ON THE DEFINABLE IDEAL GENERATED BY NONBOUNDING C.E. DEGREES

YUE YANG AND LIANG YU

Abstract. Let $[\mathrm{NB}]_1$ denote the ideal generated by nonbounding c.e. degrees and NCup the ideal of noncuppable c.e. degrees. We show that both $[\mathrm{NB}]_1 \cap \mathrm{NCup}$ and the ideal generated by nonbounding and noncuppable degrees are new, in the sense that they are different from M, $[\mathrm{NB}]_1$ and NCup — the only three known definable ideals so far.

1. Introduction

This paper is part of the study of definable subsets in the structure of computably enumerable degrees $\mathcal{R}$. One of the most significant results is that all jump classes except the low degrees are definable, by Nies, Shore and Slaman [5], [6]. This was done in the mid 1990s. However, even until 2000, no nontrivial definable ideals are known except M and NCup. Recall that two noncomputable c.e. degrees $\mathbf{a}$ and $\mathbf{b}$ form a *minimal pair* if $\mathbf{a}\wedge\mathbf{b} = \mathbf{0}$; M is the set of all *cappable* c.e. degrees i.e., the halves of minimal pairs; a c.e. degree $\mathbf{a}$ is *noncuppable* if for all incomplete c.e. degree $\mathbf{b}$, $\mathbf{a}\vee\mathbf{b}$ is incomplete; and NCup is the set of all noncuppable c.e. degrees. The problem was raised in 1999 Boulder's meeting by Shore (Question 2.8 [7]): Are there other definable ideals? Furthermore, Shore asked (Question 2.9 [7]): If $\mathcal{B}$ is a (particular) definable subset of $\mathcal{R}$ is there a way to define the ideal generated by $\mathcal{B}$? Recently, Nies [4] proved the following powerful result:

Theorem 1.1. *Let $\mathcal{B}$ be a definable subset of $\mathcal{R}$. Then the ideal generated by $\mathcal{B}$ is definable in $\mathcal{R}$.*

Theorem 1.1 produces many definable ideals, the concern then shifts to whether or not they are new. For example, let us say a c.e. degree $\mathbf{t}$ is a diamond top if

$$(\exists \mathbf{x}, \mathbf{y} \neq \mathbf{0})[\mathbf{x} \vee \mathbf{y} = \mathbf{t} \text{ and } \mathbf{x} \wedge \mathbf{y} = \mathbf{0}].$$

Consider the ideal DT which is generated by of all diamond tops.

Proposition 1.2. *The ideal DT coincides with the ideal M.*

Proof. Clearly, every diamond top, being a join of a minimal pair, is in M. On the other hand, every degree $\mathbf{x}$ in M is below a diamond top $\mathbf{t}$ which is the join of $\mathbf{x}$ and its companion. $\square$

1991 *Mathematics Subject Classification.* 03D25.

The first author was partially supported by NSTB OAP programme and NUS Grant No. R-146-000-028-112 (Singapore). The second author is supported by postdoctoral fellowship from the New Zealand Institute for Mathematics and its Applications, NSF of China No. 6010213 and No. 19931020. Both authors wish to thank André Nies for discussions.

By studying the nonbounding degrees, Nies [4] obtained a new definable ideal $[\mathrm{NB}]_1$. We say a c.e. degree **b** *nonbounding* if it does not bound any minimal pairs. It is known that the set of all nonbounding degrees does not form an ideal (see, for example, Ambos-Spies and Soare [1]). Let $[\mathrm{NB}]_1$ be the ideal generated by the nonbounding degrees. By Theorem 1.1, $[\mathrm{NB}]_1$ is definable. Since every nonbounding degree is cappable and M is an ideal, we know that $[\mathrm{NB}]_1$ is a subset of M. Furthermore, Nies showed that $[\mathrm{NB}]_1$ coincides with neither M nor NCup:

Theorem 1.3 (Nies).
1. *There is a cappable degree which is not in $[NB]_1$. Thus $[NB]_1$ is properly contained in M.*
2. *There is a cuppable degree in $[NB]_1$. Thus $[NB]_1$ is not a subset of NCup.*
3. *There is a c.e. degree which is both noncuppable and nonbounding. Thus the intersection of NCup and $[NB]_1$ is not empty.*

It is natural to ask the following question so that we have more precise information about the ideals:

Question 1.4. *Is NCup a subset of $[NB]_1$?*

In the first part of the paper, we give a negative answer of the question.

Theorem 1.5. *There exists a noncuppable c.e. degree* **a**, *which is not in $[NB]_1$.*

Once we know that the two ideals NCup and $[\mathrm{NB}]_1$ are not containing each other, it is natural to look at the ideal I generated by both noncuppable and nonbounding degrees.

Theorem 1.6. *There exists a cappable c.e. degree* **a** *which is not in I.*

Thus we obtain two new definable ideals: the intersection of NCup and $[\mathrm{NB}]_1$ and the ideal I.

We organise the paper as follows. Section 2 and 3 are devoted to the proof of Theorem 1.5 and Theorem 1.6 respectively. In each section, we have subsections for requirements and strategies; formal proof; and verification. When we explain the strategies, we will deal with special case such as the degree generated by two elements, instead of n elements. We believe that the explanation of this special case illustrates the main ideas, which might be obscured by the complicated indexing in the general setting. Once the idea is understood, we give the construction for the general case.

Notation and terminology are standard and generally follow Soare [8]. The basic knowledge of tree constructions in computability theory is assumed. We use capital Greek letters such as Φ to denote Turing functionals, and the corresponding lower case letter $\varphi(A;x)$ to denote the use function for $\Phi(A;x)$. If the Turing functional Φ applies to the join of two sets X and Y, we will write $\Phi(XY)$ instead of $\Phi(X\oplus Y)$. During the course of a construction, whenever we define a parameter as *fresh*, we mean that it is defined as the least natural number which is greater than any number mentioned so far. We assume that the priority tree grows *upwards*.

2. The Proof of Theorem 1.5

Fix a complete c.e. set K_0. Our target is to build a c.e. set A such that:

(1) A is noncuppable; and
(2) for any c.e. sets $B_1, B_2, \ldots, B_n$, if $A \leq_{\mathrm{T}} B_1 \oplus B_2 \oplus \cdots \oplus B_n$, then one of the B_k's ($1 \leq k \leq n$) bounds a minimal pair. In other words, either $A \not\leq_{\mathrm{T}} B_1 \oplus B_2 \oplus \cdots \oplus B_n$, or there is a minimal pair X_k and Y_k below B_k for some $k : 1 \leq k \leq n$.

We now look at each individual requirement. For notational simplicity, we work on the special case $n = 2$ until we give the full construction. The strategies for a requirement O described below often corresponds to the α-version of O, where α is a node on the priority tree labelled O.

2.1. **Description of noncuppable strategies.** We follow the strategies used by Li, Slaman and Yang [3]. The noncuppable requirements guarantee that for all c.e. set W, $A \oplus W$ is complete implies that W is complete. Fix an effective enumeration of Turing functionals Γ_e ($e \in \omega$). For each pair of natural numbers e_1 and e_2, we build a Turing functional Δ_e such that if $\Gamma_{e_1}(AW_{e_2}) = D$, then $\Delta_e(W_{e_2}) = K_0$, where e is the code $e = \langle e_1, e_2 \rangle$ and D is an auxiliary set built by us. For simplicity, we would like to view K_0 as a subset of D, so that any number enumerated into K_0 is enumerated into D automatically. Thus let us assume that K_0 (D, respectively) is a subset of even (odd, respectively) numbers and K is the (disjoint) union of K_0 and D. The noncuppable requirements N_e are as follows.

- N_e: If $\Gamma_{e_1}(AW_{e_2}) = K$, then $\Delta_e(W_{e_2}) = K_0$.

From now on, when we define the value $\Delta_e(W_{e_2}; p)$, we always assume that p is an even number; we may assume that if q is an odd number, then $\Delta_e(W_{e_2}; q) = 0$ with empty use. We also assume that the candidates targeting D are chosen from odd numbers.

We have subrequirements $M_{e,p}$ (p is an even natural number) working for N_e, each $M_{e,p}$ is responsible for defining $\Delta_e(W_{e_2}; p)$.

- $M_{e,p}$: If $\Gamma_{e_1}(AW_{e_2}) = K$, then $\Delta_e(W_{e_2}; p)$ is defined and equal to $\Gamma_{e_1}(AW_{e_2}; p)$.

