ILLINOIS JOURNAL OF MATHEMATICS
Volume 44, Number 3, Winter 2000

A HIERARCHY FOR CUPPABLE DEGREES

ANGSHENG LI, GUOHUA WU AND ZAIYUE ZHANG

ABSTRACT. We say that a *computably enumerable* (c.e., for short) degree $\mathbf{a}$ is *cuppable* if there is a c.e. degree $\mathbf{b} \neq \mathbf{0}'$ such that $\mathbf{a} \vee \mathbf{b} = \mathbf{0}'$. A c.e. degree $\mathbf{a}$ is called *low$_n$-cuppable*, $n > 0$, if there is a low$_n$ c.e. degree $\mathbf{l}$ such that $\mathbf{a} \vee \mathbf{l} = \mathbf{0}'$. Let $\mathbf{LC}_n$ be the set of all low$_n$-cuppable c.e. degrees. In this paper, we show that $\mathbf{LC}_1 \subset \mathbf{LC}_2 \subseteq \mathbf{LC}_3 \subseteq \cdots$, so giving a hierarchy for a class of cuppable degrees.

1. Introduction

We say that a set $A \subseteq \omega = \{0, 1, 2, \cdots\}$ is *computably enumerable* (c.e., for short), if there is a computable function to enumerate the elements of it. The c.e. sets are fundamental to the computability theory because they are very "near" to the computable functions, and because they form fine structures within the noncomputable universe. Turing (1939) introduced the relation of relative computability between sets. Given $A, B \subseteq \omega$, A is *computable in B* (or *Turing reducible to B*) if there is an algorithm to decide whether or not $x \in A$ for any $x \in \omega$, when given answers to all questions of the form "Is $y \in B$?". We write $A \leq_T B$ to indicate that A is computable in B and $A \equiv_T B$ if $A \leq_T B$ and $B \leq_T A$. The equivalence class of A under $\equiv_T$ is the (Turing) *degree* of A and is written as $\deg_T(A) = \mathbf{a}$. A degree is called *computably enumerable* (c.e.), if it contains a c.e. set. The early undecidable problems from a wide range of mathematics and computer science were united by the fact that they all have the same degree $\mathbf{0}'$. Post (1944) noted that $\mathbf{0}'$ is exactly the greatest computably enumerable degree, and asked whether there is a c.e. degree other than $\mathbf{0}$ (the least c.e. degree) and $\mathbf{0}'$.

Friedberg [1957], and independently Muchnik [1956], answered Post's question by constructing two incomparable c.e. degrees. Improving this, Sacks [1963] showed that every nonzero c.e. degree $\mathbf{a}$ can be written as $\mathbf{a}_0 \vee \mathbf{a}_1$ for some incomparable c.e. degrees $\mathbf{a}_0$, $\mathbf{a}_1$, where $\mathbf{a}_0 \vee \mathbf{a}_1$ is the least upper bound of $\mathbf{a}_0$ and $\mathbf{a}_1$, and Sacks [1964] proved that the computably enumerable degrees are dense. Shoenfield [1965] then conjectured that for any finite partial orderings $P \subseteq Q$, with the least element 0 and the greatest element 1, any embedding of P into $\mathcal{E}$ (the set of all c.e. degrees) can be extended to an embedding of Q into the same $\mathcal{E}$.

Received March 16, 1999.
1991 Mathematics Subject Classification. Primary 03D25, 03D30; Secondary 03D35.
The first-named author was partially supported by an EPSRC Research Grant, "Turing Definability", No. GR/M 91419 (UK), and by NSF Grant No. 69973048 and No. 19931020 (P.R. China).

Two consequences of the conjecture which were listed by Shoenfield himself are:

C1. There are no incomparable c.e. degrees $\mathbf{a}, \mathbf{b}$ such that $\mathbf{a} \wedge \mathbf{b}$ (the greatest lower bound of $\mathbf{a}, \mathbf{b}$) exists;

C2. For any c.e. degrees $\mathbf{0} < \mathbf{c} < \mathbf{a}$, there is a c.e. degree $\mathbf{b} < \mathbf{a}$ such that $\mathbf{b} \vee \mathbf{c} = \mathbf{a}$.

C1 is refuted by the minimal pair theorem of Lachlan [1966], and independently of Yates [1966], which states that there are c.e. degrees $\mathbf{a}_0, \mathbf{a}_1 \neq \mathbf{0}$ such that $\mathbf{a}_0 \wedge \mathbf{a}_1 = \mathbf{0}$. The pair $(\mathbf{a}_0, \mathbf{a}_1)$ is called a *minimal pair*, and the method used in the proof is called the *minimal pair method.*

We say that a degree $\mathbf{a} \in \mathcal{E}$ has the *anticupping* (a.c.) *property* if there is a c.e. degree $\mathbf{b}$ such that $\mathbf{0} < \mathbf{b} < \mathbf{a}$ and that for no c.e. degree $\mathbf{c} < \mathbf{a}$ does $\mathbf{a} = \mathbf{b} \vee \mathbf{c}$. C2 asserts that no $\mathbf{a} \in \mathcal{E}$ has the a.c. property. However, Yates, Cooper [1974a] and Harrington [1976] showed that $\mathbf{0}'$ has the a.c. property. A c.e. degree $\mathbf{a}$ is called *cuppable*, if there is a c.e. degree $\mathbf{b} \neq \mathbf{0}'$ such that $\mathbf{a} \vee \mathbf{b} = \mathbf{0}'$, and *noncuppable*, otherwise. At the opposite extreme of the a.c. property, Harrington showed the *plus-cupping* property: there exists a c.e. degree $\mathbf{a} \neq \mathbf{0}$ for which every nonzero c.e. degree $\mathbf{x} \leq \mathbf{a}$ cups to every c.e. degree $\mathbf{y} \geq \mathbf{a}$, so that not every degree has the a.c. property. (For an interesting weak version of this, see Fejer and Soare [1981].)

