MAXIMAL PAIRS OF C.E. REALS IN THE COMPUTABLY LIPSCHITZ DEGREES

YUN FAN AND LIANG YU

Abstract. Computably Lipschitz reducibilty (noted as $\leq_{cl}$ for short), namely computations where the use on the oracle on argument x is $x+c$ for some constant c, was suggested as a measure of relative randomness. In this paper, we introduce the definition of *cl-maximal pair of c.e. reals*. We prove that for any non-computable Δ^0_2 real, there exists a c.e. real so that no c.e. real can *cl*-compute both of them. Thus, each non-computable c.e. real is the half of a *cl*-maximal pair of c.e. reals.

1. Introduction

In randomness and incomputability we have two fundamental measures: plain complexity C and prefix-free complexity K. When we look at reals, especially c.e. reals under notions of relative randomness, we would like to understand C and K as "reducibilities". For instance, where for $E = K$ or C, we have $\alpha \leq_E \beta$ iff $E(\alpha \upharpoonright n) \leq E(\beta \upharpoonright n) + O(1)$.

There are a number of natural reducibilities which imply $\leq_C$ and $\leq_K$ reducibilities. One is Solovay reducibility [9], which is a powerful tool in studying relative randomness of c.e. reals. But Solovay reducibility presents various shortcomings outside this class [6]. Another reducibility, *strong weak truth table* reducibility, was suggested by Downey, Hirschfeldt and LaForte [7, 6] as an alternative for Solovay reducibility. This reducibility has appeared in the literature with various names, e.g. *computably Lipschitz* reducibility (due to a characterization of it in terms of effective Lipschtz functions) [2] and *linear* reducibility [4] . We here adopt the terminology in [2] and note it as $\leq_{cl}$ for short.

Definition 1.1. *Given two reals α and β in the unit, $\alpha \leq_{cl} \beta$ if there is a Turing functional Γ and a constant c such that $\alpha = \Gamma^\beta$ and the use of Γ on any argument x is bounded by $x + c$. The Turing functionals which have their use restricted in such a way are called cl-functionals.*

To make the technical details of the proofs slightly simpler, we often work an even restrictive reducibility than computable Lipschitz: *identity bounded Turing* reducibility (*ibT* or $\leq_{ibT}$ for short) is a computable Lipschitz reduction for which the constant c is 0. This reducibility was introduced by Soare in connection with applications of computability theory to differential geometry (see [11, 3]).

The first author is supported by DFG (446 CHV 113/240/0-1) and NSFC (10420130638). Both authors are supported by NSFC(10701041) and Research Fund for the Doctoral Program of Higher Education (20070284043).

The main justification for $\leq_{cl}$ as a measure of relative randomness is the following proposition.

Proposition 1.2 (Downey,Hischfeldt and Lafort [6])**.** *If $\alpha \leq_{cl} \beta$ are c.e. reals, then for all n, $K(\alpha \upharpoonright n) \leq K(\beta \upharpoonright n) + O(1)$.*

Having defined *cl*-reducibility we can consider the structure of the induced degrees. The *cl*-degrees are the equivalence classes under *cl*-reducibility. Notice that they either contain *only random* reals or only *non-random* reals. Since *cl*-reducibility is a strengthening of weak truth table reducibility, the structure of *cl*-degrees is interesting from the computation complexity.

The structure of *cl*-degrees of c.e. reals presents some difficulties, which is neither a lower semi-lattice, nor an upper semi-lattice [6]. The lack of join operator of *cl*-degrees of c.e. reals was first proven directly by Downey, LaForte,and Hirschfeldt [7], but follows from the next theorem given by Yu and Ding[12].

Theorem 1.3 (Yu and Ding [12])**.** *There is no cl-complete c.e. real.*

Actually, Yu and Ding proved something stronger:

Corollary 1.4. *There are two c.e.reals α and β so that no c.e.real γ satisfies $\alpha \leq_{cl} \gamma$ and $\beta \leq_{cl} \gamma$.*

Moreover, real α in Corollary 1.4 can be restricted as a c.e. set by Fan [8]. Discussed with Ambos-spies and Merkle, we introduce the definition of *cl-maximal pair of c.e. reals.*

Definition 1.5. *A pair (α, β) of c.e. reals is a cl-maximal pair of c.e. reals if no c.e. real γ can cl-compute both of them.*

By this definition, Corollary 1.4 can be expressed as follows: there exists a *cl*-maximal pair of c.e. reals. Moreover, the half of a *cl*-maximal pair can be a c.e. set [8]. In [5], Barmpalias, Downey and Greenberg characterized array computability by *cl*-maximal pairs of c.e. reals.

Proposition 1.6 (Barmpalias, Downey and Greenberg[5])**.** *A c.e. degree $\boldsymbol{a}$ is array non-computable iff there is a cl-maximal pair of c.e. reals in $\boldsymbol{a}$.*

In this paper we investigate *cl*-maximal pairs of c.e. reals. Trivially, no computable real is a half of a *cl*-maximal pair of c.e. reals. Is there a non-computable c.e. real which is not half of any *cl*-maximal pair of c.e. reals? We give a negative answer.

Theorem 1.7. *Given a non-computable Δ^0_2 real α, there is a c.e. real β so that no c.e. real can cl-compute both of them. So every c.e. real is the half of a cl-maximal pair of c.e. reals.*

We present some applications of Theorem 1.7 to the Kolmogorov complexity theory.

Corollary 1.8. *There is a c.e. real β so that $\overline{\lim}_n K(\beta \upharpoonright n) - n = +\infty$ and $\underline{\lim}_n K(\beta \upharpoonright n) - K(n) < +\infty$.*

Proof. Recall the result by Stephan (see [6]) that for any c.e. reals α and β, $\underline{\lim}_n K(\alpha \upharpoonright n) - K(\beta \upharpoonright n) = +\infty$ implies $\alpha \leq_{cl} \beta$. Let α be a non-computable K-trivial real, choose a c.e. real β as described by Theorem 1.7. Since α is K-trivial and $\alpha \not\leq_{cl} \beta$, we have $\underline{\lim}_n K(\beta \upharpoonright n) - K(n) < +\infty$ by Stephan's result. Moreover, $\overline{\lim}_n K(\beta \upharpoonright n) - n = +\infty$, since every c.e. random real γ has the property that $\lim_n K(\gamma \upharpoonright n) - n = +\infty$. □

Corollary 1.9. *There is a c.e. real β for which if $\gamma \geq_{cl} \beta$ is a c.e. real, then $\underline{\lim}_n K(\gamma \upharpoonright n) - K(n) < +\infty$.*

Proof. Let α and β be the reals in Corollary 1.8. Suppose that $\gamma \geq_{cl} \beta$ is a c.e. real, then $\gamma \not\geq_{cl} \alpha$. By Stephan's result again, $\underline{\lim}_n K(\gamma \upharpoonright n) - K(n) < +\infty$. □

Note that Corollary 1.9 can be viewed as a strength of the main result in [2] and a "localization" of Hirschfeldt's result that there is a real which is not *cl*-reducible to any complex real.

We assume some background on computability theory and some knowledge of standard conventions, most of which can be found in Soare [10]; knowledge of algorithmic randomness is also helpful. Notations about the binary expansion of reals are useful.

Notation 1.10. The digits on the right of the decimal point in the binary expansion are numbered as $1, 2, 3, \cdots$ from left to right. The digits on the left of the decimal point are numbered as $0, -1, -2, \cdots$. "The digits lower (higher) than b" is written as "$\geq$ ($\leq$) b", and "the digits between h and l (where $h < l$)" is written as an interval " $[h, l]$". Symbol b^k is a string with k consecutive numbers b, where $b \in \{0, 1\}$. Symbol $\mathcal{A}$ denotes a set of numbers for which the lowest (rightmost) non-zero digit is digit 1, i.e. the binary expansion of such number is $b_{-i} \cdots b_0.b_1$ ($b_{-i} \neq 0, i \in \mathbb{N}$).