Let α be a node labelled N_e. The strategy for α works as follows. We omit the index e during the discussion if there is no confusion. Define the length of agreement function $l(\alpha, s)$ between $\Gamma(AW)$ and K as usual. We define current stage s to be α-*expansionary stages* if $l(\alpha, s)$ is longer than any $l(\alpha, t)$ where $t < s$ is an α-accessible stage. α has two possible outcomes: ∞ for infinitely many α-expansionary stages; and 0 for finitely many ones.

About the outcome ∞, α has substrategies $M_{e,p}$. Each $M_{e,p}$ is responsible for defining $\Delta(W : p)$. To make the definition of Δ consistent, the nodes to the right must follow the definition of nodes to the left; and whenever we access a node α, we must make all $\Delta(W; p)$ defined at some nodes to the right of α undefined (see more details in [3]). The strategy for $M_{e,p}$ works as follows. We first check if the use $\gamma(AW; p)$ has changed since the stage at which it was accessible for the last time. If yes, then $M_{e,p}$ has outcome ∞, because it indicates that $\Gamma(AW; p)$ is partial; otherwise, we select a number (called a *flip point*) d not yet in D, delay the definition of $\Delta(W; p)$ until $\Gamma(AW; d)$ is defined; then define $\Delta(W; p) = \Gamma(AW; p)$ with use $\gamma(AW; d)$ and restrain A up to $\gamma(AW; d)$. When we need to make $\Delta(W; p)$ undefined, we put d into D and wait for a stage at which W changes below $\gamma(AW; d)$. Under the assumption that $\Gamma(AW;d) = D(d)$, W must change below $\gamma(AW;d)$. Since $\gamma(AW;d) = \delta(W;p)$, $\Delta(W;p)$ becomes undefined.

The implementation of forcing $\Delta(W;p)$ to be undefined goes as follows. Suppose τ is a node above $\alpha\hat{}\infty$ which has outcomes $o_1 <_L o_2$. If there are $\Delta(W;p)$'s which have been defined by some nodes extending $\tau\hat{}o_2$, then we delay τ to have outcome o_1. We will refer this situation as "*α stops τ having outcome o_1*". Before we declare that τ is accessible, we must force W to change so that those $\Delta(W;p)$'s become undefined. We act as follows: Pick the smallest p such that $\Delta(W;p)$ is defined at some node extending $\tau\hat{}o_2$, let d be the flip point for p. Put d into D and set up a link (α,τ). Wait for the next α-expansionary stage $t > s$. At stage t, we travel the link to τ and α will not stop τ having outcome o_1.

If there are more than one N-requirements below τ, then we have to deal with them one by one. For example, assume N_0 and N_1 are the only two N-requirements which are assigned to α_0 and α_1 respectively such that

$$\alpha_0\hat{}\infty \subseteq \alpha_1\hat{}\infty \subseteq \tau.$$

Then we deal with N_1 by putting the corresponding flip point d_1 into D and setting up a link (α_1,τ). At the next α_1-expansionary stage, we travel the link to τ and cancel the link (α_1,τ). Next we deal with N_0 by putting its flip point d_0 into D and setting up a new link (α_0,τ). As long as the link exists, N_1 is bypassed so that we do not define more Δ_1 axioms for N_1. At the next α_0-expansionary, we travel through the link (α_0,τ) to τ and τ can be accessible now.

2.2. **Description of bounding strategies.** Fix effective enumerations of c.e. sets B_e ($e \in \omega$) and effective enumerations of Turing functionals Φ_e, Ψ_e and Θ_e ($e \in \omega$). Our job is to show that either $A \not\leq_{\mathrm{T}} B_{e_1} \oplus B_{e_2}$ or there is a minimal pair X_{e_k} and Y_{e_k} below B_{e_k} for some $k \in \{1,2\}$. Fix k. Recall the typical strategies of constructing a minimal pair X_{e_k} and Y_{e_k}.

- $P^k_{e,2i}$: $X_{e_k} \neq \Psi_i$; and
- $P^k_{e,2i+1}$: $Y_{e_k} \neq \Psi_i$;
- $T^k_{e,i}$: If $\Theta_i(X_{e_k}) = \Theta_i(Y_{e_k}) = f$ and f is total, then f is computable.

We split the bounding requirement into R_e, $S_{e,i}$ and $T_{e,i}$ as follows.

- R_e: If $A = \Phi_{e_0}(B_{e_1}B_{e_2})$ where $e = \langle e_0, e_1, e_2\rangle$, then there are c.e. sets X_{e_k} and $Y_{e_k} \leq_{\mathrm{T}} B_{e_k}$ ($k = 1,2$) such that one pair of X and Y form a minimal pair.

The (α-th version of) strategy for R_e is as follows. Let B denote $B_{e_1} \oplus B_{e_2}$. We first test if $\Phi_{e_0}(B) = A$ by measuring the length of agreement $l(\alpha,s)$ between $\Phi_{e_0}(B)$ and A, where

$$l(\alpha,s) = \mu y(\Phi_{e_0}(B;y) \neq A(y)[s]).$$

We define α-expansionary stages as in the noncuppable strategies. α has two possible outcomes: ∞ for infinitely many α-expansionary stages and 0 for finitely ones. At node α we also build the sets X_{e_k} and Y_{e_k} computable from B_{e_k} by permitting method. The candidates targeting X_{e_k} and Y_{e_k} will be chosen at some nodes β working for α, but the control is at α.

Above the node $\alpha\hat{}\infty$, we will satisfy subrequirements $S_{e,i}$ and $T^k_{e,i}$.

- $S_{e,i}$: P^1_{e,i_1} is satisfied or P^2_{e,i_2} is satisfied where $i = \langle i_1, i_2 \rangle$.
- $T^k_{e,i}$: Same as in the minimal pair requirement where $k = 1, 2$.

From now until the end of this section, we reserve the letters α, β and γ to denote the nodes on the priority tree labelled R_e and its subrequirements $S_{e,i}$ and $T^k_{e,i}$ respectively. If no confusion, we drop the index e in the discussion. In particular, we write X_1 instead of X_{e_1}, etc.

Consider a node β labelled $S_{e,i}$. We have to deal with a pair of positive strategies P^1_{e,i_1} and P^2_{e,i_2}. For example, let us work on the pair $X_1 \neq \Psi_{i_1}$ and $X_2 \neq \Psi_{i_2}$. Each individual P-strategy is done by Friedberg-Muchnik diagonalization. To cope with permitting, we need to do some set up. Pick a fresh number a, in particular, $a \notin A$. This a remains fixed, unless β is initialised. Wait for a stage at which $\Phi_{e_0}(B; a) \downarrow = 0$. Pick witnesses $v_k > \varphi(B; a)$, $k = 1, 2$. v_k will be the diagonalization candidate targeting X_k. When β is accessible for the next time, we check if $\varphi(B; a)$ has moved, if yes, then we start over, we use ∞ to denote this outcome. If ∞ is the true outcome, then $\Phi_{e_0}(B)$ is partial, we have a global win for R_e. Let us assume that $\varphi(B; a)$ is eventually fixed. Whenever β is accessible, we check if $\Psi_{i_k}(v_k) \downarrow = 0$ for both $k = 1$ and $k = 2$. If $\Psi_{i_k}(v_k)$ never converges to 0 for some $k = 1, 2$, then we have an easy win for $S_{e,i}$. We use 1 to denote this outcome. Otherwise, suppose that at some stage $t > s$ we find that $\Psi_{i_k}(v_k) \downarrow = 0$ for both $k = 1$ and $k = 2$, then we put a into A and set up a link of the form (α, β). We wait for a B-change below $\varphi(B; a)$. If there is no such a change, then $\Phi(B) \neq A$, thus we get a global win for R (α will have outcome 0 forever). Suppose there is such a change in $B = B_{e_1} \oplus B_{e_2}$. If B_{e_1} changes below the use $\varphi(B; a)$, then v_1 is permitted by B_{e_1}, we enumerate v_1 into X_1. Otherwise, that is, B_{e_1} does not change below the use $\varphi(B; a)$, then B_{e_2} must change, we put v_2 into X_2. In both cases we let β have outcome 0 and we cancel all other witnesses at β.

The outcomes at a node β labelled $S_{e,i}$ are (from left to right): $\infty < 0 < 1$.

The purpose of link is to make sure the X_k and Y_k are computable from B_{e_k}. The main worry is that some number v located at β, which has passed B's permission but the node β is never accessible again. The link allow us to reach β at the next α-expansionary stage. The fate of v will be determined without any delay. Of course, if there are no more α-expansionary stages, then we do not need to build X_k or Y_k.

