelberg, 1987.
11. K. Weihrauch. *Computability.* EATCS Monographs on Theoretical Computer Science Vol. 9, Springer-Verlag, Berlin, Heidelberg, 1987.
12. K. Weihrauch. *An Introduction to Computable Analysis.* (In Preparation.)
13. K. Weihrauch and X. Zheng. A finite hierarchy of recursively enumerable real numbers, MFCS'98, Brno, Czech Republic, August 24 – 28, 1998.