2. The Proof idea of Theorem 1.7

Given a non-computable Δ^0_2 real α, we want to construct a c.e. real β in stages so that for all *cl*-functionals Φ_e, Ψ_e and c.e. reals γ_e:

$$R_e : \alpha \neq \Phi_e^{\gamma_e} \vee \beta \neq \Psi_e^{\gamma_e},$$

where the use of both functionals is (bounded by) $x + e$ ($e \in \omega$).

We describe an *ibT*-game amongst α, β and γ in stages as follows: (1) During the stages of the game, real γ increases; (2) If α or β changes at stage $t + 1$ and b is the leftmost change position in both of them, then γ increases in such a way that some γ-digit at a position $\leq b$ changes.

Following this game, real γ *ibT*-computes α and β simultaneously. We say γ follows the *least effort strategy* to *ibT*-compute α and β, if let $\gamma_{t+1} = \gamma_t \upharpoonright (b + 1) + 2^{-b}$.

The *cl*-game amongst α, β and γ can be deduced similarly. We say, γ follows the *least effort strategy* to *cl*-compute α and β if at each stage γ increases by the least amount which can rectify the *cl*-functionals holding its computations of α, β.

Now we recall the proof idea of Corollary 1.4, which is helpful to prove Theorem 1.7. In Corollary 1.4, it suffices to define c.e. reals α and β such that for all *cl*-functionals Φ_e, Ψ_e and c.e. reals γ_e:

$$\overline{R_e} : \alpha \neq \Phi_e^{\gamma_e} \vee \beta \neq \Psi_e^{\gamma_e},$$

where the use of both functionals is (bounded by) $x + e$ ($e \in \omega$).

To construct α, β in Corollary 1.4, the following lemmas are the key points.

Lemma 2.1 (Yu and Ding [12]). *Let $n \in \mathbb{N}^+$, We define an ibT-procedure to increase α, β in $(0, n]$ in stages as follows: if stage t is odd, let $\alpha_{t+1} = \alpha_t + 2^{-n}$; if stage t is even, let $\beta_{t+1} = \beta_t + 2^{-n}$. Real γ takes the least strategy to ibT-compute α, β. Then γ ends up with n until $\alpha(i) = \beta(i) = 1$ for all $i \in (0, n]$.*

We say an *ibT*-procedure satisfying the property in Lemma 2.1 is a P_n-procedure. Number "n" in symbol P_n means that this *ibT*-procedure increases γ to n.

Lemma 2.2 (Yu and Ding [12]). *In the cl-game amongst α, β and γ, where γ has to follow instructions of the type "change a digit at position $\leq b$", the least effort strategy is the best strategy for γ. In other words, if a different strategy produces γ' then at each stage s of the game $\gamma_s \leq \gamma'_s$.*

Listing all requirements of Corollary 1.4 effectively, we assign disjoint intervals for distinct $\overline{R_e}$-requirements. Fix $\overline{R_e}$, assume that every stronger requirement than it has been assigned an attack interval. Choose n larger than these attack intervals, and set $l = n + 2^{n+e}$. Interval $[n, l]$ is the attack interval of $\overline{R_e}$. A stage t is *expansionary* for requirement $\overline{R_e}$ if the reductions $\alpha = \Phi_e^{\gamma_e}$ and $\beta = \Psi_e^{\gamma_e}$ are longer than its attack interval at stage t. Reals α and β in Corollary 1.4 are defined as follows: at stage $t + 1$, if $\overline{R_e}$ is the strongest requirement with stage t as an expansionary stage, then let $\alpha_{t+1} = \alpha_t + 2^{-l}$ or $\beta_{t+1} = \beta_t + 2^{-l}$ alternatively as in the $P_{2^{n+e}}$-procedure. So requirement $\overline{R_e}$ is met. Otherwise, $\gamma_e \geq 2^{n+e} \times 2^{-n-e} = 1$ by Lemma 2.1 and lemma 2.2, which is contrary to the fact that $\gamma_e < 1$. There is no interaction between all $\overline{R_e}$-requirements and so we meet them without injury. For more details, see [12, 1].

The difference between R_e and $\overline{R_e}$ is that real α is given in R_e-requirements but in $\overline{R_e}$-requirements real α is defined in stages. Now we prove Theorem 1.7 through revising the proof idea of Corollary 1.4. Fix requirement R_e, we hope to assign an attack interval I_e to R_e and define the expansionary stage for R_e as for $\overline{R_e}$. Then we design a procedure to decide how to change the digits in interval I_e of β after the expansionary stages for R_e appear. When this procedure ends, real γ that takes the least effort strategy to *cl*-compute α, β is not less than 1. So requirement R_e is met by Lemma 2.2.

So how to assign interval I_e and define such procedure to increase β are two problems we faced. If interval I_e of R_e starts from n, this procedure corresponds to an *ibT*-procedure, which increases β so that real γ that follows the least strategy to *ibT*-compute α and β should not be less than 2^{n+e}. Based on the fact that real α is non-computable, we could construct such *ibT*-procedure in section 4. Since the given real α is out of control, during the stages of this *ibT*-procedure, we follow the changes of α to decide when and which digit of real β to change. Thus interval I_e of R_e, the space in which this *ibT*-procedure performs, can not be fixed in advanced. So how to give an attack interval to R_e is different from the attack interval for $\overline{R_e}$ of Corollary 1.4. Interval I_e of R_e is extended effectively as we build this procedure in stages. Meanwhile, *the expansionary stage t* for R_e should be revised as follows: the reductions $\alpha = \Phi_e^{\gamma_e}$ and $\beta = \Psi_e^{\gamma_e}$ are longer than its *current* attack interval at stage t.

Listing all requirements of Theorem 1.7 effectively, we assign disjoint intervals for distinct R_e-requirements. Since each attack interval follows from the changes of real α, if interval I_e of requirement R_e is extended, we have to assign a new attack interval to the weaker requirement than it. Therefore the attack interval of an R_e-requirement changes not only because of its corresponding procedure, but also some stronger requirement than it. Thus to accommodate all requirements simultaneously, we initialize each R_e-requirement occasionally, but finitely often. So the whole construction of β takes a genuine finite injury method. We show the proof details of Theorem 1.7 in section 5.

Lemma 2.2 is also the crucial fact to Theorem 1.7. The following lemma was shown to construct Yu-Ding procedure in [12, 1], which is also in some sense at the heart of our procedures.

Lemma 2.3 (Passing through lemma)**.** *Suppose that in some cl-game (e.g. like the above) γ has to follow instructions of the type "change a digit at a position $\leq b$". Although $\gamma_0 = 0$, some γ' plays the same game while starting with $\gamma'_0 = \sigma$ for a finite binary expansion σ. If the strategies of γ, γ' are the same (i.e. the least effort strategy described above) and the sequence of instructions only ever demand change at digits $> |\sigma|$ then at every stage s, $\gamma'_s = \gamma_s + \sigma$.*

3. The $P_{n,k}$-procedure

In this section, we present the $P_{n,k}$-procedure, which is more stricter than the P_n-procedure by Yu and Ding [12]. More important is that the construction idea of the $P_{n,k}$-procedure helps us to construct the procedure to define β in Theorem 1.7 in next section. The $P_{n,k}$-procedure depends on inputs n and k. It is defined as follows.

Definition 3.1. *Let $n \in \mathcal{A} \cap [1, +\infty)$ and $k \in \mathbb{N}^+$. An ibT-procedure to define α, β and γ is a $P_{n,k}$-procedure if the changes of α, β and γ meet the following statements:*

1. *At stage 0, $\alpha_0 = \beta_0 = \gamma_0 = 0$;*
2. *Reals α and β increase in stages;*
3. *Real γ follows the least effort strategy to ibT-compute α, β at every stage;*
4. *At every stage, $\alpha \upharpoonright k = 0$ and each α-digit change occurs at most once;*
5. *Real $\beta = 2^{-1}$ and real $\gamma = n$ when this procedure ends.*

For symbol $P_{n,k}$, number "n" means that how large real γ is when this procedure ends; number "k" means that during this procedure, the digits in interval $(0, k)$ of α are equal to 0.

