A JOIN THEOREM FOR THE COMPUTABLY ENUMERABLE DEGREES

CARL G. JOCKUSCH, JR., ANGSHENG LI, AND YUE YANG

Abstract. It is shown that for any computably enumerable (c.e.) degree \( w \), if \( w \neq 0 \), then there is a c.e. degree \( a \) such that \((a \vee w)' = a'' = 0''\) (so \( a \) is low\(_2\) and \( a \vee w \) is high). It follows from this and previous work of P. Cholak, M. Groszek and T. Slaman that the low and low\(_2\) c.e. degrees are not elementarily equivalent as partial orderings.

1. Introduction

This paper concerns properties of the join operation on the computably enumerable degrees. Part of our motivation comes from the study of the join operation on the Turing degrees by D. Posner and R. Robinson. In [9], Theorem 1, they showed that in the Turing degrees, for any nonzero degree \( w \leq 0' \), there exists a degree \( a \) such that \( a \vee w = a' = 0' \), where \( a' \) denotes the Turing jump of \( a \). In contrast, by S. B. Cooper [3] and C. E. M. Yates (unpublished), there exist nonzero noncuppable degrees in \( \mathcal{R} \) — the class of all c.e. degrees. Hence there is no hope for such a join theorem to hold in \( \mathcal{R} \). However, we can still salvage the idea by applying the jump operator one more time. The main theorem of this paper establishes the strongest possible analogue for the c.e. degrees of the Posner-Robinson join theorem [9].

Theorem 1.1 (Join theorem for c.e. degrees). For any c.e. degree \( w \), if \( w \neq 0 \), then there is a c.e. degree \( a \) such that \((a \vee w)' = a'' = 0''.\) Equivalently, for any nonzero c.e. degree \( w \), there is a low\(_2\) c.e. degree \( a \) such that \( a \vee w \) is high.

Many interesting results follow from Theorem 1.1. The most significant one is that the jump classes low and low\(_2\) are not elementarily equivalent. To see that, we need to recall the notion of deep degrees and almost deep degrees, and we make some remarks about definable ideals along the way.

M. Bickford and C. Mills [1] introduced the notion of deep degree. They called a c.e. degree \( a \) deep if for every c.e. degree \( x \), \( x' = (a \vee x)' \), namely, joining with
a preserves the jump of every c.e. degree. S. Lempp and T. Slaman [3] showed that the only deep degree is 0. Extending the notion of deep degree, P. Cholak, M. Groszek and T. Slaman [2], page 900, called a c.e. degree a n-deep if for every c.e. degree x, x(n) = (a ∨ x)(n). Let DP_n be the set of all n-deep degrees. They stated without proof that, for all n, 0 is the only n-deep degree. This follows at once from our main result, Theorem 1.1. They also stated without proof that for every nonzero c.e. degree w there is a nonhigh c.e. degree a such that a ∨ w is high, and our main result strengthens this by showing that a can be chosen to be low_2.

In the same paper [2], Cholak, Groszek and Slaman introduced the notion of almost deep degrees, the c.e. degrees a such that for any low c.e. degree x, a ∨ x is low. They obtained the following result.

**Theorem 1.2** (Cholak, Groszek and Slaman [2]). There is a nonzero almost deep degree.

The motivation is to find definable ideals in the c.e. degrees. The collection of all almost deep degrees is a nontrivial ideal. However, it is not known to be definable, since it is defined in terms of low degrees. By a result of A. Nies, R. Shore and T. Slaman [3], all other jump classes (L_m (n ≥ 2) and H_n (n ≥ 0)) are definable. This raises the possibility that one could generalise the notion of almost deep degrees and get sequences of definable ideals. For example, one could consider the c.e. degrees a such that for every low_a c.e. degree x, a ∨ x is low_n: Define PL_n by

$$PL_n = \{a \in \mathcal{R} \mid (\forall x \in \mathcal{R})[x^{(n)} = 0^{(n)} \iff (a \lor x)^{(n)} = 0^{(n)}]\}.$$  