Using the Turing jump, we can define a hierarchy for a subset of the c.e. degrees. For $n > 0$, define a c.e. degree $\mathbf{a}$ to be low$_n$ (high$_n$) if $\mathbf{a}^{(n)} = \mathbf{0}^{(n)} (\mathbf{a}^{(n)} = \mathbf{0}^{(n+1)})$, where $\mathbf{x}^{(n+1)} = (\mathbf{x}^{(n)})'$, $\mathbf{x}^{(0)} = \mathbf{x}$. Let $\mathbf{L}_n$ and $\mathbf{H}_n$ be the set of all low$_n$ and high$_n$ c.e. degrees. For $n = 1$, an element of $\mathbf{L}_1$ is also called *low*, and an element of $\mathbf{H}_1$ is also called *high.* Now we have a *high/low hierarchy* that $\mathbf{L}_n \subset \mathbf{L}_{n+1}$ and that $\mathbf{H}_n \supset \mathbf{H}_{n+1}$ for all n. The high/low hierarchy is closely related to the structure of the c.e. degrees. For instance, the Robinson splitting theorem [1971] asserts that there is no low Lachlan nonsplitting base, the Cooper minimal pair theorem [1974b] states that every high c.e. degree bounds a minimal pair.

We say that a c.e. degree $\mathbf{a}$ is *cappable*, if there is a c.e. degree $\mathbf{b} \neq \mathbf{0}$ such that $\mathbf{a} \wedge \mathbf{b} = \mathbf{0}$, and *noncappable*, otherwise. Let $\mathbf{M}$ and $\mathbf{NC}$ be the set of all cappable and all noncappable degrees, respectively. Ambos-Spies, Jockusch, Shore and Soare [1984] showed that $\mathbf{M}$ is an ideal, that $\mathbf{NC}$ is a filter, and that $\mathbf{NC} = \mathbf{PS} = \mathbf{LC}$, where $\mathbf{PS}$ is the set of all degrees of promptly simple sets, $\mathbf{LC}$ is the set of all low-cuppable c.e. degrees, the degrees which join to $\mathbf{0}'$ with a low c.e. degree.

Extending the notion of low-cuppability, we say that a c.e. degree $\mathbf{a}$ is low$_n$-*cuppable*, if there is a low$_n$ c.e. degree $\mathbf{b}$ such that $\mathbf{a} \vee \mathbf{b} = \mathbf{0}'$. Let $\mathbf{LC}_n$ be the set of all low$_n$-cuppable c.e. degrees. By the lowness hierarchy, we have $\mathbf{LC}_1 \subseteq \mathbf{LC}_2 \subseteq \mathbf{LC}_3 \subseteq \cdots$. In the present paper, we show the following:

THEOREM. $\mathbf{LC}_1 \subset \mathbf{LC}_2$.

The result provides a base for a hierarchy for the cuppable degrees, and it also extends the *Harrington cup and cap theorem* [1978], which asserts that there is some c.e. degree both cuppable and cappable.

The paper is organized as follows. In Section 2, we describe the requirements and describe the strategies to satisfy the requirements. In Section 3, we describe an effective construction to build the desired objects. In Section 4, we verify that the construction satisfies all of the requirements.

Our notation and terminology are standard and generally follow Soare [1987]. During the course of a construction, notations such as A, Φ are used to denote the current approximations to these objects, and if we want to specify the values of objects, A, Φ say, which are the approximation immediately at the end of stage s, then we denote them by A_s, $\Phi[s]$, etc.. For a *partial computable* (p.c.) functional, Φ say, the *use function* is denoted by the corresponding lower case letter ϕ. The value of the use function of a converging computation is the greatest number which is actually used in the computation. For a p.c. functional which is not built by us, if a computation is not defined, then we define its use function to be -1. For a p.c. functional, Γ say, which is built by us, if $\Gamma(x)$ is not defined, then we regard the use function $\gamma(x)$ as ω. During the course of a construction, whenever we define a parameter, p say, as *fresh*, we mean that p is defined as the least natural number which is greater than any number mentioned so far, and in particular, if p is defined as fresh at stage s, then $p > s$.

2. Requirements and strategies

To prove the theorem, we construct c.e. sets A, B, L, and partial computable functionals Γ, Ω, to satisfy the following requirements:

G: $K = \Gamma(A, L)$;

P_e: $B \neq \xi_e$;

R_e: $\Phi_e(A) = \Phi_e(B) = h$ total $\rightarrow h$ is computable;

L_e: $\mathrm{Tot}^{\mathrm{L}} = \Omega(\emptyset''; e)$

where $e \in \omega$, $\mathrm{Tot}^{\mathrm{L}} = \{e\colon \Phi_e(L) \text{ is total}\}$, $\{(\xi_e, \Phi_e)\colon e \in \omega\}$ is an effective enumeration of all pairs (ξ, Φ) such that ξ is a partial computable function, Φ is a partial computable functional, and K is a fixed creative set.

Let $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{l}$ be the Turing degrees of A, B and L respectively. By the G-requirement and the L-requirements, $\mathbf{a} \in \mathbf{LC}_2$. By the P- and the R-requirements, $\mathbf{a}$ is cappable, so $\mathbf{a} \notin \mathbf{LC}_1$. Therefore meeting the requirements is sufficient to prove the theorem.

A strategy is an effective procedure designed to satisfy certain requirement. All strategies are arranged on a priority tree, and a strategy is identical to a node of the priority tree.

The G-strategy The G-strategy will proceed as follows:

1. If there is an x such that $\Gamma(A, L; x) \downarrow \neq K(x)$, then let k be the least such x, enumerate $\gamma(k)$ into L, and let $\Gamma(A, L; x)$ be undefined for all $x \geq k$.

2. If k is the least number x such that $\Gamma(A, L; x) \uparrow$, then define $\Gamma(A, L; x) = K(k)$ with $\gamma(k)$ fresh.