We now look at the substrategy $T^k_{e,i}$ at node γ. Without loss of generality, let us assume that $k = 1$. We follow the typical minimal pair construction. Let $l(\Theta, \gamma)$ measure the length of agreement between $\Theta_i(X_1)$ and $\Theta_i(Y_1)$. At any stage, we will preserve at least one side of the computation up to $l(\Theta, \gamma)$. At γ-expansionary stages, we allow numbers to enter either X_1 or Y_1 but not both. At non-γ-expansionary stages, γ imposes a finite restraint on all nodes above or to the right of γ. γ has two outcomes: ∞ and 0.

Before we give the formal proof for the general case, let us look at the potential conflict.

The main concern is about the coordination between clearing Δ procedure and the permitting. The permitting should not be delayed by the noncuppable strategy which wishes to clear the Δ's.

Observe that when we travel the link from R to S, the action is to put a number v into X. This will not injure any noncuppable requirement. Recall that the only reason to clear Δ is to prevent numbers entering A. In other words, as long as we do not put numbers into A, the noncuppable strategy will not be injured. Thus, the action at S has no conflict with M, thus we can execute the S-strategy at β without delay. This is different from the first visit of β, when we had not set up the link. At that time, we must clear all the axioms of Δ to the right before we allow S to put a number a into A.

Secondly, there is no crossing of links. The reason is when we want to set up a link (α, β) between R and S, α is not covered at that moment. The same holds for link (α, τ) between N and M.

2.3. **The Proof.** To avoid the possible confusions of notations, we restate the bounding requirements in the general setting.

First, fix a computable bijection $\langle e_1, \dots, e_n\rangle \mapsto e$ from $\omega^{<\omega}$ to ω. Fix effective enumerations of c.e. sets B_e $(e \in \omega)$ and Turing functionals Φ_e, Ψ_e and Θ_e $(e \in \omega)$. The bounding requirements and subrequirements R_e, $S_{e,i}$ and $T_{e,i}$ are as follows.

- R_e: If $A = \Phi_{e_0}(B_{e_1}B_{e_2}\dots B_{e_n})$ where $e = \langle e_0, e_1, \dots, e_n\rangle$, then there are c.e. sets X_{e_k} and $Y_{e_k} \leq_{\mathrm{T}} B_{e_k}$ $(1 \leq k \leq n)$ such that at least one pair of X_{e_k} and Y_{e_k} form a minimal pair.

The building of minimal pair is done by subrequirements $S_{e,i}$ and $T^k_{e,i}$:

- $S_{e,i}$: $P^k_{e,i}$ is satisfied for some $1 \leq k \leq n$ where $e = \langle e_1, e_2, \dots, e_n\rangle$ and $P^k_{e,i}$ is the positive requirement of the noncomputability of X_{e_k} and Y_{e_k}.
  - $P^k_{e,2i}$: $X_{e_k} \neq \Psi_i$, and
  - $P^k_{e,2i+1}$: $Y_{e_k} \neq \Psi_i$.
- $T^k_{e,i}$: If $\Theta_i(X_{e_k}) = \Theta_i(Y_{e_k}) = f$ and f is total, then f is computable.

We now describe the priority tree T. Fix a computable priority list of the requirements and subrequirements such that the subrequirements $M_{e,p}$ (or $S_{e,i}$, $T^k_{e,i}$ respectively) appear after N_e, (R_e, respectively). We label T inductively in the usual manner. We label each node on T with a requirement or a subrequirement. The root node on T is labelled R_0. Suppose that τ is a node on T. If τ is labelled $S_{e,i}$ then τ has three outgoing edges labelled $\infty, 0, 1$ from left to right; otherwise, that is, if τ is labelled $N_e, M_{e,p}$, R_e or $T^k_{e,i}$, then τ has two outgoing edges labelled ∞ and 0, with ∞ to the left of 0.

We say that a requirement N_e is *satisfied* at τ if there is a node $\alpha \subset \tau$ labelled N_e such that either $\alpha\hat{\ }0 \subseteq \tau$ or there is an η labelled with a subrequirement $M_{e,p}$ working for N_e such that $\eta\hat{\ }\infty \subseteq \tau$. (*Namely, we see a global win for N_e at η.*) If N_e is satisfied at τ then all its subrequirements are satisfied at τ.

We say that a requirement R_e is *satisfied* at τ if there is a node $\alpha \subset \tau$ labelled R_e such that either $\alpha\hat{\ }0 \subseteq \tau$ or there is an η labelled with a subrequirement $S_{e,i}$ working for R_e such that $\eta\hat{\ }\infty \subseteq \tau$. *Namely, we see a global win for R_e at η.* If R_e is satisfied at τ then all its subrequirements are satisfied at τ.

Continuing the inductive definition of T, if all $\alpha \subset \tau$ have been labelled, then τ is labelled with the highest priority O such that O is either a requirement which never appeared before or O is a unsatisfied new subrequirement.

2.4. **Conventions and parameters.** Let α be a node on T. We now list a collection of parameters related to α, which will be used in the construction. Strictly speaking, we should write $q(\alpha, s)$ for the value of parameter q at the beginning of stage s, however, when there is no confusion, we will simply write q. We may also drop the indices of the requirements.

(1) If α is labelled N_e, then no parameter is needed.
(2) If α is labelled $M_{e,p}$, then it has a flip point d for $\Delta(W;p)$.
(3) If α is labelled R_e, then no parameter is needed.
(4) If α is labelled $S_{e,i}$, then it has the following parameters
  - a node σ below α labelled R_e, for which it is working, we will call σ the *head of* α;
  - a number r which is the finite restraint imposed by the T-nodes which are working for the same R and below or to the left of α. Although those T-nodes may put different restraints $r^X_{e_k}$ or $r^Y_{e,k}$ on different sets X_{e_k} or $Y_{e,k}$. It does no harm if we take r to be the maximal of those small restraints.
  - a number a, called an *agitator*, which will be used to seek an permission from $B = B_{e_1} \oplus B_{e_2} \oplus \cdots \oplus B_{e_n}$;
  - a set of n numbers $v_k > \varphi^B(a)$ where $v_k > r$ for every k with $1 \leq k \leq n$.
(5) If α is labelled $T^k_{e,i}$, then its environment contains its length of agreement l and a finite restraint r to preserve l.

2.5. **Construction.** We now describe the stage by stage construction. At stage s, we first specify a string TP_s of length less than or equal to s, called the *accessible string*, then act along the accessible string.

We define the accessible string inductively from the root. The root of the tree is always accessible.

At the inductive step, suppose that the node α is accessible. If the length of α is equal to s then we let $\alpha = \mathrm{TP}_s$ and go to the next stage.

Suppose that the length of α is less than s. Then we first determine the outcome o of α. Before we declare that $\alpha\hat{\ }o$ is accessible, we check if there is any N-requirement below α which stops α having outcome o. If yes, then we stop defining the accessible string, start the procedure below, referred as *clearing* Δ *for* o *at* α and delay all actions; otherwise, we let $\alpha\hat{\ }o$ be accessible and take actions accordingly.

The procedure of clearing Δ for o at α is as follows.

Given α and an outcome o. Ask if there is a pair of requirements N_e and $M_{e,p}$, such that

(1) N_e is assigned to some node β below α, and
(2) $M_{e,p}$ is a subrequirement for N_e, $M_{e,p}$ is assigned to some node τ to the right of $\alpha\hat{\ }o$ and at τ we have defined $\Delta_e(W;p)$.

If yes, then let β_0 be the longest such β, p_0 be the smallest p for β_0 and d_0 be the flip point for p_0. Put d_0 into D and set a link (β_0, α). Initialise all nodes not labelled M which are to the right of $\alpha\hat{\ }o$, that is, cancel all actions desired by these nodes; cancel all parameters; cancel all restraint imposed by these nodes; and cancel all links involving the node.

We now continue the definition of outcome of α and the next accessible node. We consider the two cases based on whether or not there is a link starting from α.

Case 1. There is a link of the form (α, τ).

Subcase 1.1. α is labelled with an N-requirement.

Then the link (α, τ) must have been set when we clear Δ for some outcome o at τ. Check if the stage s is α-expansionary.

- If no, then let $\alpha\hat{\ }0$ be accessible (Since 0 is the rightmost outcome of α, no Δ-clearing is needed).
- If yes, then go to τ and cancel the link (α, τ). Check if there is an N-requirement which stops τ having outcome o.
  - If yes, then repeat the procedure of clearing Δ for o at τ;
  - otherwise, let $\tau\hat{\ }o$ be accessible and acts as described in the construction below.

Subcase 1.2. α is labelled with an R-requirement, say R_e.

Then τ must be labelled with an S-requirement, say $S_{e,i}$. Check if the stage s is α-expansionary.