Let $l_{n,k}$ be the largest number changed in α and β of this *ibT*-procedure, we say $(0, l_{n,k}]$ is the attack interval of the $P_{n,k}$-procedure.

Notice that if we substitute a $P_{2^{n+e},n}$-procedure for the $P_{2^{n+e}}$-procedure to construct α, β in Corollary 1.4, then (α, β) is a *cl*-maximal pair of c.e. reals but α corresponds to a c.e. set [8].

We define a $P_{n,k}$-procedure by induction on n. Firstly, we construct a $P_{1,k}$-procedure for any $k \in \mathbb{N}^+$ as the foundation.

Procedure 3.2. *Suppose that $\alpha_0 = \beta_0 = \gamma_0 = 0$ and fix $k \in \mathbb{N}^+$, an ibT-procedure of α, β and γ is defined in stages as follows.*

(1) *Real γ follows the least effort strategy to ibT-compute α, β at every stage;*

(2) *At stage j, let $\beta_j(j+1) = 1$ for $j = 1, \cdots, k-1$. At stage k, let $\alpha_k(k) = 1$. At stage $k+1$, let $\beta_{k+1} = 2^{-1}$. End this procedure.*

Procedure 3.2 depends on input k. If $k = 4$, we follow Procedure 3.2 to show the changes of α, β and γ in stages in Table 1. So Procedure 3.2 is a $P_{1,4}$-procedure.

Table 1

	stage 1	stage 2	stage 3	stage 4	stage 5
α	0.0000	0.0000	0.0000	0.0001	0.0001
β	0.0100	0.0110	0.0111	0.0111	0.1000
γ	0.0100	0.0110	0.0111	0.1000	1.0000

Lemma 3.3. *Procedure 3.2 is a $P_{1,k}$-procedure. So the attack interval of this $P_{1,k}$-procedure is $(0, k]$, i.e., $l_{1,k} = k$.*

Proof. By Lemma 2.3, "let $\beta_j(j+1) = 1$" adds $2^{-(j+1)}$ to γ_{j-1}. So $\gamma_{k-1} = 2^{-2} + \cdots + 2^{-k} = 0.01^{k-1}$. At stage k, "let $\alpha_k(k) = 1$ " adds 2^{-k} to γ_{k-1}. Then $\gamma_k = 2^{-2} + \cdots + 2^{-k} + 2^{-k} = 2^{-1} = 0.1$. Finally, at stage $k+1$, "let $\beta = 2^{-1}$ " adds 2^{-1} to γ_k, which increases $\gamma_{k+1} = 1$.

No digit of $\alpha \upharpoonright k$ changes till stage $k+1$ when Procedure 3.2 ends. So we have $\alpha \upharpoonright k = 0$. Procedure 3.2 ends at stage $k+1$ and it meets the definition of a $P_{1,k}$-procedure. □

If a $P_{n,k}$-procedure is defined, we can apply it in some interval with the length of $l_{n,k}$ as follows.

Lemma 3.4 (Transformation lemma). *Given stage s, and $h \in \mathbb{N}$. Suppose that number k is larger than any non-zero digit in α_s, the digits $> h$ of β_s and γ_s are equal to 0.*

(1) *If $\alpha_j(b)$ or $\beta_j(b)$ changes in a $P_{n,k}$-procedure at stage $j \in \mathbb{N}^+$, then let $\alpha_{s+j}(h+b) = \alpha_j(b)$ or $\beta_{s+j}(h+b) = \beta_j(b)$;*

(2) *Real γ follows the least strategy to ibT-compute α and β at every stage.*

(3) *When this $P_{n,k}$-procedure ends, stop changing α, β, and γ.*

This process is called that a $P_{n,k}$-procedure is performed in interval $(h, h+l_{n,k}]$ from stage s. Then $\alpha \upharpoonright k = \alpha_s \upharpoonright k$, $\beta = \beta_s + 2^{-h-1}$ and $\gamma = \gamma_s + n \cdot 2^{-h}$ when it ends. Moreover, the digits $> (h+1)$ of β and γ are equal to 0 when this $P_{n,k}$-procedure ends.

Proof. Number k is larger than any non-zero digit of α_s, and only digit $> h+k$ of α change occurs during we perform a $P_{n,k}$-procedure. So $\alpha \upharpoonright k = \alpha_s \upharpoonright k$ when this $P_{n,k}$-procedure ends. Since digits $> h$ of β_s and γ_s are equal to 0, we have $\beta = \beta_s + 2^{-h-1}$, $\gamma = \gamma_s + n \cdot 2^{-h}$ by Lemma 2.3 when this $P_{n,k}$-procedure ends.

□

Now we construct a $P_{n+0.1,k}$-procedure by induction on the finite $P_{n,j}$-procedures for $j \in \mathbb{N}^+$.

Procedure 3.5. *Suppose that the $P_{n,j}$-procedures for any $j \in \mathbb{N}^+$ have been defined.*

Fix $k \in \mathbb{N}^+$ and $\alpha_0 = \beta_0 = \gamma_0 = 0$, an ibT-procedure of α, β and γ is defined in stages as follows.

(1) *Real γ follows the least effort strategy to ibT-compute α, β at every stage.*

(2) *During the stages, we perform a P_{n,k_n}-procedure in interval $(h_n, h_n + l_{n,k_n}]$ from the stage when h_n and k_n are set.*

Stage $t = 0$. Let $h_n = 1$ and $k_n = \max\{2, k\}$.

Stage $t + 1$. If $\gamma_t \geq n - 0.1$ and digits in $[2, k]$ of γ_t are equal to 1, then let $\alpha_{t+1}(k) = 1$. If $\gamma_t = n$, let $\beta_{t+1} = 2^{-1}$ and end the whole procedure.

Otherwise, if this P_{n,k_n}-procedure ends, increase h_n by a quantity of 1, set k_n not less than any digit mentioned before in α and β; if not, keep on performing this P_{n,k_n}-procedure in interval $(h_n, h_n + l_{n,k_n}]$.

Procedure 3.5 depends on inputs n and k. Marks h_n and k_n are reset at the same time. Note that "perform a P_{n,k_n}-procedure in interval $(h_n, h_n + l_{n,k_n}]$" relies on them. We follow Procedure 3.5 for $n = 1, k = 4$ to give a $P_{1.1,4}$-procedure as an example in stages in Table 2.

Table 2

	$P_{1,4}$	$P_{1,5}$	$P_{1,6}$	$P_{1,7}$	stage $s+1$	stage $s+2$
	$(1,5]$	$(2,7]$	$(3,9]$	$(4,11]$		
α	0.00001	0.0000101	0.000010101	0.00001010101	0.00011010101	0.00011101
β	0.01000	0.0110000	0.011100000	0.01111000000	0.01111000000	0.10000000
γ	0.10000	0.1100000	0.111000000	0.11110000000	1.00000000000	1.10000000

Given $n = 1, k = 4$, let h_1^j (k_1^j) be the j^{th}-set h_1 (k_1) for $j = 1, 2, 3, 4$. We set $h_1^1 = 1$ and $k_1^1 = \max\{2, k\} = 4$ at stage 0 , then perform a P_{1,k_1^1} ($P_{1,4}$)-procedure in $(h_1^1, h_1^1 + l_{1,k_1^1}] = (1, 1 + l_{1,4}] = (1, 5]$. When this $P_{1,4}$-procedure ends, we reset h_1, k_1, then $h_1^2 = 2$ and $k_1^2 = 5$. Then we perform a P_{1,k_1^2} ($P_{1,5}$)-procedure in $(h_1^2, h_1^2 + l_{1,k_1^2}] = (2, 2 + l_{1,5}] = (2, 7]$. Following Procedure 3.5, $(h_1^3, k_1^3) = (3, 6)$ and $(h_1^4, k_1^4) = (4, 7)$. So we perform a P_{1,k_1^j}-procedure in $(h_1^j, h_1^j + l_{1,k_1^j}]$ successively. When the P_{1,k_1^4} ($P_{1,7}$)-procedure ends at stage t, then $\gamma_t > 0.1$ and the digits in $[2, 4]$ of γ_t are equal to 1. Hence, let $\alpha_{t+1}(4) = 1$, then $\gamma_{t+1} = 1$; let $\beta_{t+2} = 2^{-1}$, then $\gamma_{t+2} = 1.1$. Since number k_1^j is not less than $k = 4$, we have $\alpha \upharpoonright 4 = 0$. For $n = 1, k = 4$, Procedure 3.5 satisfies the definition of a $P_{1.1,4}$-procedure, which is built on the finite P_{1,k_1^j}-procedures for $j = 1, 2, 3, 4$.