The G-strategy runs the basic actions as above. We note that the G-strategy never enumerates any element into A, so the G-strategy does not injure any R-strategy. However, the G-strategy is not sufficient to satisfy the requirement G. Otherwise, we always code the markers γ into L, so that $K \leq_T L$, contradicting the L-requirements which are ensuring the low$_2$-ness of L. Therefore the G-strategy above threatens some of the L-requirements. This will be resolved gradually (see below).

Returning to the building of Γ, we will ensure that the use function γ of Γ will have the following basic properties:

(1) For any k, s, if $\Gamma(A, L; k)[s] \downarrow$, then $\gamma(k)[s] \notin A_s \cup L_s$;
(2) For any x, y, if $x < y$, and $\gamma(y) \downarrow$, then $\gamma(x) \downarrow$ and $\gamma(x) < \gamma(y)$;
(3) Whenever we define $\gamma(k)$, we define it as fresh;
(4) $\Gamma(A, L; x)$ is injured at stage s iff at stage s, there is an $y \leq x$ such that $\gamma(y)$ is enumerated into A or L;
(5) If $\Gamma(A, L; k)[s] \downarrow$ and $k \in K_{s+1} - K_s$, then there is an $n \leq k$ such that $\gamma(n)[s] \in A - A_s$ or $\gamma(n)[s] \in L - L_s$.

We say that (1)–(5) above are *γ-rules*. The γ-rules ensure that if $\Gamma(A, L)$ is total, then $\Gamma(A, L) = K$. G is satisfied.

Thus the point is we need to guarantee the totality of $\Gamma(A, L)$.

An L-strategy To satisfy an L- requirement, we will implicitly build a partial computable functional Ω to ensure that $\mathrm{Tot}^L = \Omega(\emptyset'')$.

Given an L-requirement, L_e say, we define the "*length function*" for L_e as follows:

$$l(e) = \max\{x \mid (\forall y < x)[\Phi_e(L; y) \downarrow]\}.$$

We say that a stage s is *L_e-expansionary* if $s = 0$ or $l(e)[s] > l(e)[t]$ for all $t < s$ and $l(e)[s] > e + 1$. If there are only finitely many L_e-expansionary stages, then $\Phi_e(L)$ is not total. This indicates to us that the L-requirements could be satisfied as follows.

Suppose the next property holds:

$(*)$ If there are infinitely many L_e-expansionary stages, then $\Phi_e(L)$ is total.

Since our construction will be arranged as a normal $\mathbf{0}''$-priority argument (every node has only finitely many branches), the *true path TP* will be computable in $\emptyset''$.

By property $(*)$, we can use TP (as oracle) to read out the totality of $\Phi_e(L)$ as follows:

Given L_e, find the unique L_e-strategy, α say, such that $\alpha \in TP$. Then if $\alpha^\frown\langle 0\rangle \in TP$, then $\Phi_e(L)$ is total; if $\alpha^\frown\langle 1\rangle \in TP$, then $\Phi_e(L)$ is not total, where 0, 1 denote infinite and finite actions, respectively.

Thus we have that $\mathrm{Tot}^{\mathrm{L}} \leq_{\mathrm{T}} TP \leq_{\mathrm{T}} \emptyset''$. The L-requirements are satisfied.

Now the key to the L-strategy is to ensure property $(*)$. That is, if there are infinitely many L_e-expansionary stages, then we have to ensure that $\Phi_e(L)$ is total.

An L-strategy, L_e say, will be injured by the G-strategy. This conflict is resolved by enumerating certain γ-uses into A to lift the γ-markers. However, if there are infinitely many L-expansionary stages, an L-strategy may enumerate the γ-uses infinitely often, this may make Γ partial.

Our solution is to introduce infinitely many substrategies, $S_{e,i}$ say, for an L-strategy, L_e (if there are infinitely many L_e-expansionary stages) say. Now an $S_{e,i}$-strategy will ensure that $\Phi_e(L)\lceil(i+1)$ converge eventually and permanently. In this case, infinitely many $S_{e,i}$ ensure the totality of $\Phi_e(L)$. L_e is satisfied. Therefore for an L-requirement, L_e say, we will introduce subrequirements $S_{e,i}$ for all $i \geq e$.

An $S_{e,i}$-strategy An $S_{e,i}$-strategy will preserve the computations $\Phi_e(L)\lceil(i+1)$. We note that the only possibility of the enumeration of L is the γ-uses which is enumerated by the G-strategy.

To preserve the computations $\Phi_e(L)\lceil(i+1)$, an $S_{e,i}$-strategy will work with a fixed *threshold*, k say. Whenever we define the threshold k, we define it as fresh. If $K\lceil k$ changes, then we cancel all the previous actions for the $S_{e,i}$-strategy, in which case we say that the $S_{e,i}$-strategy is *reset*. The point is that an $S_{e,i}$-strategy will be reset only finitely many times.

An $S_{e,i}$-strategy will proceed as follows:

1. Wait for a stage at which $\Phi_e(L)\lceil(i+1)$ are all defined;
2. Enumerate $\gamma(k)$ into A, set $r = \max\{\phi_e(x)\colon\ x \leq i\}$ and stop.

Suppose that the $S_{e,i}$-strategy will never be reset at any stage $> s_0$. Since there are infinitely many L_e-expansionary stages, we will reach step 1 at a stage $> s_0$, s_1 say; then by the actions in step 2, any γ-uses which will be enumerated into L at a stage $> s_1$ will be greater than s_1, since all new uses are chosen fresh. Therefore, $\Phi_e(L)[s_1]\lceil(i+1)$ will be preserved forever (We assume that the use function for a given p.c. functional is bounded by stages.) $S_{e,i}$ is satisfied. The parameter r is just to indicate that the strategy has passed Step 2 which will be useful in the description of the construction and the verfication. And we note that an $S_{e,i}$-strategy acts only finitely many times.

A P-strategy A P-strategy will satisfy a P-requirement, $B \neq \xi$ say. It is a Friedberg-Muchnik procedure and proceeds as follows:

1. Appoint a witness, b say, which is fresh;
2. Wait for a stage at which $\xi(b) \downarrow= 0 = B(b)$, then enumerate b into B, and stop.