Or one could consider the c.e. degrees a such that for any c.e. degree x, a ∨ x is high_n implies x is high_n: Define PH_n by

$$PH_n = \{a \in \mathcal{R} \mid (\forall x \in \mathcal{R})[x^{(n)} = 0^{(n+1)} \iff (a \lor x)^{(n+1)} = 0^{(n+1)}]\}.$$  

By the definability of jump classes other than low_1, for each n ≥ 1, both PL_n+1 and PH_n are definable ideals of \mathcal{R}. Cholak, Groszek and Slaman [2] stated without proof (as remarked above) that PH_1 = \{0\}. They also stated that it is conceivable that PL_2 ≠ \{0\}, and the same may hold for PL_n, n ≥ 3. By Theorem 1.1 we now know that all such ideals are trivial:

**Corollary 1.3.** For each n ≥ 1,

$$DP_n = PL_{n+1} = PH_n = \{0\}.$$  

Although Theorem 1.1 rules out some approaches to producing nontrivial definable ideals, it sheds some light on the elementary equivalence problems of jump classes. By making use of splitting properties, it is known that for all pairs (m, n), if m ≤ 2 and n ≥ 2, then Th(L_m) ≠ Th(L_n) (see the discussion in Li [7]). Recently, after showing that Th(H_1) ≠ Th(H_n) (7), Li [7.5] raises the question again:

1. Are there any m ≠ n such that Th(H_m) = Th(H_n)?
2. Are there any m ≠ n such that Th(L_m) = Th(L_n)?
3. In particular, are the low_1 and low_2 c.e. degrees elementarily equivalent?

By Theorem 1.1 and Theorem 1.2 we can answer (3), whereas the other two remain open.

**Corollary 1.4.** The low_1 and low_2 c.e. degrees are not elementarily equivalent.
In fact, it follows by the same argument that for all \( n \geq 2 \) the low\(_1\) and low\(_n\) c.e. degrees are not elementarily equivalent.

The rest of the paper is devoted to the proof of Theorem I and it is organised as follows. In section 2, we describe the requirements and the strategies to satisfy the requirements; in section 3, we describe the priority tree of strategies and describe the full construction; and finally in section 4, we verify that all requirements are satisfied.

Our notation and terminology are standard and generally follow Soare [10]. We assume that the reader is familiar with tree constructions. We say that a number is fresh at a given point in the construction if it is the least natural number greater than any number mentioned so far.

### 2. Requirements and strategies

#### 2.1. The requirements

Fix a c.e. set \( W \). Our construction will be nonuniform. We construct c.e. sets \( A \) and \( B_e \) together with Turing functionals \( \Gamma, \Delta_e \) and a computable partial function \( \Omega_{e,i} \) such that, for each \( e, i \in \omega \), either \( A^e \leq_T (A \oplus W)' \) via the functional \( \Gamma \), or \( B'_e \leq_T B_e \oplus W \) via \( \Delta_e \), or (the characteristic function of) \( W \) is computed by \( \Omega_{e,i} \). Notice that if \( B'_e \leq_T B_e \oplus W \), then we can simply take \( A \) to be \( B_e \) to establish Theorem I. Also, if \( a'' \leq (a \lor w)' \) for c.e. degrees \( a \) and \( w \), then \( 0'' \leq a'' \leq (a \lor w)' \leq 0'' \) so \( a'' = (a \lor w)' = 0'' \). Thus it suffices to satisfy the following requirements:

- **\( R_e \):** \( \text{Tot}^4(e) = \lim_y \Gamma(A, W; e, y) \) or \( (\exists B_e, \Delta_e)[\Delta_e(B_e, W)] \) total and \((\forall i)S_{e,i}\),
- **\( S_{e,i} \):** \( W = \Omega_{e,i} \) or \( \Delta_e(B_e, W; i) = B'_e(i) \),

where \( e, i \in \omega \), \( \text{Tot}^4 = \{ e \mid \Phi_e(A) \text{ is total} \} \), \( B'_e = \{ i \mid \Psi_i(B_e; i) \} \} \), \( \{ \Phi_e \mid e \in \omega \} \) and \( \{ \Psi_i \mid i \in \omega \} \) are fixed effective enumerations of all Turing functionals.