- If no, then let $\alpha\hat{\ }0$ be accessible (as discussed in Subcase 1.1, no Δ-clearing is needed).
- If yes, then go to τ and cancel the link (α, τ). Now one of the B_{e_k} $(1 \leq k \leq n)$ must have changed below $\varphi(B; a)$ where $B = B_{e_1} \oplus B_{e_2} \oplus \cdots \oplus B_{e_n}$ and a is the agitator; and the witnesses v_k $(1 \leq k \leq n)$ are all available. Choose the least k such that B_{e_k} has changed below $\varphi(B; a)$, enumerate v_k into the set it was targeting. Go to the next stage.

Case 2. There is no link of the form (α, τ).

Then we first decide the outcome o of α, then check if there is any N which stops α having outcome o. If yes, then start the procedure of clearing Δ for o as described earlier; if no, take the actions described below.

We decide the outcome o and take the actions based on the label of α as follows.

(1) α is labelled N_e.

Check if s is an α-expansionary stage. If yes, then check if there is any node stopping α having outcome ∞. If yes, then start the clear Δ process; if no, let $o = \infty$. Otherwise, that is, s is not an α-expansionary stage, check if there is an α-expasionary stage $v < s$ since $\alpha\hat{\ }\infty$ was visited for the last time, at which α was stopped having outcome ∞. If yes, then let $o = \infty$, otherwise let $o = 0$. No action is required since the jobs are distributed to the subrequirements $M_{e,p}$.

(2) α is labelled $M_{e,p}$.

Let d be the flip point for α (if d is undefined, then pick it fresh). If $\Gamma_{e_1}(AW_{e_2}; d)$ is undefined, then go to next stage. Otherwise, check if $\gamma_{e_1}(AW_{e_2}; d)[s] \neq \gamma_{e_1}(AW_{e_2}; d)[t]$, where t is the stage at which α was accessible for the last time. If yes, then $o = \infty$; otherwise $o = 0$.

Actions. We take action only when α has outcome 0. Check if $\Delta_e(W_{e_2}; p)$ is defined at stage s.

- If yes, then do nothing;
- otherwise, define

$$\Delta_e(W_{e_2}; p) = \Gamma_{e_1}(AW_{e_2}; p)$$

with use $\gamma_{e_1}(AW_{e_2}; d)$ and set a restraint on A of amount $\gamma_{e_1}(AW_{e_2}; d)$.

(3) α is labelled R_e.

Check if s is α-expansionary. If yes, then let $o = \infty$; otherwise, let $o = 0$. As in the N-requirement, no action is needed since the jobs are done by its subrequirements.

(4) α is labelled $S_{e,i}$. Without loss generality, let us assume that S consists of the positive requirements of the form: $X_{e_k} \neq \Psi_{i_k}$ for $1 \leq k \leq n$. Let σ denote its head.

First check if $S_{e,i}$ has been *satisfied*, that is, there was a stage $t < s$ at which we put a number v_k into X_{e_k} for some $1 \leq k \leq n$.

- If yes, then let $o = 0$.
- Otherwise, let a be the agitator (if a is not defined, then choose it fresh). Check if $\Phi_{e_0}(B; a)$ is undefined or $\varphi_{e_0}(B; a)[s] \neq \varphi_{e_0}(B; a)[t]$, where $B = B_{e_1} \oplus B_{e_2} \oplus \cdots \oplus B_{e_n}$ and t is the stage at which α was accessible for the last time. If yes, then let $o = \infty$ and cancel the witnesses v_k for every $k : 1 \leq k \leq n$. Otherwise let v_k be the witness targeting X_{e_k} (if v_k is not defined or has been cancelled, then pick it fresh). Check if for all k with $1 \leq k \leq n$, $\Psi_{i_k}(v_k) \downarrow = 0[s]$. If no, then let $o = 1$; if yes, starting the process of clearing Δ for the outcome 0. When the clearing process is done, put a into A and set a link of the form (σ, α).

(5) α is labelled $T^k_{e,i}$. Check if s is an α-expansionary stage. If yes, let $o = \infty$, otherwise let $o = 0$.

At the end of the stage, we *initialise* all nodes to the right of TP_s.

This finishes the construction.

2.6. **Verification.** We now verify that the construction works. We begin with the lemma showing that the true path exists.

Lemma 2.1. *For any $e \in \omega$, there is a unique node α on T such that α is the leftmost one of length e which is accessible (and not covered by any link) infinitely often.*

Proof. We do an induction on e. Suppose true for e. Let α be the leftmost string of length e which is accessible infinitely often. We need discuss the cases involving links, as other cases are routine. Let s_0 be the least stage after which α is never initialised.

First we show that we do not stop at α forever. Observe that we stop at α at stage s only when α is accessible at s and we set up a link of the form (β, α) for some node β and end the stage.

At the stage $t_0 > s$, when α is accessible again, the link is travelled hence get cancelled.

Case 1. If the link was set because some N-requirement stops α having outcome o, then the link can only transfer into another link (β', α) for some $\beta' \subset \beta$. Since there are only finitely many such β's, eventually α will be accessible and no link of the form (β, α) exists.

Case 2. If α is labelled with an S-requirement, then at any stage $t > t_0$ when α is accessible again, it will have outcome 0.

Secondly, we argue that we do not skip any node. The concern is: When α is accessible at some stage $s > s_0$, we travel some link of the form (α, τ) to some node τ, instead of having an outcome o. By construction, once a link is travelled, it gets cancelled. Thus when α is accessible again, there will be no link of the form (α, τ), we will access $\alpha\hat{\ }o$ for some o.

This finishes the proof of Lemma 2.1. □

Let TP be the *true path* in T, that is, TP is the leftmost path which is accessible infinitely often. By Lemma 2.1, TP exists.

We argue by induction along TP that every requirement is satisfied. We split the proof into two lemmas.

Lemma 2.2. *Let α be a node on TP and O be the label of α. Then*

(a) *Suppose that O is N_e. Then $\alpha\hat{\ }\infty \subset TP$ if and only if there are infinitely many α-expansionary stages. Let β be node on TP labelled $M_{e,p}$ working for α. Then the flip point d at β is eventually fixed. Moreover,*
  (a1) *if $\beta\hat{\ }0 \subset TP$, then $\Delta_e(W_{e_2}; p)$ is defined;*
  (a2) *if $\beta\hat{\ }\infty \subset TP$, then $\Gamma_{e_1}(AW_{e_2}; p)\uparrow$.*

(b) *Suppose that O is R_e. Then $\alpha\hat{\ }\infty \subset TP$ if and only if there are infinitely many α-expansionary stages. Let β (γ respectively) be the node on TP labelled $S_{e,i}$ ($T^k_{e,i}$ respectively) working for α. Also assume that $e = \langle e_0, e_1, \dots, e_n\rangle$, $B = B_{e_1} \oplus B_{e_2} \oplus \cdots \oplus B_{e_n}$ and S consists of the positive requirements of the form: $X_{e_k} \neq \Psi_{i_k}$ for $1 \leq k \leq n$. Then*
  (b1) *if $\beta\hat{\ }\infty \subset TP$ then $\Phi_{e_0}(B) \neq A$, where $B = B_{e_1} \oplus \cdots \oplus B_{e_n}$;*
  (b2) *if $\beta\hat{\ }1$, or $\beta\hat{\ }0 \subset TP$, then $\Psi_{i_k} \neq X_{e_k}$ for some $1 \leq k \leq n$.*
  (b3) *$\gamma\hat{\ }\infty \subset TP$ if and only if there are infinitely many γ-expansionary stages.*

Proof. We prove (a) and (b) by simultaneous induction. We begin with statement (a).

Suppose that α is labelled N_e. If $\alpha\hat{\ }\infty \subset$ TP, then obviously there are infinitely many α-expansionary stages. Suppose that $\alpha\hat{\ }0 \subset$ TP and s is the stage after which no nodes to the left of $\alpha\hat{\ }0$ are accessible. If there is an α-expansionary stage t after s, then after clearing Δ for the outcome ∞ at α, say at $t' > t$, we still make $\alpha\hat{\ }\infty$ accessible even if t' is not α-expansionary. Thus at stage t', $\alpha\hat{\ }\infty$ is accessible, contradicting the choice of s. Hence there are only finitely many α-expansionary stages.

Let β be the node on TP labelled $M_{e,p}$ working for α. As β only changes its flip point when it is initialised, the flip point d at β is eventually fixed.