The P-strategy above satisfies the P-requirement by one of the following cases:

Case 1. d: $\xi(b) \downarrow= 0 \neq 1 = B(b)$ for some b.
Case 2. w: $\xi(b) \downarrow\neq 0 = B(b)$ or $\xi(b) \uparrow$ for some b.

Therefore, the P-strategy has two possible outcomes $d,\ w$ with priority ordering $d <_{\mathrm{L}} w$.

An R-strategy An R-strategy, α say, attempts to satisfy an R-requirement, R_e say. It is a standard minimal pair strategy. That is, α tries to preserve computations up to the last length of agreement between $\Phi_e(A)$ and $\Phi_e(B)$, unless it has gotten a new length of agreement longer than any seen before. In the latter case, α imposes no restraint.

To be more precise, define the length function of agreement as usual:

$$l^R(e)[s] = \max\{x : (\forall y < x)(\Phi_e(A; y)[s] \downarrow= \Phi_e(B; y)[s] \downarrow)\};$$

s is *R_e-expansionary* if, for all $t < s$, $l^R(e)[s] > l^R(e)[t]$.

There are two possible outcomes for an R-strategy, 0, 1, with $0 <_{\mathrm{L}} 1$, where 0 indicates that there are infinitely many R_e-expansionary stages, and 1 indicates the finite case.

Obviously, if there are only finitely many R_e-expansionary stages, then $l^R(e)[s]$ will be bounded during the full construction, and so $\Phi_e(A) \neq \Phi_e(B)$, R_e is satisfied.

Note that only S-strategies enumerate elements into A, and only P-strategies enumerate elements into B, and at most one of $A,\ B$ gets one element at a stage. In addition, each such a strategy enumerates only finitely many elements into the corresponding set.

We now look at the satisfaction of an R-requirement, R say (we drop the index). Let α be the R-strategy (which will be on the true path TP). First, by the choice of α, α is initialized by a strategy $\xi <_{\mathrm{L}} \alpha$ only finitely many times; second, every P- or S-strategy $\beta \subset \alpha$ will act only finitely many times. Therefore, α is initialized only finitely many times. Let s_0 be the greatest stage at which α is initialized. By the choice of s_0, α will never be injured by any strategy $< \alpha$ at any stage $> s_0$.

Suppose that there are infinitely many α-expansionary stages, i.e., $\alpha\frown\langle 0\rangle$ will be on the true path TP. We compute $\Phi_\alpha(A) = \Phi_\alpha(B) = h_\alpha$ as follows.

Given x, find the least α-expansionary stage, s_1+1 say, such that $l^R(e)[s_1+1] > x$. We conclude that $h_\alpha(x) = \Phi_\alpha(A; x)[s_1] = \Phi_\alpha(B; x)[s_1]$. Suppose that $v_1 = s_1+1 <$

$v_2 < v_3 < \cdots$ are all α-expansionary stages. By the strategy, there is at most one of A, B received an element, but not both, during stage v_1. Therefore, we have the following:

(1) One of $(1)_a$ and $(1)_b$ below holds.
$(1)_a$ $A_{s_1}\lceil v_1 = A_{v_1}\lceil v_1$;
$(1)_b$ $B_{s_1}\lceil v_1 = B_{v_1}\lceil v_1$.

By the construction, all nodes ξ with $\alpha^\frown\langle 0\rangle <_L \xi$ are initialized; therefore, we have:

(2) Any number which is enumerated into $A \cup B$ during stages $s \in (v_1, v_2)$ will be greater than v_1.

By (1), $h_\alpha(x)[s_1] = \Phi_\alpha(A; x)[v_1]$ or $h_\alpha(x)[s_1] = \Phi_\alpha(B; x)[v_1]$, so by (2), we have $h_\alpha(x)[s_1] = h_\alpha(x)[v_2 - 1]$.

Suppose by the induction that $h_\alpha(x)[s_1] = h_\alpha(x)[s_n]$ for $s_n = v_n - 1$. Since there is at most one of A, B received elements during stage v_n, we have:

$(1)'$ One of $(1)'_a$ and $(1)'_b$ below holds.
$(1)'_a$ $A_{s_n}\lceil v_n = A_{v_n}\lceil v_n$;
$(1)'_b$ $B_{s_n}\lceil v_n = B_{v_n}\lceil v_n$.

By the initialization, all ξ with $\alpha^\frown\langle 0\rangle <_L \xi$ are initialized at the end of stage v_n. Therefore, we have:

$(2)'$ Any number which is enumerated into $A \cup B$ at a stage $s \in (v_n, v_{n+1})$ will be greater than v_n.

By $(1)'$, $h_\alpha(x)[s_n] = \Phi_\alpha(A; x)[v_n]$ or $h_\alpha(x)[s_n] = \Phi_\alpha(B; x)[v_n]$, so by $(2)'$ and by the choice of v_{n+1}, we have $h_\alpha(x)[s_1] = h_\alpha(x)[s_n] = h_\alpha(x)[v_{n+1} - 1]$.

Now, for all n, $h_\alpha(x)[s_1] = h_\alpha(x)[s_n]$, where $s_n = v_n - 1$. Thus, if $\Phi_\alpha(A; x) = \Phi_\alpha(B; x) = h_\alpha(x)$, then $h_\alpha(x)[s_1] = \Phi_\alpha(A; x) = \Phi_\alpha(B; x)$. We note that s_1 is effectively decided. If $\Phi_\alpha(A) = \Phi_\alpha(B) = h$ is total, then $\Phi_\alpha(A)$ is computable, R is satisfied.

3. The construction

Before describing the construction, we define the priority tree, T say, effectively.