We will use a tree construction, but we first describe how individual requirements are met.

#### 2.2. The basic module for \( R_e \)

Let \( \alpha \) be a node labelled \( R_e \). Roughly speaking, \( R_e \) has two basic responsibilities. First, \( R_e \) defines the functional \( \Gamma(A, W; e, y) \) for more and more values of \( y \) and it also corrects the wrong values of \( \Gamma(A, W; e, z) \) which were defined by nodes to the right of \( \alpha \); this is similar to making \( A \oplus W \) high. Second, \( R_e \) preserves more and more \( \Phi_e(A; x) \) computations; this is similar to making \( A \) low\(_2\). Before we make the \( R_e \)-strategy more precise, let us make some conventions on building \( \Gamma \), called \( \Gamma \)-rules.

Whenever we define \( \Gamma(A, W; e, y) \), we define its use \( \gamma(e, y) \) as fresh, and locate it at some node of the priority tree. If \( \gamma(e, y) \) is enumerated into \( A \), then \( \Gamma(A, W; e, y) \) is set to be undefined automatically. If \( W \uplus (\gamma(e, y) + 1) \) changes, then unless we explicitly set \( \Gamma(A, W; e, y) \) to be undefined, \( \Gamma(A, W; e, y) \) is redefined with the same value and the same use automatically. We do the same if \( A \uplus \gamma(e, y) \) changes. The \( \Gamma \)-rules will ensure that actions of other requirements and irrelevant \( W \)-changes do not make \( \Gamma(A, W) \) nontotal.

Returning to the \( R_e \)-strategy, we define the length function by

\[
\ell(e) = \max\{ x \mid (\forall y < x)[\Phi_e(A; y)] \}
\]

We say that \( s \) is \( R_e \)-expansionary if \( \ell(e)[s] > \ell(e)[v] \) for all \( v < s \). At non-\( R_e \)-expansionary stages, we will define \( \Gamma(A, W; e, y) = 0 \) for more and more \( y \). This ensures that if there are only finitely many \( R_e \)-expansionary stages, then
Suppose that there are infinitely many \( R_e \)-expansionary stages. Then the \( S_{e;i} \)-strategies of \( R_e \) will try to correct the values of \( \Gamma(A,W;e,y) \), at the risk of destroying some computation \( \Phi_e(A;x) \) which should be preserved. The hope is to build a c.e. set \( B_e \), and a Turing functional \( \Delta_e \) to try to satisfy \( S_{e;i} \) for all \( i \in \omega \).

2.3. The basic module for \( S_{e;i} \). An \( S_{e;i} \)-strategy usually works at \( R_e \)-expansionary stages. The goals for \( S_{e;i} \) are: Keep \( \Delta_e(B_e,W;i) = B'_e(i) \), which is similar to making \( B_e \oplus W \) complete; and preserve \( \Psi_i(B_e;i) \), which is similar to making \( B_e \) low. We define \( \Delta_e \) as follows. If \( \Delta_e(B_e,W;i) \) is currently undefined, we define it to be \( B'_e[i](s) \) with fresh use \( \delta_e(i) \). \( \Delta_e \) will have its \( \Delta \)-rules similar to those for \( \Gamma \), in order to ensure that actions of other requirements and irrelevant \( W \)-changes will not make \( \Delta_e(B_e,W;i) \) non-total. Whenever \( \Psi_i(B_e;i) \) becomes convergent, \( S_{e;i} \) has the ability to put \( \delta_e(i) \) into \( B_e \) to correct \( \Delta_e(B_e,W;i) \). However, \( S_{e;i} \) will not do this immediately but rather “open a gap”, in a fashion to be described. If \( W \)-changes below \( \delta_e(i) + 1 \) during the gap, then \( \Delta_e(B_e,W;i) \) can be corrected and the markers \( \delta_e(j) \) can be moved to numbers bigger than \( \psi_e(i) \) without putting their current positions into \( B_e \). This provides a finitary win for \( S_{e;i} \). If such \( W \)-changes do not occur, then \( S_{e;i} \) puts \( \delta_e(i) \) into \( B_e \) to correct \( \Delta_e(B_e,W;i) \), and in this case, progress is made towards showing that \( W \) is computable.