We now prove statement (a1). Suppose that $\beta\hat{}0 \subset$ TP. Let s be the stage at which $\beta\hat{}0$ is accessible and after which no node to the left of $\beta\hat{}0$ is accessible. Then by construction, either $\Delta_e(W_{e_2}; p)$ has been defined before s or we defined it at stage s with use $\gamma_{e_1}(AW_{e_2}; d)[s]$. As $\gamma_{e_1}(AW_{e_2}; d)[s]$ is fixed (otherwise, $\alpha\hat{}\infty$ would be accessible), $\Delta_e(W_{e_2}; p)$ will never be injured after stage s, since its use will be of the form $\gamma_{e_1}(AW_{e_2}; d')$ for some $d' < d$. This establishes statement (a1). Statement (a2) follows from the condition of β having outcome ∞.

We now prove statement (b). If there is no confusion, the index e is dropped.

The argument about R_e is similar to the one about N in (a).

Suppose $\alpha\hat{}\infty \subset$ TP. Let β be the node on TP labelled $S_{e,i}$ as in the statement of the theorem. Fix a stage s_0 after which no node to the left of β is accessible and β is never initialised. Thus the agitator a is fixed.

We begin with (b1). Suppose $\beta\hat{}\infty \subset$ TP. By construction, β has outcome ∞ only when $\varphi_{e_0}(B; a)$ moved. Thus, $\Phi_{e_0}(B)$ is partial, hence not equal to A.

Next we consider (b2). Suppose $\beta\hat{}0 \subset$ TP. Clearly we have $\Psi_{i_k}(v_k) = 0$ and $v_k \in X_{e_k}$ at some stage for some k with $1 \leq k \leq n$. Since these fact can never be injured, we have $\Psi_{i_k}(v_k) \neq X_{e_k}$. Suppose $\beta\hat{}1 \subset$ TP, then $\Psi_{i_k}(v_k) \neq 0$ and $v_k \notin X_{e_k}$ for some k with $1 \leq k \leq n$. Again we have $\Psi_{i_k} \neq X_{e_k}$.

Statement (b3) follows from the definition of $\gamma\hat{}\infty$ being accessible. $\square$

Finally we show that all requirements are satisfied.

Lemma 2.3. *For each e in ω the noncuppable requirement N_e and bounding requirement R_e are satisfied.*

Proof. First we argue that the requirements N_e are satisfied.

Suppose that there is a c.e. set W_{e_2} such that $A \oplus W_{e_2}$ is complete. Since the set D, which we built, is c.e., there is a Turing functional Γ_{e_1} such that $\Gamma_{e_1}(AW_{e_2}) = K$. Let e be the code of the pair $\langle e_1, e_2\rangle$ and α be the unique node labelled N_e on TP. We show that $\Delta_e(W_{e_2})$ is total and equal to K_0.

Fix a stage s_0, after which α never gets initialised. Clearly, since $\Gamma_{e_1}(AW_{e_2}) = K$, we know $\alpha\hat{}\infty$ is on TP. We consider two cases based on whether N_e has a global Σ_3-outcome.

Case 1. There is a node β on TP labelled $M_{e,p}$ working for α, such that $\beta\hat{}\infty \subset$ TP.

Then by statement (a2) in Lemma 2.2, $\Gamma_{e_2}(AW_{e_1}; p)$ is undefined, contradicting to $\Gamma_{e_1}(AW_{e_2}) = K$. Thus case 1 is vacuous.

Case 2. For all nodes β_p on TP labelled $M_{e,p}$ working for α, $\beta_p\hat{}0 \subset$ TP.

In this case, by statement (a1) in Lemma 2.2, the Turing functional $\Delta_e(W_{e_2})$ is total.

Fix a number p. We argue $\Delta_e(W_{e_2}; p)$ is equal to $\Gamma_{e_1}(AW_{e_2}; p)$. Let s be the stage after which no node to the left of β_p is accessible. As argued in the proof of statement (a1), $\Delta_e(W_{e_2}; p)$ will not change after s. Let s^- be the last stage before s at which we define $\Delta_e(W_{e_2}; p)$, say at node β^-. Then at stage s^-, $\Delta_e(W_{e_2}; p)[s^-] = \Gamma_{e_1}(AW_{e_2}; p)[s^-]$. After s^-, no node to the left of β^- is accessible, otherwise this Δ

would be cleared. Therefore, the flip point d^- at β^- is fixed and the finite restraint of amount $\gamma_{e_1}(AW_{e_2}; d^-)[s^-]$ is permanent on A. By the assumption on s^-, W_{e_2} will not change below $\gamma_{e_1}(AW_{e_2}; d^-)[s^-]$ after s^-. Therefore

$$\begin{aligned}\Delta_e(W_{e_2}; p) &= \Delta_e(W_{e_2}; p)[s^-] \\ &= \Gamma_{e_1}(AW_{e_2}; p)[s^-] \\ &= \Gamma_{e_1}(AW_{e_2}; p).\end{aligned}$$

Thus N_e is satisfied.

Let us consider the bounding requirements. Given any c.e. sets B_{e_k} $(1 \leq k \leq n)$ and $B = B_{e_1} \oplus \cdots \oplus B_{e_n}$, suppose that $A \leq_{\mathrm{T}} B$ and $\Phi_{e_0}(B) = A$. Consider the requirement R_e where $e = \langle e_0, e_1, ..., e_n\rangle$. Let α be the unique node on TP labelled R_e. Then by statement (b) in Lemma 2.2, $\alpha\hat{\ }\infty \subset$ TP. Furthermore, for any S-node $\beta \subset$ TP working for α, $\beta\hat{\ }\infty \not\subset$ TP (otherwise by (b1) in Lemma 2.2, $\Phi_{e_0}(B)$ would be partial).

We argue that for some k with $1 \leq k \leq n$ both X_{e_k} and Y_{e_k} are not computable. Suppose not, i.e., for all k with $1 \leq k \leq n$, one of X_{e_k} and Y_{e_k} is computable. Then for all k such that $1 \leq k \leq n$, there is i_k such that P^k_{e,i_k} is unsatisfied. Let $i = \langle i_1, \ldots, i_n\rangle$, let β be the node labelled $S_{e,i}$ on true path. Since we have either $\beta\hat{\ }0$ or $\beta\hat{\ }1 \subset$ TP, by (b2) we have one of X_{e_k} or Y_{e_k} is not computable, a contradiction.

We now show that $T^k_{e,i}$ is satisfied. Suppose that $\Theta_i(X_{e_k}) = \Theta_i(Y_{e_k}) = f$ and f is total. We show that f is computable. Let γ be a node on TP labelled $T^k_{e,i}$. By Lemma 2.2, $\gamma\hat{\ }\infty \subset$ TP. Let s_0 be a stage after which $\gamma\hat{\ }\infty$ is never initialised. Fix z, we compute $f(z)$ as follows. Wait for the first stage $s > s_0$ such that $\gamma\hat{\ }\infty$ is accessible at stage s and $l(\gamma; s) > z$. We claim that $f(z) = \Theta_i(X_{e_k}; z)[s]$. It suffices to show that for any $t > s$, one of $\Theta_i(X_{e_k}; z)[t]$ and $\Theta_i(Y_{e_k}; z)[t]$ is equal to $\Theta_i(X_{e_k}; z)[s]$. Any node below or to the left of $\gamma\hat{\ }\infty$ cannot act; any node to the right of $\gamma\hat{\ }\infty$ will obey the finite restraint imposed by γ. The only worry is some S-node β above $\gamma\hat{\ }\infty$ working for the same α and β wants to put a number, say v, into one side, say X_{e_k}. Thus v must be ready at an γ-expansionary stage at which we set a link between β and α. Before the next α-expansionary stage, $l(\gamma)$ remains unchanged. At next α-expansionary stage, we only put v into X_{e_k} and end the stage, thus $\Theta_i(Y_k; z)$ remains.

It remains to argue that X_{e_k} and Y_{e_k} are both computable from B_{e_k}. Without loss of generality, we only prove $X_{e_k} \leq_{\mathrm{T}} B_{e_k}$. Let s_0 be the stage after which α labelled R_e is never initialised. Fix a number z. First notice that we can compute whether or not z is chosen as a witness targeting X_{e_k}: We just wait for a stage at which some number $z' > z$ appeared in the construction; if by that stage z has not been chosen as a witness, then z will never be, since we always choose witness fresh. If z is not chosen as a witness targeting X_{e_k}, then z will not be in X_{e_k}. Without loss of generality, let us assume that z is a witness chosen at some node β. Notice that if z enters X_{e_k} at stage s, then by construction we must pass stage a t at which we set up the link (α, β), and then we travel the link at stage s.