We show that Procedure 3.5 is a $P_{n+0.1,k}$-procedure.

Lemma 3.6. (1) *Let h_n^j (k_n^j) be the j^{th}-set h_n (k_n) for $j \in \mathbb{N}^+$. If the P_{n,k_n^j}-procedure in $(h_n^j, h_n^j + l_{n,k_n^j}]$ ends at stage t_n^j, then $\gamma_{t_n^j} = n \cdot (1 - 2^{-j})$ and the digits $> h_n^{j+1}$ of $\beta_{t_n^j}$ and $\gamma_{t_n^j}$ are equal to 0.*

(2) *Procedure 3.5 is a $P_{n+0.1,k}$-procedure.*

Proof. (1) We prove this property by induction on j. By the construction, $h_n^j = j, h_n^{j+1} = h_n^j + 1$ and $k_n^1 = \max\{2, k\}$.

It is proved for h_n^1, k_n^1 by Lemma 3.4 for $s = 0, h = h_n^1, k = k_n^1$.

Suppose that this property is satisfied for $j > 1$. When a $P_{n,k_n^{j+1}}$-procedure is performed in interval $(h_n^{j+1}, h_n^{j+1} + l_{n,k_n^{j+1}}]$, it meets the assumption of Lemma 3.4 for $s = t_n^j, h = h_n^{j+1}, k = k_n^{j+1}$. Thus we have

$$\beta_{t_n^{j+1}} = \beta_{t_n^j} + 2^{-(h_n^{j+1}+1)} = \beta_{t_n^j} + 2^{-h_n^{j+2}}$$

$$\gamma_{t_n^{j+1}} = n \cdot (1 - 2^{-j}) + n \cdot 2^{-h_n^{j+1}} = n \cdot (1 - 2^{-j}) + n \cdot 2^{-(j+1)} = n \cdot (1 - 2^{-(j+1)})$$

Meanwhile, the digits $> h_n^{j+2}$ of $\beta_{t_n^{j+1}}$ and $\gamma_{t_n^{j+1}}$ are equal to 0.

(2) If $\gamma_t \geq n - 0.1$ and the digits in $[2, k]$ of γ_t are equal to 1, "let $\alpha_{t+1}(k) = 1$" increases $\gamma_{t+1} = n$; " let $\beta_{t+2} = 2^{-1}$" increases $\gamma_{t+2} = n + 0.1$. So it suffices to prove the existence of such t.

Assume that number n has the binary expansion $b_{-i} \cdots b_0.b_1 (b_{-i} \neq 0)$. Then we have

$$\begin{aligned}\gamma_{t_n^{i+k+1}} &= n \cdot (1 - 2^{-(i+k+1)}) = n - 0.0^k b_{-i} \cdots b_0 b_1 \\ &= n - 0.1 + 0.1 - 0.0^k b_{-i} \cdots b_0 b_1 \\ &= n - 0.1 + 0.01^{k-1} + (0.0^{k-1}1 - 0.0^k b_{-i} \cdots b_0 b_1)\end{aligned}$$

That is, $\gamma_{t_n^{i+k+1}} \geq n - 0.1$ and each $\gamma_{t_n^{i+k+1}}$-digit in $[2, k]$ is equal to 1. So stage t_n^{i+k+1} is stage t we expected.

For $k_n \geq k$, we have $\alpha_{t+2} \upharpoonright k = 0$ by Lemma 3.4. So Procedure 3.5 is a $P_{n+0.1,k}$-procedure.

□

Therefore, by Procedure 3.2 and Procedure 3.5, there is a $P_{n,k}$-procedure for $n \in \mathcal{A} \cap [1, +\infty)$ and $k \in \mathbb{N}^+$.

4. The $P^\alpha_{n,h,s}$-procedure

In this section, given a non-computable Δ^0_2 real α and number $n \in \mathcal{A} \cap [1, +\infty)$, we construct an *ibT*-procedure by induction to increase β, so that real γ that takes least effort strategy to *ibT*-compute α and β is not less than n. For the given real α, let $\{\alpha_s\}_{s\in\omega}$ be its effective approximation.

Recall that in Procedure 3.5 this $P_{n+0.1,k}$-procedure is built on the finite P_{n,k_n^j}-procedures, which are performed in $(h_n^j, h_n^j + l_{n,k_n^j}]$ for $j = 1, \cdots, i+k+1$ successively. The key points to realize this induction are: (1) for $h_n^j = j$, a P_{n,k_n^j}-procedure increases α and β with the digits $> j$ (where real β ends up with 2^{-j-1}) to add a quantity of $n \cdot 2^{-j}$ to γ; (2) these finite P_{n,k_n^j}-procedures could join together successively. Now we revise this induction idea to accommodate the case that α is given. Generally we hope that: (1) for $h \in \mathbb{N}^+$ there is a procedure to change the exact digits $> h$ of β at the exact stages (where real β ends up with 2^{-h-1}) to add a quantity of $n \cdot 2^{-h}$ to γ as α changes; (2) these procedures could be defined successively as h increases.

For example, let $n = 1$, as real α changes, suppose that there is a procedure for every $h \in \mathbb{N}^+$, which increases β with digits $> h + 1$ by 2^{-h-1} to add a quantity of $1 \cdot 2^{-h}$ to γ. And these procedures are defined successively as h increases. Then the quantity added to real γ could join together by Lemma 2.3. (Fix h, when real β ends up with 2^{-h-1}, the digits $> (h + 1)$ of real γ are equal to 0.) So γ is equal to

$2^{-1}+2^{-2}+\cdots+2^{-h}=0.1^h$. Since real α is non-computable, there is a stage t so that some $\alpha_t \upharpoonright h$-digit changes. Otherwise, $\alpha \upharpoonright h = \alpha_t \upharpoonright h$. That is, real α is computable, which is a contradiction. Then at stage t, $\gamma_t = 1$; at stage $t+1$, let $\beta_{t+1} = 0.1$ (for $\beta_t \upharpoonright 2 = 0$), then $\gamma_{t+1} = 1.1$. Based on a family of procedures to add $1 \cdot 2^{-h}$ to γ for $h \in \mathbb{N}^+$, the purpose to increase real $\gamma = 1.1$ succeeds. So the induction from $n = 1$ to $n = 1.1$ is feasible for the given non-computable real α.

To make the technical details of the induction for the case α is given clearer, we introduce the $P^\alpha_{n,h,s}$-procedure.

Definition 4.1. *Let $h, s \in \mathbb{N}$, and $n \in \mathcal{A} \cap [1, +\infty)$. For the given non-computable Δ^0_2 real α, fix $\alpha_s, \beta_s, \gamma_s$, and suppose that the digits $> h$ of β_s and γ_s are equal to 0. An ibT-procedure to define β and γ is a $P^\alpha_{n,h,s}$-procedure if the changes of β and γ meet the following statements after stage s:*

(1) *Real β increases with the digits $> h$ in stages;*
(2) *Real γ follows the least effort strategy to ibT-compute α, β at every stage;*
(3) *Real $\beta = \beta_s + 2^{-h-1}$ and real $\gamma \geq \gamma_s + n \cdot 2^{-h}$ when this procedure ends;*
(4) *The digits $> (h + 1)$ of β and γ are equal to 0 when this procedure ends.*

Notice that since the digits $> h$ of β_s are equal to 0, and real $\beta = \beta_s + 2^{-h-1}$ when this procedure ends, statement (4) in Definition 4.1 is trivial.