2.4. Description of one gap/cogap strategy. We now see how a single \( R_e \) can be combined with its \( S_{e;i} \)-substrategies. (Thus, we treat \( e \) as fixed and \( i \) as variable in this discussion.) \( S_{e;i} \) will have an auxiliary set \( C_i \) consisting of the numbers \( x \) such that \( \Phi_e(A;x) \) is defined and preserved (meaning: able to survive the \( \Gamma(A,W;e,y) \) correction). Let us call \( C_i \) the clearing set of \( S_{e;i} \). \( S_{e;i} \) also builds the computable partial function \( \Omega_{e;i} \), hoping to establish the computability of \( W \). At every stage, the domain of \( \Omega_{e;i} \) is a finite initial segment of the natural numbers.

\( S_{e;i} \) will be a gap/cogap strategy, which has the following parameters:

- \( p(i) \): the largest number in the domain of \( \Omega_{e;i} \). If the domain of \( \Omega_{e;i} \) is empty, let \( p(i) = -1 \);
- \( r(i) \): the \( A \)-restraint imposed by \( S_{e;i} \) during \( A \)-cogaps of \( S_{e;i} \).

The \( S_{e;i} \)-strategy proceeds as follows:

1. (Opening an \( A \)-gap) Wait for an \( R_e \)-expansionary stage, say \( v \), at which the \( A \)-gap is currently closed: \( \Psi_i(B_e;i) \downarrow \) and either \( \Delta_e(B_e,W;i) = 0 \) or \( \delta_e(i) \leq \psi_e(i) \). Then open an \( A \)-gap as follows:
   - Drop its restraint on \( A \) by setting \( r(i) = -1 \).
   - For all \( y \), if \( \Gamma(A,W;e,y) = 0 \) and \( \Gamma(A,W;e,y) \) is located at some node corresponding to an \( S_{e;j} \) for \( j > i \) and \( \gamma(e,y) > R(i) = \max\{r(i') \mid i' \leq i\} \), then enumerate \( \gamma(e,y) \) into \( A \). For each \( j > i \), set \( C_j = \emptyset \).
   - For all \( z \leq \delta_e(i)[s] \) with \( \Omega_{e;i}(z) \) not already defined, define \( \Omega_{e;i}(z) = W(z)[s] \). (We threaten to make \( \Omega_{e;i} = W \).)
   - Let \( y \) be the least \( z \) such that \( \Gamma(A,W;e,z) \uparrow \), define \( \Gamma(A,W;e,y) = 1 \) with \( \gamma(e,y) \) fresh, and locate it at the node corresponding to \( S_{e;i} \). The last two actions correct the values of \( \Gamma(A,W;e,y) \) which are wrongly defined to be zero at some node to the right, except those “controlled” by higher priority nodes.
holds eventually and permanently. So we consider only the following two cases:

Case 2a. (Successful Closure) $W_s \upharpoonright (\delta_e(i) + 1) \neq W_s \upharpoonright (\delta_e(i) + 1)$.

Then: For all $j \geq i$, if $\Delta_e(B_e; W; j)$ is defined, then set it to be undefined and stop. (By doing so, we lift all $\delta_e(j)$ beyond $\psi_e(i)$ hence preserve $\Psi_e(B_e; i) \downarrow$, and we are able to redefine $\Delta_e(B_e; W; i) = 1$ correctly. So $S_{e,i}$ is satisfied finitarily.)