Definition 1. (i) We define *the priority ranking of the requirements* as follows:

$$G < P_0 < R_0 < L_0 < S_{0,0} < P_1 < R_1 < L_1 < S_{0,1} < S_{1,1}$$
$$< \cdots < P_n < R_n < L_n < S_{0,n} < \cdots < S_{n,n} < P_{n+1} < \cdots,$$

where for any requirements, X, Y say, if $X < Y$, then X has higher priority than Y;

(ii) A P-strategy has two possible outcomes, d, w, with $d <_{\mathrm{L}} w$;

(iii) An R-strategy has two possible outcomes, 0, 1, with $0 <_{\mathrm{L}} 1$, to denote infinite and finite actions, respectively;

(iv) An L-strategy has two possible outcomes, 0, 1, with $0 <_{\mathrm{L}} 1$, to indicate infinite and finite actions respectively;

(v) An S-strategy has only one possible outcome, denoted by 1.

Definition 2. Let $\xi \in T$.

(i) We say that P_e *is satisfied at* ξ, if there is a P_e-strategy α such that $\alpha \subset \xi$.

(ii) We say that R_e *is satisfied at* ξ, if there is an R_e-strategy α such that $\alpha \subset \xi$.

(iii) We say that L_e *is satisfied at* ξ, if there is an L_e-strategy α such that $\alpha^\frown\langle 1\rangle \subseteq \xi$. We say that L_e is *active at* ξ, if there is an L_e-strategy α such that $\alpha \subset \alpha^\frown\langle 0\rangle \subseteq \xi$, in which case L_e is said to be *active at* ξ via α.

(iv) We say that $S_{e,i}$ *is satisfied at* ξ, if either L_e is satisfied at ξ, or L_e is active at ξ via α, and there is an $S_{e,i}$-strategy β such that $\alpha^\frown\langle 0\rangle \subseteq \beta \subset \xi$.

Now we construct the priority tree T as follows:

Definition 3. (i) We define the root node, $\lambda = \langle\ \rangle$ say, as a P_0-strategy.

(ii) The immediate successors of a node are the possible outcomes of the corresponding strategy.

(iii) For $\xi \in T$, ξ works for the highest priority requirement which has not been satisfied, and not been active at ξ.

Continuing the inductive steps above, we have built our priority tree T.

Definition 4. Let $\xi \in T$.

(i) If $\xi = \alpha$ is an R_e-strategy, then s is *α-expansionary* if $s = 0$ or $l(e)[s] > l(e)[v]$ for all $v < s$ at which α is visited.

(ii) If $\xi = \tau$ is an L_e-strategy, we say that s is *τ-expansionary* if $s = 0$ or $l(e)[s] > l(e)[v]$ for all $v < s$ at which τ is visited, and $l(e)[s] > e + 1$.

A P-strategy α has one parameter, $b(\alpha)$. An $S_{e,i}$-strategy β has two parameters, $k(\beta)$, $r(\beta)$. L-strategies and R-strategies have no parameter attached. If a strategy is initialized, then any parameter of it will be cancelled. If $\xi < \delta$, and ξ is initialized, then δ is initialized simultaneously and automatically. If an S-strategy γ is reset, then set $r(\gamma)$ to be undefined (if it is defined).

The construction Without loss of generality, we suppose that K is enumerated at odd stages, $2n + 1$, $n \in \omega$, and that only one element is enumerated into K at each such a stage.

Stage 0. Set $A = B = L = \emptyset$, and initialize all nodes.

Stage $s + 1 = 2n + 1$. Let k be such that $k \in K_{s+1} - K_s$.

1. If $\Gamma(A, L; k) \downarrow$, then:
   Enumerate $\gamma(k)$ into L;
   For any strategy α, if the threshold $k(\alpha)$ of α is defined such that $k < k(\alpha)$, then α is reset;
   Go to stage $s + 2$.
2. Otherwise, let x be the least y such that $\Gamma(A, L; y) \uparrow$, define $\Gamma(A, L; x) = K(x)$ with $\gamma(x)$ fresh, and go to stage $s + 2$.

Stage $s + 1 = 2(n + 1)$. A strategy ξ is *visited at* stage $s + 1$, if ξ is eligible to act at a substage t of stage $s + 1$. First, we allow the root node λ to be eligible to act at substage $t = 0$.

Substage t. Let ξ be eligible to act at substage t. If $t = s + 1$, then initialize any γ with $\xi <_L \gamma$, and go to stage $s + 2$. Otherwise, there are four cases:

Case 1. $\xi = \alpha$ is a P-strategy.
Run program α below.

$\alpha1$. If $b(\alpha) \downarrow$ and $b(\alpha) \in B$, then let $\alpha^\frown\langle d\rangle$ be eligible to act at the next substage;
$\alpha2$. If $b(\alpha) \downarrow$, $\xi(b(\alpha)) \downarrow= 0 = B(b(\alpha))$, then enumerate $b(\alpha)$ into B, initialize all $\beta \not\leq \alpha$ and go to stage $s + 2$;
$\alpha3$. If $b(\alpha) \uparrow$, then define $b(\alpha)$ as fresh, initialize all $\xi \not\leq \alpha$ and go to stage $s + 2$;
$\alpha4$. Otherwise, let $\alpha^\frown\langle w\rangle$ be eligible to act at the next substage.

Case 2. $\xi = \beta$ is an R_e-strategy.

Run program β:

$\beta1$. If $s + 1$ is a β-expansionary stage, then let $\beta^\frown\langle 0\rangle$ be eligible to act at the next substage.
$\beta2$. Otherwise, let $\beta^\frown\langle 1\rangle$ be eligible to act at the next substage.

Case 3. $\xi = \eta$ is an L_e-strategy.
Run program η as in Case 2.