In Symbol $P^\alpha_{n,h,s}$, real "α" is the given one; stage "s" means when we start this procedure; numbers "n" and "h" together means that this procedure only changes the digits $> h$ of β to add $n \cdot 2^{-h}$ to γ_s at least. Every $P^\alpha_{n,h,s}$-procedure depends on α, n, h, s.

Firstly, assume that $\alpha_0 = \beta_0 = \gamma_0 = 0$, if real α or the approximation of α is different (see Table 3), then a $P^\alpha_{1,1,0}$-procedure as follows is different.

Table 3

Case (i)	stage 1	stage 2	stage 3	stage 4	stage 5	stage 6
α	0.0001000	0.0000010	0.0000011	0.0000001	0.0001001	0.0000011
β	0.0010000	0.0011000	0.0011100	0.0011110	0.0011111	0.0100000
γ	0.0010000	0.0011000	0.0011100	0.0011110	0.0100000	0.1000000

Case (ii)	stage 1	stage 2	stage 3	stage 4	stage 5
α	0.0000010	0.0000111	0.0000010	0.0000111	0.1000000
β	0.0010000	0.0011000	0.0011100	0.0011110	0.0100000
γ	0.0010000	0.0011000	0.0011100	0.0100000	0.1000000

Subscript h is necessary since Transformation Lemma is inapplicable for when α is given. Subscript s is necessary. For instance, if $s = 0$, when a $P^\alpha_{n,h,0}$-procedure ends depends on α, which is out of control. If the $P^\alpha_{n,h,0}$-procedure ends at stage s', to realize the induction from n to $n+0.1$ we need to define a $P^\alpha_{n,h+1,s'}$-procedure. So the arbitration of s and h provides us not only to fulfill the induction but also to apply the fitful procedure to produce a strategy for requirement R_e in Theorem 1.7.

Let $l^\alpha_{n,h,s}$ be the largest digit mentioned in β when the $P^\alpha_{n,h,s}$-procedure ends. Then $(h, l^\alpha_{n,h,s}]$ is the attack interval of this $P^\alpha_{n,h,s}$-procedure.

Now our purpose is to prove the existence of the $P^{\alpha}_{n,h,s}$-procedure for $n \in \mathcal{A} \cap [1,+\infty)$ and $h, s \in \mathbb{N}$. At first, we define the $P^{\alpha}_{1,h,s}$-procedures as the foundation for this induction.

Procedure 4.2. *Given a non-computable Δ^0_2 real α, and $h, s \in \mathbb{N}$. Fix $\alpha_s, \beta_s, \gamma_s$, the digits $> h$ of β_s and γ_s are equal to 0. An ibT-procedure of β and γ is defined in stages as follows:*

(1) *Real γ follows the least effort strategy to ibT-compute α, β at every stage;*

(2) *At stage $s+j$, let $\beta_{s+j}(h+j+1) = 1$ for $j = 1, 2, \cdots$, till stage t that the highest changing α_t-digit b is higher than the changing β_t-digit. Finally, let $\beta_{t+1} = \beta_s + 2^{-h-1}$. End this procedure.*

We give an example in Table 4 to explain this procedure. Fix stage s, and set $h = 2$, $\alpha_s = 0.0001001, \beta_s = 0.1, \gamma_s = 0.1$. If real α changes in the following cases, then let β and γ change in stages according to Procedure 4.2.

Table 4

Case (i)	stage s	stage $s+1$	stage $s+2$	stage $s+3$	stage $s+4$	stage $s+5$
α	0.0001001	0.0001111	0.0000010	0.0000011	0.0001000	0.0001011
β	0.1000000	0.1001000	0.1001100	0.1001110	0.1001111	0.1010000
γ	0.1000000	0.1001000	0.1001100	0.1001110	0.1010000	0.1100000

Case (ii)	stage s	stage $s+1$	stage $s+2$	stage $s+3$	stage $s+4$
α	0.0001001	0.0001000	0.0000001	0.0000100	0.1000000
β	0.1000000	0.1001000	0.1001100	0.1001110	0.1010000
γ	0.1000000	0.1001000	0.1001100	0.1010000	1.0000000

In Case (i), $\gamma_{s+5} = \gamma_s + 1 \cdot 2^{-2}$; in Case (ii), $\gamma_{s+4} > \gamma_s + 1 \cdot 2^{-2}$. So this is a $P^{\alpha}_{1,2,s}$-procedure. From this example, the change of α may increase $\gamma > \gamma_s + 2^{-2}$. This is the reason why in a $P^{\alpha}_{n,h,s}$-procedure real γ ends up with $\geq \gamma_s + n \cdot 2^{-h}$.

Lemma 4.3. *Procedure 4.2 is a $P^{\alpha}_{1,h,s}$-procedure. Moreover, the digits $> (h+1)$ of β and γ are equal to 0 when Procedure 4.2 ends.*

Proof. During the stages of Procedure 4.2, only the digits $\geq h+1$ of β changes. When β ends up with $\beta_s + 2^{-h-1}$, the digits $> h+1$ of β and γ are equal to 0. If no α_t-digit b exists, for $\gamma_{s+j} = \gamma_s + 0.0^{h+1}1^j$ we have $\alpha \upharpoonright (h+j+2) = \alpha_{s+j} \upharpoonright (h+j+2)$. That is, real α is computable, which is a contradiction. Thus stage t exists. And if $b \geq (h+1)$, $\gamma_t = \gamma_s + 2^{-h-1}$; if $b < (h+1)$, $\gamma_t = \gamma_s \upharpoonright (b+1) + 2^{-b} > \gamma_s + 2^{-h-1}$. Hence $\gamma_{t+1} \geq \gamma_s + 2^{-h}$. So Procedure 4.2 is a $P^{\alpha}_{1,h,s}$-procedure. □

Next we give an example to get a $P^{\alpha}_{1.1,h,0}$-procedure from the finite P^{α}_{1,h_1,s_1}-procedures for $h_1, s_1 \in \mathbb{N}$. Furthermore, it indicates us how to realize the induction from n to $n+0.1$.

Procedure 4.4. *Given a non-computable Δ^0_2 real α and $h \in \mathbb{N}, s = 0$. Assume that $\alpha_0 = \beta_0 = \gamma_0 = 0$, an ibT-procedure of β and γ is defined in stages as follows:*

(1) *Real γ follows the least effort strategy to ibT-compute α, β at every stage;*

(2) *During the stages, we define a P^{α}_{1,h_1,s_1}-procedure as Procedure* 4.2 *for the given* h_1 *and* s_1 *succesively.*

Stage $t = 0$. *Let* $s_1 = 0$ *and* $h_1 = h + 1$.

Stage $t + 1$. *If* $\gamma_t \geq 2^{-h}$, *let* $\beta_{t+1} = 2^{-h-1}$ *and end the whole procedure.*

Otherwise, if this P^{α}_{1,h_1,s_1}*-procedure ends at stage* $t + 1$, *then reset* $s_1 = t + 1$ *and increase* h_1 *by a quantity of* 1; *if not, keep on defining this* P^{α}_{1,h_1,s_1}*-procedure.*

Note that different marks s_1 and h_1 correspond to a different P_{1,h_1,k_1}-procedure. Both of them are reset at the same time.