To see if $z \in X_{e_k}$ from B_{e_k}, we wait for the (least) stage t_0 at which $B_{e_k,t_0} \upharpoonright z = B_{e_k} \upharpoonright z$, and let t_1 be the first stage larger than t_0 at which α is accessible. If there is no link of the form (α, β), then $z \in X_{e_k}$ if and only if $z \in X_{e_k,t_1}$. If there is a link, then $z \in X_{e_k}$ if and only if z enters X_{e_k} by the end of next α-expansionary stage. This ends all verification. $\square$

3. Proof of Theorem 1.6

Fix computable bijections $\langle e_0, \dots, e_n \rangle \mapsto e$ from $\omega^{<\omega}$ to ω and a complete c.e. set K. Let I be the ideal generated by noncuppable and nonbounding c.e. degrees. We prove Theorem 1.6 by constructing a c.e. set A whose degree is cappable but not in I.

To make A cappable, we build its companion degree C such that A and C form a minimal pair. By absorbing the noncomputable requirement of A into the requirement R_e, we have the following typical minimal pair requirements:

- Q_e: $C \neq \overline{W}_e$.
- N_e: If $\Xi_e(A) = \Xi_e(C)$ is total, then $\Xi_e(C)$ is computable.

Since NCup is an ideal, to ensure A not in I, it suffices to make A not below the join of one noncuppable and finitely many nonbounding degrees:

- R_e: If $A = \Phi_{e_0}(D_{e_1} \oplus B_{e_2} \oplus \cdots \oplus B_{e_n})$ where $e = \langle e_0, e_1, \dots, e_n \rangle$, then we build c.e. sets X_{e_k} and $Y_{e_k} \leq_{\mathrm{T}} B_{e_k}$ $(2 \leq k \leq n)$ such that either one pair of them forms a minimal pair or D_{e_1} is cuppable.

3.1. **Description of strategies.** The strategies for Q_e and N_e are the normal ones for building minimal pairs. We concentrate on the strategy for R. For notation simplicity, we use $n = 2$ for illustration, we also drop the index e if there is no confusion.

Let α be a node labelled R_e. We first measure the length of agreement between $\Phi(DB)$ and A, and define the α-expansionary stage as usual. If the current stage is α-expansionary, then we have outcome ∞, otherwise have outcome 0. Let us assume that ∞ is the true outcome, otherwise it is trivial. We build c.e. sets X and Y such that both are computable from B by permitting and have infimum $\mathbf{0}$ by the following subrequirement $M_{e,i}$.

- $M_{e,i}$: If $\Theta_i(X_e) = \Theta_i(Y_e)$ is total, then $\Theta_i(X_e)$ is computable.

We also attempt to make both X and Y noncomputable; when the attempt fails we will demonstrate that D is cuppable by building a functional $\Gamma_{e,i}$ and a c.e. set $E_{e,i}$ such that $\Gamma_{e,i}(DE_{e,i}) = K$ and $E_{e,i}$ is incomplete. Thus we have the subrequirement $S_{e,i}$ for all i.

- $S_{e,i}$: $P_{e,i}$ is satisfied (via some positive action by some strategy $T_{e,i,j}$ working for $S_{e,i}$), or $\Gamma_{e,i}(DE_{e,i}) = K$ and for all j, $\Lambda_j(E_{e,i}) \neq K$, where $P_{e,i}$ is the noncomputable subrequirements for X and Y:
- $P_{e,2i}$: $X \neq \Psi_i$,
- $P_{e,2i+1}$: $Y \neq \Psi_i$.

Let β be a node labelled $S_{e,i}$. We build the functional $\Gamma_{e,i}$ and the c.e. set $E_{e,i}$ at β. To ensure the correctness of $\Gamma_{e,i}(DE_{e,i}) = K$, whenever we see a (least) disagreement $\Gamma_{e,i}(DE_{e,i})(m) \neq K(m)$ at β, we put the use $\gamma(m)$ into $E_{e,i}$ to redefine Γ. β has two outcome 0 for winning $P_{e,i}$; and ∞ for building Γ.

Above $\beta\hat{}\infty$, we have subrequirements $T_{e,i,j}$ for all $j \in \omega$.

- $T_{e,i,j}$: Either $P_{e,i}$ is satisfied by default or $\Lambda_j(E_{e,i}) \neq K$.

To make $\Lambda_j(E_{e,i}) \neq K$, we measure $l(s)$ which is the length of agreement between $\Lambda_j(E_{e,i})$ and K at stage s. We try to lift the use $\gamma(j)$ (hence all markers $\gamma(m)$ for $m \geq j$) beyond $\lambda_j(l)$. Whenever $\gamma(j)$ is less than $\lambda(l)$ (this will happen, for example, when l gets increased), we try to force a D-change below $\gamma(j)$. We will ensure that if we do not win $P_{e,i}$, then D must change below $\gamma(j)$. Thus the use $\gamma(j)$ can be lifted without putting numbers into E. In the end, if $\gamma(j)$ still goes to infinity, then we would have that $E_{e,i}$ is computable and $\Lambda_j(E_{e,i})$ is total, consequently K would be computable, which is a contradiction. The argument is similar to the one in the proof of plus-cupping theorem by Fejer and Soare [2].

The strategy for $T_{e,i,j}$ acts in cycles. The action in each cycle m is as follows:

(1) Choose a fresh agitator $a \notin A$.
(2) Wait for a stage s at which $\Phi_{e_0}(D_{e_1}B_{e_2}; a)$ is defined.
(3) Assume that $P_{e,i}$ is $X \neq \Psi_i$. Select a number $v \notin X$ and $v > \varphi_{e_0}(D_{e_1}B_{e_2}; a)$, v will be the witness targeting X. Pick a fresh number z, which will be used by the strategy for making D cuppable; more specifically, we try to move the marker $\gamma(z)$ beyond $\lambda_j(l)$ where l is the length of agreement between $\Lambda_j(E_{e,i})$ and K. Initially $\gamma(z) > \varphi(DB; a)$. Wait for a stage s at which l is larger than $\gamma_{e,i}(D_{e_1}E_{e,i}; z) + 1$ and $\Psi_i(v) \downarrow= 0$. From now on, once $\varphi_{e_0}(DB; a)$ moves, we start over by resetting v and z, putting $\gamma_{e,i}(D_{e_1}E_{e,i}; z)$ into $E_{e,i}$ and back to the beginning of this step.
(4) At stage s, put a into A and setup a link from R_e to $T_{e,i,j}$.
(5) At the next α-expansionary stage, we travel the link to T. If B has changed, then we put v into X and $S_{e,i}$ will have outcome 1 forever; if D has changed, then we close the cycle m.

We now analyse the outcomes:
Case 1: For some m, we stuck in the cycle m forever.

- If $\varphi(DB; a)$ keeps moving, then $\Phi_{e_0}(D_{e_1}B_{e_2}; a) \uparrow$, we have a global win for R_e, we use ∞ to denote this outcome.
- If we stuck at (3) forever waiting for either $\Psi_i(v) \downarrow= 0$ or the length of agreement l gets beyond $\Gamma_{e,i}(D_{e_1}E_{e,i}; z)$, then we either win $S_{e,i}$ by default or have $\Lambda_j(E_{e,i}) \neq K$. we use 0 for this outcome.

Case 2: For every m, we close the cycle m. We argue that the case is vacuous.

First notice that $E_{e,i}$ will be computable: After we close the cycle m, $E_{e,i} \upharpoonright \gamma(z)$ will not change. Secondly, $\Lambda_j(E_{e,i})$ is total: In each cycle, the marker $\gamma(z)$ was lifted over at least one $\lambda_j(E_{e,i}; x)$ for some new value x. Finally, the length of agreement between $\Lambda_j(E_{e,i})$ and K goes to infinity. Thus, K is computable, which is a contradiction.

3.2. **Construction.** We restate the requirements R_e in the general setting for the sake of notations.

Let $\langle e_0, \ldots, e_n\rangle \mapsto e$ be a fixed computable bijection from $\omega^{<\omega}$ to ω. Fix effective enumerations of c.e. sets D_e and B_e ($e \in \omega$) and Turing functionals Φ_e, Ψ_e, Θ_e and

Λ_e ($e \in \omega$). The requirements and subrequirements R_e, $M_{e,i}$, $S_{e,i}$ and $T_{e,i,j}$ are as follows.