Case 4. $\xi = \zeta$ is an $S_{e,i}$-strategy, for some e, i.
Run program ζ:

$\zeta1$. If $r(\zeta) \downarrow$, then let $\zeta^\frown\langle 1\rangle$ be eligible to act at the next substage;
$\zeta2$. If $r(\zeta) \uparrow$, $k(\beta) \downarrow$, and $\Phi_e(L)\lceil(i + 1)$ are all defined, then enumerate $\gamma(k(\zeta))$ into A, let $r(\zeta) = \max\{\phi_e(x) | x \leq i\}$, initialize all $\delta \not\leq \zeta$ and go to stage $s + 2$;

$\zeta 3$. If $r(\zeta)\uparrow$, $k(\beta)\downarrow$, and there is an $x \le i$ such that $\Phi_e(L;x)\uparrow$, then initialize all $\delta \not\le \zeta$ and go to stage $s+2$;

$\zeta 4$. If $r(\zeta)\uparrow$, $k(\beta)\uparrow$, then define $k(\zeta)$ as fresh, initialize all $\delta \not\le \gamma$ and go to stage $s+2$.

This completes the description of the construction.

4. The verification

We now verify that the construction satisfies the requirements.

LEMMA 1. *For any x, s:*

(i) *If $\Gamma(A, L; x)[s]\downarrow$, then $\gamma(x)[s] \notin (A_s \cup L_s)$.*
(ii) *If $\Gamma(A, L; x+1)[s]\downarrow$, then $\Gamma(A, L; x)[s]\downarrow$ and $\gamma(x)[s] < \gamma(x+1)[s]$.*
(iii) *If $\Gamma(A, L; x)$ changes to be undefined at stage $s+1$, then $A_{s+1}\lceil(\gamma(x)[s]+1) \neq A_s\lceil(\gamma(x)[s]+1)$ or $L_{s+1}\lceil(\gamma(x)[s]+1) \neq L_s\lceil(\gamma(x)[s]+1)$.*
(iv) *If $s < v$, $\Gamma(A, L; x)[s]\downarrow$, $A_v\lceil(\gamma(x)[s]+1) = A_s\lceil(\gamma(x)[s]+1)$ and $L_v\lceil(\gamma(x)[s]+1) = L_s\lceil(\gamma(x)[s]+1)$, then $\Gamma(A, L; x)[v]\downarrow$, and $\gamma(x)[v] = \gamma(x)[s]$.*

Proof. This is immediate from the construction.

For an even stage s, let δ_s be the strategy which is visited at the last substage of stage s. Define *the true path of the construction* $TP = \liminf_{s\in\{2n:\, n\in\omega\}} \delta_s$.

LEMMA 2. *For all $\alpha \in TP$:*

(i) *There is an a such that $\alpha^\frown\langle a\rangle \in TP$.*
(ii) *$\alpha^\frown\langle a\rangle \in TP$ is visited infinitely often.*
(iii) *$\alpha^\frown\langle a\rangle \in TP$ is initialized or reset only finitely many times.*
(iv) *If α is a P- or an S-strategy, then α acts only finitely many times.*

Proof. We prove it inductively. For the root node $\lambda \in TP$, λ will be initialized only at stage $s = 0$. By the construction, λ is visited at every even stage > 0. Therefore $\lim_s b(\lambda)[s] = b(\lambda) < \omega$. By Case 1 of the construction, if $b(\lambda) \in B$, then $\lambda^\frown\langle d\rangle \in TP$, otherwise, $\lambda^\frown\langle w\rangle \in TP$. Let s_1 be the stage at which $\lim_s b(\lambda)[s] = b(\lambda)$ is defined. If $\lambda^\frown\langle w\rangle \in TP$, then for every even $s > s_1$, $\lambda^\frown\langle w\rangle \in TP$ is visited at stage s, and $\lambda^\frown\langle w\rangle$ will never be initialized at any stage $> s_1$, so $\lambda^\frown\langle w\rangle$ will be initialized or reset only finitely many times, and λ will never act at any stage $> s_1$. If $\lambda^\frown\langle d\rangle \in TP$, then let s_2 be the stage at which $b(\lambda)$ is enumerated into B. Then $\lambda^\frown\langle d\rangle$ will never be initialized at any stage $> s_2$, so $\lambda^\frown\langle d\rangle$ will be initialized or reset only finitely many times. And for any even $s > s_2$, $\lambda^\frown\langle d\rangle$ is visited at stage s, and λ will never act at any stage $> s_2$. The lemma holds for $\alpha = \lambda$.

Suppose by induction that the lemma holds for all $\alpha' \subset \alpha$, and that $\alpha \in TP$. We have:

1. α is visited infinitely often.
2. There is a stage s_0 such that no $\beta \subseteq \alpha$ is reset or initialized after stage s_0 and such that for all $\delta \subset \alpha$, if δ is a P- or an S-strategy, then δ does not act after stage s_0.

Now we have four cases.

Case 1. α is a P-strategy. By the choice of s_0, and by the hypotheses, $\lim_s b(\alpha)[s] = b(\alpha) < \omega$ exists. Let s_1 be the stage at which $b(\alpha)$ is defined. By Case 1 of the construction, if $b(\alpha) \in B$, then $\alpha^\frown\langle d\rangle \in TP$; otherwise, $\alpha^\frown\langle w\rangle \in TP$. If $b(\alpha) \in B$, then let s_2 be the stage at which $b(\alpha)$ is enumerated into B; otherwise, let $s_2 = s_1$. Then if $\alpha^\frown\langle a\rangle \in TP$, then $\alpha^\frown\langle a\rangle \in TP$ will be initialized or reset only finitely many times. By the choice of s_2, if $s > s_2$, $\alpha^\frown\langle a\rangle \in TP$, and α is visited at stage s, then $\alpha^\frown\langle a\rangle$ is visited at stage s. Again by the choice of s_2, α will never act after stage s_2. The lemma holds in Case 1.