Lemma 4.5. (1) *If there is no stage* t *that* $\gamma_t \geq 2^{-h}$, *then marks* s_1 *and* h_1 *can be reset sufficiently often.*

(2) *Let* s_1^j (h_1^j) *be the* j^{th}*-set* s_1 (h_1) *for* $j \in \mathbb{N}^+$, *then* $\gamma_{s_1^{j+1}} \geq 2^{-h} - 2^{-h_1^j}$ *and the digits* $> h_1^{j+1}$ *of* $\beta_{s_1^{j+1}}$ *and* $\gamma_{s_1^{j+1}}$ *are equal to* 0. *Furthermore, if no stage* t *appears that* $\gamma_t \geq 2^{-h}$, *then* $\gamma_{s_1^{j+1}} \upharpoonright (h + j + 1) = 0.0^h1^j$.

(3) *Procedure* 4.4 *ends up with* $\gamma \geq 1.1 \times 2^{-h}$. *So it is a* $P^{\alpha}_{1.1,h,0}$*-procedure.*

Proof. (1) Fix s_1 and h_1, it follows that $s_1 = 0$ or s_1 is the stage when Procedure 4.2 ends. Then $\beta_{s_1} = 0$ or β_{s_1} clears all digits $> h_1$. So all digits $> h_1$ of γ_{s_1} are equal to 0. It allows to perform Procedure 4.2 for this h_1 and s_1. By Lemma 4.3, both s_1 and h_1 are reset when it ends.

(2) We prove the property by induction on j. Notice that the $P^{\alpha}_{1,h_1^j,s_1^j}$-procedure ends at stage s_1^{j+1}. For $h_1^1 = h + 1$ and $h_1^{j+1} = h_1^j + 1$, we have $h_1^j = h + j$.

If $j = 1$, then $\beta_{s_1^2} = 2^{-h_1^1-1}$ by Procedure 4.2; $\gamma_{s_1^2} \geq 2^{-h_1^1} = 2^{-h} - 2^{-h_1^1}$ by Lemma 4.3. Moreover, the digits $> (h_1^1 + 1) = h_1^2$ of $\beta_{s_1^2}$ and $\gamma_{s_1^2}$ are equal to 0.

If $j > 1$, suppose that $\beta_{s_1^j}$ and $\gamma_{s_1^j}$ satisfies the property. Since the digits $> h_1^j$ of $\beta_{s_1^j}$ and $\gamma_{s_1^j}$ are equal to 0, it allows us to perform Procedure 4.2 for h_1^j and s_1^j. By the construction of Procedure 4.2, $\beta_{s_1^{j+1}} = \beta_{s_1^j} + 2^{-h_1^j-1}$. Since the digits $> h_1^j$ of $\gamma_{s_1^j}$ are equal to 0, by Lemma 2.3 $\gamma_{s_1^{j+1}} \geq \gamma_{s_1^j} + 2^{-h_1^j} \geq 2^{-h} - 2^{-h_1^{j-1}} + 2^{-h_1^j} = 2^{-h} - 2^{-h_1^j}$. Moreover, digits $> (h_1^j + 1) = h_1^{j+1}$ of $\beta_{s_1^{j+1}}$ and $\gamma_{s_1^{j+1}}$ are equal to 0.

If there is no stage t so that $\gamma_t \geq 2^{-h}$, then $\gamma_{s_1^{j+1}} \in [2^{-h} - 2^{-h_1^j}, 2^{-h}) = [2^{-h} - 2^{-h-j}, 2^{-h}) = [0.0^h1^j, 0.0^{h-1}1)$. So $\gamma_{s_1^{j+1}} \upharpoonright (h + j + 1) = 0.0^h1^j$.

(3) Since $\gamma_{s_1^j} \upharpoonright (h + j) = 0.0^h1^{j-1}$, Interval $(h, h + j)$ of $\gamma_{s_1^j} \upharpoonright (h + j)$, the digits in which are equal to 1, can be long enough as j increases. It follows that there is a stage $s' \in (s_1^{j_0}, s_1^{j_0+1})$ for some $j_0 \in \mathbb{N}^+$ so that $\alpha_{s'} \upharpoonright (h + j_0)$-digit b change occurs. Otherwise, $\alpha \upharpoonright (h + j) = \alpha_{s_1^j} \upharpoonright (h + j)$ for any j, and so real α is computable (for s_1^j is computable), which is a contradiction. As a result, $\gamma_{s'} \geq \gamma_{s'} \upharpoonright (b + 1) + 2^{-b} = 0.0^h1^{j_0-1} \upharpoonright (b + 1) + 2^{-b} \geq 2^{-h}$.

Therefore there must be a stage t so that $\gamma_t \geq 2^{-h}$. Finally, " let $\beta_{t+1} = 2^{-h-1}$" increases $\gamma_{t+1} \geq 2^{-h} + 2^{-h-1} = 1.1 \cdot 2^{-h-1}$. Meanwhile, the digits $> (h + 1)$ of β_{t+1} and γ_{t+1} are equal to 0. So Procedure 4.4 is a $P^{\alpha}_{1.1,h,0}$-procedure. □

Generally, given n, h, s, a $P^{\alpha}_{n+0.1,h,s}$-procedure would be built on the finite $P^{\alpha}_{n,h^j_n,s^j_n}$-procedures. If $n=1$, each $P^{\alpha}_{1,h^j_1,s^j_1}$-procedure can be given by procedure 4.2; if $n>1$, each $P^{\alpha}_{n,h^j_n,s^j_n}$-procedure should be constructed by induction.

The relationship between a $P^{\alpha}_{n+0.1,h,s}$-procedure and the $P^{\alpha}_{1,h',s'}$-procedure follows from the fact: let $m \in \mathcal{A} \cap [1,n]$, each $P^{\alpha}_{m+0.1,h_{m+0.1},s_{m+0.1}}$-procedure is built on the finite $P^{\alpha}_{m,h^j_m,s^j_m}$-procedures, where $h_{n+0.1}=h$, $s_{n+0.1}=s$.

Procedure 4.6. *Given a non-computable Δ^0_2 real α, $h,s \in \mathbb{N}$, and $n \in \mathcal{A}$. Assume that the digits $>h$ of β_s and γ_s are equal to 0, an ibT-procedure of β and γ is defined in stages as follows:*

(1) *Real γ follows the least effort strategy to ibT-compute α,β at every stage;*

(2) *During the stages $t>s$, we define a P^{α}_{1,h_1,s_1}-procedure as Procedure 4.2 for the given h_1 and s_1.*

Stage $t=s$. Let $m \in \mathcal{A} \cap [1,n+0.1]$, and set $s_m=s$. And set $h_{n+0.1}=h$ and $h_m = h_{m+0.1}+1$ for $m \in [1,n]$.

Stage $t+1$. Choose the largest $m \in [1,n+0.1]$ so that $\gamma_t \geq \gamma_{s_m} + (m-0.1)\cdot 2^{-h_m}$, let $\beta_{t+1} = \beta_t \upharpoonright (h_m+2) + 2^{-h_m-1}$. If $m=n+0.1$, end the whole procedure. Or else, for $k \in \mathcal{A} \cap [1,m]$, reset $s_k=t+1$ and increase h_k by a quantity of 1.

Otherwise, if no $m \in [1,n+0.1]$ meets $\gamma_t \geq \gamma_{s_m} + (m-0.1)\cdot 2^{-h_m}$, keep on defining a P^{α}_{1,h_1,s_1}-procedure as Procedure 4.2.

Note that marks s_m and h_m denote when and which β-digit change occurs. Both of them are reset at the same time. We prove that Procedure 4.6 is a $P^{\alpha}_{n+0.1,h,s}$-procedure.