Case 2b. (Unsuccessful Closure) Otherwise, then:

- Let $x$ be the least number which is not in $C_i$, define the $A$-restraint $r(i) = \varphi_e(x)$, and enumerate $\delta_e(j)$ into $B_e$ for all $j \geq i$. (We set $r(i) = \varphi_e(x)$ in order to preserve $\Phi_e(A; x)$ until a $W$-change (if any) gives an opportunity to add $x$ to $C_i$ as described in the next paragraph. Note that $\Phi_e(A; x)$ is currently defined because we are at an $R_e$-expansory stage, and all numbers $y$ previously added to $C_i$ were such that $\Phi_e(A; y)$ was defined when $y$ was added to $C_i$. Note also that for all $y$ with $\gamma(e, y) \downarrow$, either $\gamma(e, y) \leq R(i)$ or $\gamma(e, y) > p(i)$, where $p(i)$ is the greatest element of the domain of $\Omega_{e,i}$)

(3) (Building $C_i$) At any stage $t$ where $r(i) \neq -1$ (not necessarily an $R_e$-expansory stage) if $W_s \upharpoonright p(i) \neq W_t \upharpoonright p(i)$, where $p(i)$ is the largest number in the domain of $\Omega_{e,i}$, then:

- For any $e, y$, if $\Gamma(A, W; e, y)$ is defined and located at some node to the right of the current one, and $\gamma(e, y) \geq p(i)$, then set $\Gamma(A, W; e, y)$ to be undefined.
- Enumerate the least number $x$ which is not in $C_i$ into $C_i$, and restart $\Omega_{e,i}$ completely. Note that, because of the restraint imposed at the last stage when the $A$-gap was (unsuccessfully) closed, $\Phi_e(A, x) \downarrow$. Furthermore, all defined marker positions $\gamma(e, y)$ are either $\leq R(i)$ or are located at some node for $S_{e,j}$ for $j < i$. Thus, if no requirement $S_{e,j}$ for $j < i$ acts at any future stage, the computation $\Phi_e(A, x)$ is permanent, and does not require any restraint to be preserved.
- Set $r(i) = -1$.

We now analyse the possible outcomes for the strategy $S_{e,i}$, under the assumption that $R_e$ has infinitely many expansory stages and $S_{e,i}$ never acts for all $i' < i$. It is easy to see that if $S_{e,i}$ acts only finitely many times, then $\Delta_e(B_e; W; i) = B_{e}(i)$ holds eventually and permanently. So we consider only the following two cases:

Case 1. Step (3) occurs infinitely many times.

By the strategy, at the stage at which $x$ is enumerated into $C_i$, $\Phi_e(A; x)$ is defined and cleared from all $\gamma$-markers located at some node to the right of that for $S_{e,i}$. Hence we are able to ensure that $\Phi_e(A)$ is total. On the other hand, Step (1) ensures that for almost every $y$, $\Gamma(A, W; e, y)$ is correctly defined to be equal to 1. Also $(\lambda y)\Gamma(A, W; e, y)$ is total, and hence $\lim_y \Gamma(A, W; e, y) = 1$. $R_e$ is satisfied through the first clause.