Case 2. α is an R-strategy. By Case 2 of the construction and by the induction hypotheses, if there are infinitely many α-expansionary stages, then $\alpha^\frown\langle 0\rangle \in TP$, otherwise $\alpha^\frown\langle 1\rangle \in TP$. If $\alpha^\frown\langle 0\rangle \in TP$, then by the choice of s_0, $\alpha^\frown\langle 0\rangle$ will never be initialized at any stage $> s_0$, so $\alpha^\frown\langle 0\rangle$ will be initialized or reset only finitely many times, and $\alpha^\frown\langle 0\rangle$ will be visited at every α-expansionary stage $> s_0$. If $\alpha^\frown\langle 1\rangle \in TP$, let s_2 be minimal after which there is no α-expansionary stage. By the choice of s_2, $\alpha^\frown\langle 1\rangle$ will not be initialized at any stage $> s_2$, so $\alpha^\frown\langle 1\rangle$ will be initialized or reset only finitely many times, and $\alpha^\frown\langle 1\rangle$ will be visited at every stage $> s_2$ at which α is visited. The lemma holds in Case 2.

Case 3. α is an L-strategy. Similar to that of Case 2.

Case 4. $\alpha = \delta$ is an $S_{e,i}$-strategy. By the choice of s_0, $\lim_s k(\delta)[s] = k(\delta)$ exists and it will be defined at a stage, $s_1(> s_0)$ say. Let s_2 be minimal $> s_1$ such that $K_{s_2}\lceil k(\delta) = K\lceil k(\delta)$. By the choice of s_2, $r(\delta)[s_2]$ is undefined, and $r(\delta)$ will not be cancelled at any stage $> s_2$ (if it is defined).

Let τ be the L_e-strategy $\subset \delta$.

We have two subcases.

Subcase 4a. $i = e$. By the definition of the τ-expansionary stage, let s_3 be the least τ- expansionary stage $> s_2$, at which δ is visited, then by the definition of τ-expansionary, $\Phi_e(L)\lceil(e+1)$ are all defined. If $r(\delta)$ is currently undefined, then δ enumerates $\gamma(k(\delta))$ into A at stage s_3. By the choice of s_0, s_1 and s_2, $\Phi_e(L)[s_3]\lceil(e+1)$ will be preserved forever, and $r(\delta)[s_3]$ will never be injured at any stage $> s_3$. δ

will never act at any stage $> s_3$, and for any $s > s_3$, if δ is visited at stage s, then so is $\delta^\frown\langle 1\rangle$.

Subcase 4b. $e < i$. By the choice of s_2, $\Phi_e(L)\lceil i$ has been preserved by a strategy $\subset \delta$ at a stage $< s_2$, s_2^- say. Therefore, during any τ-expansionary stage $> s_2$, $\Phi_e(L)\lceil(i+1)$ are all defined. Thus at the least τ-expansionary stage $> s_2$, s_3 say, at which δ is visited, $\Phi_e(L)\lceil(i+1)$ are all defined, in which case δ preserves $\Phi_e(L)\lceil(i+1)$ forever (if $r(\delta)$ has not been defined yet). So the S-strategy δ will not act at any stage $> s_3$. By the construction, $\xi = \delta^\frown\langle 1\rangle$ will never be initialized at a stage $> s_3$, and ξ will be visited at any stage $> s_3$ at which δ is visited and δ will never act at any stage $> s_3$. The lemma holds in Case 4.

The lemma follows.

We now verify that the construction satisfies all requirements.

LEMMA 3. $TP \leq_T \emptyset''$.

Proof. By Lemma 2 (i), TP is a total function. By the definition of TP, the question of whether "$\alpha \in TP$" is a proposition of finitely many Σ_2^0- and Π_2^0-questions, by the definition of the ture path TP. So, $TP \leq_T \emptyset''$. The lemma follows.

LEMMA 4. *Let $\alpha \in TP$, where $\alpha^- = \beta$ is a P-strategy.*

(i) *If $\alpha = \beta^\frown\langle d\rangle$, then $\xi_\beta(b(\beta))\downarrow = 0 \neq 1 = B(b(\beta))$.*
(ii) *If $\alpha = \beta^\frown\langle w\rangle$, then $\xi_\beta(b(\beta))\downarrow \neq 0 = B(b(\beta))$, or $\xi_\beta(b(\beta))\uparrow$.*

Proof. This is immediate from the construction.

LEMMA 5. *Let $\alpha \in TP$, where $\alpha^- = \beta$ is an R_e-strategy.*

(i) *If $\alpha = \beta^\frown\langle 1\rangle$, then $\Phi_e(A) \neq \Phi_e(B)$.*
(ii) *If $\alpha = \beta^\frown\langle 0\rangle$, and $h = \Phi_e(A) = \Phi_e(B)$ is total, then h is computable.*

Proof. Fix e. If $\alpha = \beta^\frown\langle 1\rangle$, then there are only finitely many β-expansionary stages. Since $\beta \in TP$, β will be visited infinitely often, and so there are only finite many R_e-expansionary stages, $\Phi_e(A) \neq \Phi_e(B)$, or one of $\Phi_e(A)$ and $\Phi_e(B)$ is not total, (i) follows.

Assume that $\alpha = \beta^\frown\langle 0\rangle$ and $h = \Phi_e(A) = \Phi_e(B)$ is total. Since $\beta \in TP$ and every P- or S-strategy $\beta \subset \alpha$ will act only finitely many times, we can choose s_0 as the greatest stage at which α is initialized. That is, after stage s_0, β will never be injured by any strategy $< \alpha$. We compute $\Phi_e(A) = \Phi_e(B) = h$ as follows.

Fix x and find the least β-expansionary stage, s_1 say, such that $s_1 > s_0$ and $l^R(e)[s_1] > x$. Then $h(x) = \Phi_e(A; x)[s_1] = \Phi_e(B; x)[s_1]$, because by the argument given in the R_e-strategy, if $v > s_1$ is a β-expansionary stage, then only one side of computations can be injured at stage v, and the other side of computations are not injured at this stage and will be preserved till the next β-expansionary stage. Thus, if $\Phi_e(A; x) = \Phi_e(B; x) = h(x)$ is defined, then $h(x) = \Phi_e(A; x)[s_1] = \Phi_e(B; x)[s_1]$. Therefore, if $\Phi_e(A) = \Phi_e(B) = h$ is total, then h is computable, R_e is satisfied and (ii) follows.