Lemma 4.7. (1) *Fix $s_{m+0.1}$, assume that s_m is set for j times often before this $s_{m+0.1}$ is reset. Let s^j_m (h^j_m) be the j^{th}-set s_m (h_m) for $j \in \mathbb{N}^+$, then $\gamma_{s^{j+1}_m} \geq \gamma_{s_{m+0.1}} + m \cdot (2^{-h_{m+0.1}} - 2^{-h^j_m})$. Therefore Procedure 4.6 is a $P^{\alpha}_{m,h^j_m,s^j_m}$-procedure between stage s^j_m and stage s^{j+1}_m.*

(2) *If mark s_m is reset sufficiently often, then mark $s_{m+0.1}$ would be reset.*

(3) *Mark $s_{n+0.1}$ can be reset. So if the digits $>h$ of β_s and γ_s are equal to 0, Procedure 4.6 is a $P^{\alpha}_{n+0.1,h,s}$-procedure.*

Proof. (1)We prove this property by induction on j.

Now we consider the case that $j=1$. When mark $s_{m+0.1}=s^1_m=s$, since $h^1_m = h_{m+0.1}+1 > h$ and the digits $>h$ of β_s are equal to 0, the digits $>h^1_m$ of $\beta_{s^1_m}$ and $\gamma_{s^1_m}$ are equal to 0. When mark $s_{m+0.1}=s^1_m \neq s$, then mark $s_{m+0.1}$ was reset before because there is a stage $t < s_{m+0.1}$ so that $\gamma_t \geq \gamma_{s_{m'}} + (m'-0.1)\cdot 2^{-h_{m'}}$ for some $m' \in \mathcal{A} \cap [m+0.1, n+0.1]$. Then the digits $>(h_{m'}+1)$ of $\beta_{s_{m+0.1}}$ are equal to 0 for "$\beta_{t+1} = \beta_t \upharpoonright (h_{m'}+2) + 2^{-h_{m'}-1}$". Since mark $h^1_m = h_{m+0.1}+1 \geq h_{m'}+1$, the digits $>h^1_m$ of $\beta_{s^1_m}$ are equal to 0.

By the construction mark h_k is larger than h^1_m from stage $s^1_m (=s_{m+0.1})$ to stage s^2_m for $k \in \mathcal{A} \cap [1,m)$. During these stages, real β experiences one of the following cases: (1) let $\beta_{s_k} = \beta_{s_k-1} \upharpoonright (h_k+2) + 2^{-h_k-1}$; (2) a digit $>h_1$ of β changes during performing Procedure 4.2. It follows that only the digits $>(h^1_m+1)$ of β change during these stages. So " $\beta_{s^2_m} = \beta_{s^2_m-1} \upharpoonright (h^1_m+2) + 2^{-h^1_m-1}$" equals "$\beta_{s^2_m} = \beta_{s^1_m} + 2^{-h^1_m-1}$".

Meanwhile, $\gamma_{s_m^2} = \gamma_{s_m^2-1} \upharpoonright (h_m^1+2) + 2^{-h_m^1-1} \geq \gamma_{s_m^1} + (m-0.1)\cdot 2^{-h_m^1} + 2^{-h_m^1-1} = \gamma_{s_{m+0.1}} + m\cdot 2^{-h_m^1} = \gamma_{s_{m+0.1}} + m\cdot(2^{-h_{m+0.1}} - 2^{-h_m^1})$. Moreover, the digits $>(h_m^1+1)$ of $\beta_{s_m^2}$ and $\gamma_{s_m^2}$ are equal to 0. Therefore Procedure 4.6 is a $P^{\alpha}_{m,h_m^1,s_m^1}$-procedure from stage s_m^1 to stage s_m^2.

Assume that this property is satisfied for $j>1$. Since mark h_k is larger than h_m^j before stage s_m^{j+1} for $k\in\mathcal{A}\cap[1,m)$, only the digits $>(h_m^j+1)$ of β change between stage s_m^j and stage s_m^{j+1}. So " $\beta_{s_m^{j+1}} = \beta_{s_m^{j+1}-1}\upharpoonright(h_m^j+2)+2^{-h_m^j-1}$" equals "$\beta_{s_m^{j+1}} = \beta_{s_m^j} + 2^{-h_m^j-1}$". Meanwhile, for $h_m^j = h_m^{j-1}+1$, we get

$$\gamma_{s_m^{j+1}} = \gamma_{s_m^{j+1}-1}\upharpoonright(h_m^j+2)+2^{-h_m^j-1} \geq \gamma_{s_m^j} + (m-0.1)\cdot 2^{-h_m^j} + 2^{-h_m^j-1}$$

$$\geq \gamma_{s_{m+0.1}} + m\cdot(2^{-h_{m+0.1}} - 2^{-h_m^{j-1}}) + (m-0.1)\cdot 2^{-h_m^j} + 2^{-h_m^j-1}$$

$$= \gamma_{s_{m+0.1}} + m\cdot(2^{-h_{m+0.1}} - 2^{-h_m^{j-1}}) + m\cdot 2^{-h_m^j}$$

$$= \gamma_{s_{m+0.1}} + m\cdot(2^{-h_{m+0.1}} - 2^{-h_m^j}).$$

Moreover, the digits $>(h_m^j+1)$ of $\beta_{s_m^j}$ and $\gamma_{s_m^j}$ are equal to 0. Therefore Procedure 4.6 is a $P^{\alpha}_{m,h_m^j,s_m^j}$-procedure from stage s_m^j to stage s_m^{j+1}.

(2) If mark $s_{m+0.1}$ is not reset, and assume that mark s_m^j is reset for j times often. Let $m = b_{-i}\cdots b_0.b_1$. If $j>i+2$, then for $h_m^j = h_{m+0.1}+j$, we get

$$\gamma_{s_m^{j+1}} \geq \gamma_{s_{m+0.1}} + m\cdot(2^{-h_{m+0.1}} - 2^{-h_m^j}) = \gamma_{s_{m+0.1}} + m\cdot(2^{-h_{m+0.1}} - 2^{-h_{m+0.1}-j})$$

$$= \gamma_{s_{m+0.1}} + m\cdot 2^{-h_{m+0.1}}\cdot(1-2^{-j}) = \gamma_{s_{m+0.1}} + 2^{-h_{m+0.1}}\cdot(m - m\cdot 2^{-j})$$

$$= \gamma_{s_{m+0.1}} + 2^{-h_{m+0.1}}\cdot(m - 0.1 + 0.1 - m\cdot 2^{-j})$$

$$= \gamma_{s_{m+0.1}} + 2^{-h_{m+0.1}}\cdot(m - 0.1 + 0.01^{j-i-2} + (0.0^{j-i-2}1 - 0.0^{j-i-1}b_{-i}\cdots b_0b_1))$$

Since the digits $>(h_{m+0.1}+1)$ of $\gamma_{s_{m+0.1}}$ are equal to 0, the digits in interval $(h_{m+0.1}+1, h_{m+0.1}+j-i-1)$ of $\gamma_{s_m^j}$ are equal to 1. As j increases, this interval will be long enough. Then there is a stage $s'\in(s_m^{j_0}, s_m^{j_0+1})$ for some $j_0\in\mathbb{N}^+$ so that $\alpha_{s'}\upharpoonright(h_{m+0.1}+j-i-1)$-digit b change occurs. It increases $\gamma_{s'}\geq\gamma_{s_{m+0.1}}+m\cdot 2^{-h_{m+0.1}}$. Otherwise, $\alpha\upharpoonright(h_{m+0.1}+j-i-1) = \alpha_{s_m^j}\upharpoonright(h_{m+0.1}+j-i-1)$ for any j, and so real α is computable (for s_m^j is computable), which is a contradiction. Therefore there is a stage that $s_{m+0.1}$ is reset.

(3) If no s_m is reset for $m>1$, then s_1 would be reset sufficiently often by Procedure 4.2 and Lemma 4.3. The proof is the same as the one for Lemma 4.5 (1).

By induction, mark $s_{1.1}$ can be reset sufficiently often. By (2), stage $s_{n+0.1}$ can be reset at some stage. By (1), Procedure 4.6 ends up with $\gamma = \gamma_s + (n+0.1)\cdot 2^{-h}$, and it is a $P^{\alpha}_{n+0.1,h,s}$-procedure.