- R_e: If $A = \Phi_{e_0}(D_{e_1}B_{e_2}\dots B_{e_n})$ where $e = \langle e_0, e_1, \dots, e_n\rangle$, then there are c.e. sets X_{e_k} and $Y_{e_k} \leq_{\mathrm{T}} B_{e_k}$ ($1 \leq k \leq n$) such that for all i, the subrequirements $M^k_{e,i}$ and $S_{e,i}$ are satisfied.
- $M^k_{e,i}$: If $\Theta_i(X_{e_k}) = \Theta_i(Y_{e_k})$ is total, then $\Theta_i(X_{e_k})$ is computable.
- $S_{e,i}$: P^k_{e,i_k} is satisfied for some $2 \leq k \leq n$ where $i = \langle i_2, ...i_n\rangle$ and $P^k_{e,i}$ is the noncomputable subrequirements for X_{e_k} and Y_{e_k}:
  - $P^k_{e,2i}$: $X_{e_k} \neq \Psi_i$,
  - $P^k_{e,2i+1}$ $Y_{e_k} \neq \Psi_i$;

  or there is a c.e. set $E_{e,i}$ and functional $\Gamma_{e,i}$ such that $\Gamma_{e,i}(D_{e_1}E_{e,i}) = K$ and for all j, the subrequirement $T_{e,i,j}$ is satisfies.
- $T_{e,i,j}$: $\Lambda_j(E_{e,i}) \neq K$.

We now describe the priority tree T. Fix a computable priority list of the requirements and subrequirements such that the subrequirements appear after the main ones. We label T inductively in the usual manner. The root node on T is labelled Q_0. Suppose that τ is a node on T. If τ is labelled Q_e then τ has two outgoing edges labelled 0 and 1; otherwise, that is, if τ is labelled N_e, R_e, $M^k_{e,i}$, $S_{e,i}$, or $T_{e,i,j}$, then τ has two outgoing edges labelled ∞ and 0, with ∞ to the left of 0.

We say that a requirement R_e is *satisfied* at τ if there is a node $\alpha \subset \tau$ labelled R_e such that either $\alpha\hat{}0 \subseteq \tau$ or there is an η labelled with a subrequirement $T_{e,i,j}$ working for R_e such that $\eta\hat{}\infty \subseteq \tau$. *Namely, we see a global win for R_e at η.* If R_e is satisfied at τ then all its subrequirements are satisfied at τ.

Continuing the inductive definition of T, if all $\alpha \subset \tau$ have been labelled, then τ is labelled with the highest priority O such that O is either a requirement which never appeared before or O is a unsatisfied new subrequirement.

Let α be a node on T. We now list a collection of parameters related to α. We may also drop the indices of the requirements, if there is no confusion.

(1) If α is labelled Q_e, then it has a witness x targeting C.
(2) If α is labelled N_e, then it has its length of agreement l and a finite restraint r to preserve l.
(3) If α is labelled R_e, then no parameter is needed.
(4) If α is labelled $M^k_{e,i}$, then it has its length of agreement l and a finite restraint r to preserve l.
(5) If α is labelled $S_{e,i}$, then it has no parameters.
(6) If α is labelled $T_{e,i,j}$, then it has the following parameters.
   - a node σ labelled R_e, for which it is working. we will call σ the *head of* α.
   - a number a, called an *agitator*, which will be used to seek an permission from $B = D_{e_1} \oplus B_{e_2} \oplus \cdots \oplus B_{e_n}$;
   - a number m indicating that we are in cycle m.
   - a number r which is the finite restraint imposed by the M-nodes which are working for the same R and below or to the left of α. Although those M-nodes may put different restraints $r^X_{e_k}$ or $r^Y_{e,k}$ on different sets X_{e_k} or

$Y_{e,k}$. It does no harm if we take r to be the maximal of those small restraints.
- a set of $n-1$ numbers $v_k > \varphi^B(a)$ where $v_k > r$ for every $2 \leq k \leq n$.
- a number z whose γ-use will be lifted beyond $\lambda_j(l)$ where l is the length of agreement between $\Lambda_j(E_{e,i})$ and K.

We now describe the stage by stage construction. At stage s, we first specify a string TP_s of length less than or equal to s, called the *accessible string*, then act along the accessible string.

We define the accessible string inductively from the root. The root of the tree is always accessible.

At the inductive step, suppose that the node α is accessible. If the length of α is equal to s then we let $\alpha = \mathrm{TP}_s$ and go to the next stage. Also, we may declare that we stop the stage, in the case when we put elements into one side of the minimal pair.

Suppose that the length of α is less than s. Then we determine the outcome o of α and let $\alpha\hat{\ }o$ be accessible and take actions based on the label of α as follows.

(1) α is labelled Q_e.

If there is a number $x \in C \cap W_e$, then let $o = 1$ and take no action.

Let x be the witness (if x is undefined, then pick it fresh), if $x \in W_{e,s}$, then put x into C. Go to next stage. If $x \notin W_{e,s}$, then let $o = 0$.

(2) α is labelled N_e.

If s is a α-expansionary stage, then set $r = 0$ and let $o = \infty$; otherwise, set $r = t$ which is the stage when $\alpha\hat{\ }\infty$ was accessible for the last time and let $o = 0$.

(3) α is labelled R_e.

Case 1. There is a link of the form (α, τ).

Then τ must be labelled with a T-requirement, say $T_{e,i,j}$. Check if the stage s is α-expansionary.

- If no, then let $\alpha\hat{\ }0$ be accessible.
- If yes, then go to τ and cancel the link (α, τ). Now τ must be in half way of some cycle, say cycle m; declare that the cycle m is closed. Observe that either D_{e_1} or one of the B_{e_k} $(2 \leq k \leq n)$ must have changed below $\varphi(B; a)$ where $B = D_{e_1} \oplus B_{e_2} \oplus \cdots \oplus B_{e_n}$ and a is the agitator; and the witnesses v_k $(1 \leq k \leq n)$ are all available. If D_{e_1} has changed then do nothing, let $o = \infty$; otherwise, that is one of the B_{e_k} has changed, choose the least such k enumerate v_k into the set it was targeting. Go to the next stage.

Case 2. There is no link of the form (α, τ).

Check if s is an α-expansionary stage. If yes, then let $o = \infty$; otherwise, let $o = 0$.

(4) α is labelled $M^k_{e,i}$. Check if s is an α-expansionary stage. If yes, set $r = 0$ and let $o = \infty$, otherwise set $r = t$ which is the stage when $\alpha\hat{\ }\infty$ was accessible for the last time and let $o = 0$.

(5) α is labelled $S_{e,i}$. Without loss generality, let us assume that S consists of the positive requirements of the form: $X_{e_k} \neq \Psi_{i_k}$ for $1 \leq k \leq n$. Let σ denote its head.

First check if $S_{e,i}$ has been *satisfied*, that is, there was a stage $t < s$ at which we put a number v_k into X_{e_k} for some $2 \leq k \leq n$ at some $T_{e,i,j}$ working for $S_{e,i}$.

- If yes, then let $o = 0$.
- Otherwise, check if there is x such that $\Gamma_{e,i}(D_{e_1}E_{e,i}; x) \neq K(x)[s]$. If yes, then choose the least one and put $\gamma_{e,i}(D_{e_1}E_{e,i}; x)$ into $E_{e,i}$. Redefine $\Gamma_{e,i}(D_{e_1}E_{e,i}; y) = K(y)[s]$ for every y with $x \leq y \leq s$ with fresh use. Let $o = \infty$.

(6) α is labelled $T_{e,i,j}$. Let t be the last stage at which α was accessible for the last time since it was initialised.

If s is the first stage that α is accessible since the last time it was initialised, or we closed a cycle m at stage t, then we *open a new cycle* $m+1$ by choosing a fresh agitator a.

Otherwise, that is, we are in the middle of some cycle, say cycle m. Check if $\Phi_{e_0}(B; a)$ is undefined or $\varphi_{e_0}(B; a)[s] \neq \varphi_{e_0}(B; a)[t]$, where $B = D_{e_1} \oplus B_{e_2} \oplus \cdots \oplus B_{e_n}$. If yes, then let $o = \infty$, cancel the witnesses v_k for every k with $2 \leq k \leq n$ and put $\gamma_{e,i}(D_{e_1}E_{e,i}; z)$ into $E_{e,i}$. Otherwise, if the witness v_k $(2 \leq j \leq k)$ and z have not been chosen, the select them fresh, in particular $v_k \notin X_{e_k}$ and $v_k > \varphi_{e_0}(B; a)$; and since $\Gamma_{e,i}(D_{e_1}E_{e,i}; z)$ will be defined later, $\gamma_{e,i}(D_{e_1}E_{e,i}; z) > \varphi_{e_0}(B; a)$. (*This is to guarantee that by putting a into A, we would either see a permission of v_k entering X_{e_k} or can lift $\gamma(z)$ to a new position by the D_{e_1}-change.*)

Let l denote the length of agreement between $\Lambda_j(E_{e,i})$ and K, and $\lambda_j(l)$ be the use. If $\Psi_{i_k}(v_k) \downarrow = 0$ for all $2 \leq k \leq n$ and $\gamma_{e,i}(D_{e_1}E_{e,i}; z) < \lambda_j(l)$, then we put a into A, set up a link (σ, α), where σ is the head of α and go to the next stage. Otherwise, let $o = 0$ and do nothing.