LEMMA 6. *Given L_e- and $S_{e,i}$- strategies τ and α (respectively) with $\tau^\frown\langle 0\rangle \subseteq \alpha \in TP$, then α succeeds in preserving the computations $\Phi_e(L)\lceil(i+1)$.*

Proof. By the argument of Case 4 in Lemma 2.

LEMMA 7. $L'' \leq_{\mathrm{T}} \emptyset''$.

Proof. Fix e and let η be the unique L_e-strategy in TP. It is easy to see that if $\eta^\frown\langle 1\rangle \in TP$, then $\Phi_e(L)$ is not total. If $\eta^\frown\langle 0\rangle \in TP$, by the construction of the priority tree, for each $i \geq e$, there is an $S_{e,i}$-strategy γ in TP. By Lemma 6, computations $\Phi_e(L)\lceil(i+1)$ are preserved. Since i is chosen arbitrarily, $\Phi_e(L)$ is total. So $\mathrm{Tot}^L \leq_{\mathrm{T}} TP$. By lemma 3, $\mathrm{Tot}^L \leq_{\mathrm{T}} \emptyset''$. Hence, $L'' \leq_{\mathrm{T}} \emptyset''$.

LEMMA 8. $\Gamma(A, L) = K$.

Proof. By Lemma 1, $\Gamma(A, L)$ is a p.c. functional. By actions during odd stages, if $\Gamma(A, L)$ is total, then $\Gamma(A, L) = K$.

Suppose that $\Gamma(A, L)$ is not total. Let k be the least number such that $\Gamma(A, L; x)\uparrow$, then, $\{\gamma(k)[s] : s \in \omega\}$ will be unbounded during the full construction. By the construction, we know that there is an S-strategy ζ such that $\lim_s k(\zeta)[s] \downarrow= k(\zeta) = k$, and the actions of ζ make $\Gamma(A, L; k)$ undefined infinitely often. So, $\zeta \not<_{\mathrm{L}} TP$. Since $\lim_s k(\zeta)[s]\downarrow$, we have $TP \not<_{\mathrm{L}} \zeta$. So, $\zeta \in TP$. Let $r(\zeta) = \lim r(\zeta)[s]$. By Lemma 2, $r(\zeta)$ exists, and so $\{\gamma(k)[s] : s \in \omega\}$ cannot be unbounded, a contradiction. So, $\Gamma(A, L)$ is total, and $\Gamma(A, L) = K$.

This completes the proof of the theorem.

REFERENCES

[1] K. Ambos-Spies, C. G. Jocksch, Jr., R. A. Shore and R. I. Soare [1984], *An algebraic decomposition of the recursively enumerable degrees and the coincidence of several degree classes with the promptly simple degrees*, Trans. Amer. Math. Soc. **281**, 109–128.

[2] S. B. Cooper [1974a], *On a theorem of C. E. M. Yates* (handwritten notes).

[3] S. B. Cooper [1974b], *Minimal pairs and high recursively enumerable degrees*, J. Symbolic Logic, **39**, 655–660.

[4] P. A. Fejer and R. I. Soare [1981], "The plus–cupping theorem for the recursively enumerable degrees" in *Logic Year* 1979–80, Lerman, Schmerl, and Soare eds., Lecture Notes in Mathematics, no. 859, Springer-Verlag, New York, pp. 49–62.

[5] R. M. Friedberg [1957], *Two recursively enumerable sets of incomparable degrees of unsolvability*, Proc. Nat. Acad. Sci. U.S.A **43**, 236–238.

[6] L. Harrington [1976], *On Cooper's proof of a theorem of Yates, Part I* (handwritten notes).

[7] L. Harrington [1978], *Plus cupping in the recursively enumerable degrees* (handwritten notes).

[8] A. H. Lachlan [1966], *Lower bounds for pairs of recursively enumerable degrees*, Proc. London Math. Soc. **16**, 537–569.

[9] A. H. Lachlan [1979], *Bounding minimal pairs*, J. Symbolic Logic **44**, 626–642.

[10] A. Li [ta], *Bounding cappable degrees*, Arch. Math. Logic, to appear.

[11] A. A. Muchnik [1956], *On the unsolvability of the problem of reducibility in the theory of algorithms*, Dokl. Akad. Nauk SSSR N.S. **108**, 194–197 (Russian).

[12] R. W. Robinson [1971], *Interpolation and embedding in the recursively enumerable degrees*, Ann. of Math. **93** (1971), 285–314.

[13] G. E. Sacks [1963], *On the degree less than* $\mathbf{0}'$, Ann. of Math. **77**, 211–231.

[14] G. E. Sacks [1964], *The recursively enumerable degrees are dense*, Ann. of Math. (2) **80**, 300–312.

[15] J. R. Shoenfield [1965], "Applications of model theory to degrees of unsolvability" in *Symposium on the Theory of Models*, Addison, Henkin, and Tarski eds., North-Holland, Amsterdam, pp. 359–363.

[16] R. I. Soare [1987], *Recursively enumerable sets and degrees*, Perspectives in Mathematical Logic, Springer–Verlag, New York.

[17] C. E. M. Yates [1966], *A minimal pair of recursively enumerable degrees*, J. Symbolic Logic **31**, 159–168.

Angsheng Li, Institute of Software, Chinese Academy of Sciences, P.O.Box 8718, Beijing 100080, People's Republic of China
liang@ox.ios.ac.cn
Current address: School of Mathematics, University of Leeds, Leeds LS 2 9JT, United Kingdom
angsheng@amsta.leeds.ac.uk

Guohua Wu, Institute of Software, Chinese Academy of Sciences, P.O.Box 8718, Beijing 100080, People's Republic of China
Current address: Department of Mathematics, Victoria University, Private Bag, Wellington, New Zealand
wu@mcs.vuw.ac.nz

Zaiyue Zhang, Department of Mathematics, Yangzhou University, Yangzhou, Jiangsu 225002, People's Republic of China