$\square$

5. The proof details of Theorem 1.7

Given a non-computable Δ_2^0 real α, it suffices to construct a c.e. real β in stages such that for all *cl*-functionals Φ_e, Ψ_e and c.e. reals γ_e:

$$R_e : \alpha \neq \Phi_e^{\gamma_e} \vee \beta \neq \Psi_e^{\gamma_e},$$

where the use of both functionals is (bounded by) $x + e$. Assume an effective list of all requirements, based on the following priority relation: $R_e < R_i$ iff $e < i$.

For every R_e-requirement, we assign an attack interval $I_e = [n_e, l_e]$. For the analysis in section 2, this I_e is given by the effective approximation $\{I_{e,t}\}_{t\in\omega}$. That is, $I_e = \lim_t I_{e,t} = [\lim_t n_{e,t}, \lim_t l_{e,t}] = [n_e, l_e]$. If at stage t the reductions $\alpha = \Phi_e^{\gamma_e}$ and $\beta = \Psi_e^{\gamma_e}$ are longer than its *current* attack interval $I_{e,t}$, we say stage t is *the expansionary stage* for R_e. Then at stage $t+1$, we say R_e *requires attention.*

During the construction, if some requirement *receives attention* at stage $t+1$, then we *initialize* all weaker requirement than it. To accommodate all requirements simultaneously, we would initialize R_e occasionally, but finitely often. Meanwhile, we assign number n_e of interval I_e for R_e-requirement finitely often. Assume that stage s_e be the stage when requirement R_e is initialized.

Construction 5.1. *We give the construction in stages.*

Stage $t = 0$. Let $\beta_0 = 0$, $I_{0,0} = [1, 2]$, $s_0 = 0$.

Stage $t + 1$. Choose the least $e \leq t$ so that R_e requires attention at stage $t + 1$. We change β_{t+1} as the β_{t+1} in the $P^{\alpha}_{2^{n_e+e},n_e,s_e}$-procedure given by Procedure 4.6 except that " γ_t" is replaced by " $2^e \cdot \gamma_{e,t}$". Then clear the digits larger than $I_{e,t}$, keep s_e, n_e unchanged. And set $n_{e,t+1} = n_e, l_{e,t+1} = l_{e,t} + 1$, i.e., the current interval $I_{e,t+1} = [n_{e,t}, l_{e,t} + 1]$. We say R_e receives attention at stage $t + 1$.

If $i < e$, then keep s_i unchanged, and let $I_{i,t+1} = I_{i,t}$.

If $e < i \leq t$, we say R_i is injured at stage $s + 1$ and initialize R_i as follows: let $s_i = t + 1$, and choose n_i larger than the attack intervals of all stronger requirements than R_i. Recall that number h_1^1 is the 1^{th}-set mark h_1 in a $P^{\alpha}_{2^{n_i+i},n_i,s_i}$-procedure by Procedure 4.6, then set $n_{i,t+1} = n_i, l_{i,t+1} = n_i + h_1^1$, i.e., the current interval $I_{i,t+1} = [n_i, l_{i,t+1}]$.

If no R_e requires attention at stage $t + 1$, end stage $t + 1$.

In the $P^{\alpha}_{2^{n_e+e},n_e,s_e}$-procedure given by Procedure 4.6, the possible changing digit of β is not lower than the next lower digit of $l_{e,t}$ when R_e receives attention again. So we expand $I_{e,t+1} = [n_{e,t}, l_{e,t} + 1]$. Since real β keeps increasing, the action "clear the digits larger than $I_{e,t}$" is reasonable as R_e receives attention.

Lemma 5.2. *Fix e, requirement R_e receives attention at most finitely often and is eventually satisfied. Moreover, interval I_e is fixed as R_e is satisfied.*

Proof. Fix e and assume by induction that Lemma 5.2 holds for all $i < e$. Choose t minimal so that no R_i, $i < e$, receives attention after stage t. Hence, requirement R_e is not initialized after stage t. By the construction, $s_e = t$. Then s_e and $n_e = n_{e,t}$ of interval I_e are unchanged forever. Since when an *ibT*-procedure, the $P^{\alpha}_{2^{n_e+e},n_e,s_e}$-procedure given by Procedure 4.6, starts at stage s_e, the digits $> n_e$ of β_{s_e} and

$\gamma_{e,s_e} \cdot 2^e$ should be equal to 0. But the digits $> (n_e + e)$ of γ_{e,s_e} may be not equal to 0. Procedure 4.6 and Lemma 4.7 for 2^{n_e+e}, n_e, s_e still work well since the digits $> (n_e+e)$ of β_{e,s_e} are equal to 0. (The quantity of $(\gamma_{e,s_e} \cdot 2^e - \gamma_{e,s_e} \cdot 2^e \upharpoonright (n_e + 1))$ is regarded as the one that some P^{α}_{m,h_m,s_m}-procedures in the $P^{\alpha}_{2^{n_e+e},n_e,s_e}$-procedure increase to $\gamma_{e,s_e} \cdot 2^e \upharpoonright (n_e + 1)$, $m \in [\mathcal{A} \cap [1, 2^{n_e+e}]$, which has been realized in advance.) Then by Lemma 4.7 and Lemma 2.2, requirement R_e receives attention at most finitely often and is eventually satisfied. Otherwise, when this $P^{\alpha}_{2^{n_e+e},n_e,s_e}$-procedure ends, $\gamma_e \cdot 2^e \geq (\gamma_{e,s_e} \cdot 2^e \upharpoonright (n_e + 1) + 2^{n_e+e}) \cdot 2^{-n_e} \geq 2^e$, i.e., $\gamma_e \geq 1$, which is a contradiction. □

References

[1] Andrew E.M. Lewis and George Barmpalias. Random reals and lipschitz continunity. *Mathematical Structures in Computer Science, 2006, **16**(5).*

[2] George Barmpalias; Andrew E.M. Lewis. A c.e. real that cannot be SW-computed by any Ω number. *Notre Dame J. Formal Logic 47 (2006), no. 2, 197–209.*

[3] George Barmpalias and Andrew Lewis. The ibT degrees of c.e. sets are not dense. *Annals of Pure and Applied Logic Volume **16**, issues 1-2,(2006).*

[4] George Barmpalias and Andrew Lewis. Randomness and the linear degrees of computability. *Annals of Pure and Applied Logic 145(2007),252-257.*

[5] George Barmpalias, Rod Downey and Noam Greenberg. Working with strong reducibilities above totally ω-c.e. degrees. *Transactions of the American Mathematical Society. In press.*

[6] Rodney G. Downey, Denis Hirschfeldt. Randomness and reducibility. *Springer-Verlag Monographs in Computer Science. In Preparation.*

[7] Rodney G. Downey, Denis Hirschfeldt and G.LaForte, Randomness and reducibility, *in Mathematical Foundations of Computer Science 2001 (Sgall,Pultr,P. Kolman,eds.), Lecture Notes in Computer Science 2136, Springer, 2001, 316C327. Final version in Journal of Computing and System Sciences, **68**(2004), 96-114.*

[8] Yun Fan. There is an sw-cuppable strongly c.e. real. *Lecture Notes of Compute Scinece, TAMC 2007.*

[9] Solovay, R.. Draft of paper (or series of papers) on Chaitins work, unpublished notes. *May, 1975, 215 pages.*

[10] Robert I. Soare. Recursively Enumerable Sets and Degrees. *Springer-Verlag, Berlin, 1987.*

[11] Robert I. Soare. Computability Thoery and Differential Geometry. *the Bulletin of Symbolic Logic, Volume 10, Issue 4 (2004), 457-486.*

[12] Liang Yu and Decheng Ding. There is no SW-complete c.e. real. *J. Symbolic Logic* **69** (2004), no. 4, 1163–1170.

Department of Mathematics, Southeast University, Nanjing, China
E-mail address: fanyun@seu.edu.cn

Institute of Mathematical Science, Nanjing University,Nanjing, China
E-mail address: yuliang.nju@gmail.com