At the end of the stage, we *initialise* all nodes to the right of TP_s.

This finishes the construction.

3.3. **Verification.**

We now verify that the construction works.

Lemma 3.1. *For any $e \in \omega$, there is a unique node α on T such that α is the leftmost one of length e which is accessible (and not covered by any link) infinitely often.*

Proof. We do an induction on e. By a similar argument as in the proof of Lemma 2.1, the link will not give us any problem. The only worry is that: At a node labelled $T_{e,i,j}$, we may pass through all cycles, thus always stop the construction at $T_{e,i,j}$. As we shall see from the inductive argument in the next lemma, passing through all cycles imply that K is computable, which is a contradiction. □

Let TP be the collection of all such α, which we call the *true path* in T.

Lemma 3.2. *Let α be a node on TP and O be the label of α. Then*

(a) *Suppose that O is Q_e. Then $\alpha\hat{}1 \subset TP$ if and only if $x \notin C$ and $x \in \overline{W_e}$; $\alpha\hat{}0 \subset TP$ if and only if $x \in C \cap W_e$.*

(b) *Suppose that O is N_e. Then $\alpha\hat{}\infty \subset TP$ if and only if there are infinitely many α-expansionary stages.*

(c) *Suppose that O is R_e. Then $\alpha\hat{}\infty \subset TP$ if and only if there are infinitely many α-expansionary stages. Let β (δ, η respectively) be the node on TP labelled $S_{e,i}$ ($T_{e,i,j}$, $M^k_{e,i}$ respectively) working for α. Also assume that $e = \langle e_0, e_1, \ldots, e_n\rangle$, $B = D_{e_1} \oplus B_{e_2} \oplus \cdots \oplus B_{e_n}$ and S consists of the positive requirements of the form: $X_{e_k} \neq \Psi_{i_k}$ for $2 \leq k \leq n$. Then*
  (c1) *if $\beta\hat{}0 \subset TP$ then there is k with $2 \leq k \leq n$ such that $\Psi_{i_k} \neq X_{e_k}$;*
  (c2) *the parameter m at δ is eventually fixed, in other words, we will stay in some cycle m forever; moreover if $\delta\hat{}\infty \subset TP$, then $\Phi_{e_0}(B) \neq A$; if $\delta\hat{}0 \subset TP$, then either $\Psi_{i_k} \neq X_{e_k}$ for some $2 \leq k \leq n$ or $\Lambda_j(E_{e,i}) \neq K$.*
  (c3) *$\eta\hat{}\infty \subset TP$ if and only if there are infinitely many η-expansionary stages.*

Proof. We prove by simultaneous induction. We only prove statement (c2), since other parts follow from the construction easily.

Fix a stage s_0 after which no node to the left of δ is accessible and δ is never initialised. Thus the parameter z is fixed. Let s_1 be a stage after which for all $z' < z$, $\gamma(z')$ stops moving; such s_1 exists by induction.

Suppose we pass through cycle m for every m. When we close a cycle, $\Gamma_{e,i}(D_{e_1}E_{e,i}; z)$ (denoted by $\Gamma(z)$) is undefined because of D_{e_1}-change; when it gets redefined it becomes bigger. Therefore $\gamma(z)$ goes to infinity. After stage s_1, $E_{e,i} \upharpoonright \gamma(z)$ never changes, therefore $E_{e,i}$ is computable. Furthermore when we close a cycle, the length of agreement l between $\Lambda_j(E_{e,i})$ and K is increased, in particular, $\Lambda_j(E_{e,i} \upharpoonright l$ is defined, and $\gamma(z)$ is lifted over its use, thus $\Lambda_j(E_{e,i})$ is total. Consequently K is computable, a contradiction. Therefore, we will eventually stay in some cycle m; hence the agitator a is eventually fixed. Clearly if $\delta\hat{}\infty \subset$ TP, then $\varphi_{e_0}(B; a)$ keeps moving. Thus, $\Phi_{e_0}(B)$ is partial, hence not equal to A. Suppose $\delta\hat{}0 \subset$ TP. Then we either wait for $\Psi_{i_k}(v_k) = 0$ for some $2 \leq k \leq n$ or wait for the length of agreement between $\Lambda_j(E_{e,i})$ and K being larger than $\varphi(B; a)$. Hence either we have $\Psi_{i_k}(v_k) \neq X_{e_k}$ or $\Lambda_j(E_{e,i}) \neq K$. □

Finally we show that all requirements are satisfied.

Lemma 3.3. *For each e in ω requirement Q_e, N_e and R_e are satisfied.*

Proof. We only argue that the requirements R_e is satisfied, since the other two are the same as the minimal pair argument.

Given any c.e. sets D_{e_1} and B_{e_k} ($2 \leq k \leq n$) and $B = D_{e_1} \oplus \cdots \oplus B_{e_n}$, suppose that $A \leq_{\mathrm{T}} B$ and $\Phi_{e_0}(B) = A$. Consider the requirement R_e where $e = \langle e_0, e_1, ..., e_n\rangle$. Let α be the unique node on TP labelled R_e. Then by statement (b) in Lemma 2.2, $\alpha\hat{}\infty \subset$ TP. Furthermore, for any T-node $\delta \subset$ TP working for α, $\delta\hat{}\infty \not\subset$ TP (otherwise $\Phi_{e_0}(B)$ would be partial).

Suppose that for all k with $2 \leq k \leq n$ both X_{e_k} and Y_{e_k} are computable. Say the fact is realized by $i = \langle i_2, \ldots, i_n\rangle$. Thus the requirements $S_{e,i}$ and $T_{e,i,j}$ for all j are never satisfied by winning the positive requirement P^k_{e,i_k}. We argue that the functional $\Gamma_{e,i}$ built at β is total. Fix any number p, the marker $\gamma(p)$ can only be pushed by finitely many δ-nodes. We may ignore those located to the left or right of true path. For those located on true path, by Lemma 3.2, we will stay in some fixed cycle, and δ must have outcome 0. Thus there is no movement of the marker eventually. By construction, we always correct the value $\Gamma_{e,i}(D_{e_1}E_{e,i})(p)$ to be $K(p)$, thus $\Gamma_{e,i}(D_{e_1}E_{e,i}) = K$. By Lemma 3.2, $\Lambda_j(E_{e,i}) \neq K$ for all j, hence $E_{e,i}$ is incomplete; in other words, D_{e_1} is cuppable.

The remaining part of the proof is the same as in Lemma 2.2.

This ends all verification. □

References

[1] Klaus Ambos-Spies and Robert I. Soare. The recursively enumerable degrees have infinitely many one-types. *Ann. Pure Appl. Logic*, 44(1-2):1–23, 1989. Third Asian Conference on Mathematical Logic (Beijing, 1987).

[2] P. A. Fejer and Robert I. Soare. The plus-cupping theorem for the recursively enumerable degrees. In *Logic Year 1979–80: University of Connecticut*, pages 49–62, 1981.

[3] Angsheng Li, Theodore A. Slaman and Yue Yang. A nonlow$_2$ c.e. degree which bounds no diamond bases. To appear.

[4] André Nies. Parameter definability in the r.e. degrees. To appear.

[5] André Nies, Richard A. Shore, and Theodore A. Slaman. Definability in the recursively enumerable degrees. *Bull. Symbolic Logic*, 2(4):392–404, 1996.

[6] André Nies, Richard A. Shore, and Theodore A. Slaman. Interpretability and definability in the recursively enumerable degrees. *Proc. London Math. Soc. (3)*, 77(2):241–291, 1998.

[7] Richard A. Shore. Natural definability in degree structures. In *Computability theory and its applications (Boulder, CO, 1999)*, pages 255–271. Amer. Math. Soc., Providence, RI, 2000.

[8] Robert I. Soare. *Recursively Enumerable Sets and Degrees.* Springer–Verlag, Heidelberg, 1987.

Department of Mathematics, Faculty of Science, National University of Singapore, Lower Kent Ridge Road, Singapore 119260.

E-mail address: matyangy@leonis.nus.edu.sg

Department of Mathematics, Nanjing University, Nanjing, Jiang Su, China

E-mail address: yuliang@mcs.vuw.ac.nz