Case 2. Otherwise, that is, after some stage $W$ never changes below $p(i)$ during any $A$-cogap. By assumption, $S_{e,i}$ never closes any $A$-gap successfully. Hence $W$ never changes below $\delta_e(i)$, which is larger than $p(i)$, during any $A$-gap. Since $p(i)[s]$ is nondecreasing in $s$ and unbounded over the construction, and $W_s \upharpoonright p(i)[s] = W \upharpoonright p(i)[s]$, $W$ is computable.
2.5. **Coordination among \( R \)-strategies.** In general, we have to protect computations of the form \( \Phi_e(A; x) \) of the \( R_e \)-strategy from injuries by other \( R'_e \)-strategies. If \( R' \) has higher priority, the usual “believable computation” trick works. If \( R' \) has lower priority than \( R \), then it can be done by the slow-down method (see for example Cooper and Li [4]). Let \( \xi \) be a node extending a node \( \alpha \langle g \rangle \) working on the \( S_{e,i} \)-strategy. We say that \( \xi \) is ready to define \( \Gamma(A, W; d, y) \), if for \( l = \max \{ b(\xi), d, y, m \}, l \in C_i \), where \( b(\xi) \) is a natural number coding the node \( \xi \) and \( m \) is the number of times that \( \xi \) has defined \( \Gamma(A, W; d, y) \). If \( \xi \) is not ready, then it simply waits. This ensures that for each \( x \) there are at most finitely many times that \( x \) is removed from \( C_i \) because the convergence of \( \Phi_e(A; x) \) is injured by the enumeration of some number less than \( \varphi_e(x) \) by some node \( \xi \) extending \( \alpha \langle g \rangle \). Hence if the \( S_{e,i} \)-strategy puts numbers into \( C_i \) infinitely often, then \( C_i = \omega \) at the end of the construction, i.e. every \( x \in \omega \) eventually remains in \( C_i \). And since \( C_i = \omega \), the delay in defining \( \Gamma(d, y) \) caused by \( S_{e,i} \) will not last forever.

3. **Priority tree and construction**

3.1. **Priority tree.** We first define the priority tree \( T \), which is \( (\omega+1) \)-branching. The tree \( T \) consists of all finite sequences of elements of the set \( \{ g_0, g_1, \ldots, 1 \} \). For each node \( \alpha \) on \( T \), \( \alpha \) has the following outgoing edges ordered from left to right:

\[
g_0 < g_1 < L_1 < \cdots < L_1 1.
\]

Fix a priority ranking of the requirements \( R_e \) by

\[
R_0 < R_1 < R_2 < \cdots
\]

For each node \( \alpha \) on \( T \), if the length of \( \alpha \) is \( e \), then we label \( \alpha \) with the requirement \( R_e \), and we associate with each node \( \alpha \langle g \rangle \) the subrequirement \( S_{e,i} \), and we associate with \( \alpha \langle 1 \rangle \) the outcome that there are only finitely many \( \alpha \)-expansionary stages, as defined below.

3.2. **Parameters and conventions.** Suppose that \( \alpha \) is a node on \( T \). \( \alpha \) has the following parameters:

(a) \( B_{\alpha} \): the c.e. set built by \( \alpha \);
(b) \( \Delta_{\alpha} \): the Turing functional built by \( \alpha \);
(c) \( b(\alpha) \): where \( b : T \rightarrow \omega \) is a 1-1 computable function;
(d) \( R(\alpha) \): the maximum \( A \)-restraint imposed by nodes to the left of \( \alpha \) or extended by \( \alpha \).

For each \( i \in \omega \) and \( \alpha \in T \) of length \( e \), we associate with the node \( \alpha \langle g \rangle \) the following parameters:

(a) \( C^\alpha_i \): the clearing set of \( \alpha \) for \( S_{e,i} \);
(b) \( r^\alpha(i) \): the \( A \)-restraint on node \( \alpha \) imposed by the \( S_{e,i} \)-strategy as described before;
(c) \( \Omega^\alpha_i \): the computable partial function built at the node \( \alpha \langle g \rangle \);
(d) \( p^\alpha(i) \): the largest number in the domain of \( \Omega^\alpha_i \) (it is \(-1 \) if \( \Omega^\alpha_i \) is empty or undefined).

**Definition 3.1.** Given an \( R_e \)-strategy \( \alpha \) and a stage \( s \):

(i) We say that \( \Phi_e(A; x) = y \) is \( \alpha \)-believable, if for all \( m, n \), if \( \Gamma(A, W; m, n) \) is defined and located at some node to the right of \( \alpha \), then either \( \gamma(m, n) > \varphi_e(x) \) or \( \gamma(m, n) \leq R(\alpha) \).
(ii) We define the length function \( l(\alpha) \) by
\[
l(\alpha) = \max\{x \mid (\forall y < x)[\Phi_e(A; y) \downarrow \text{via an } \alpha\text{-believable computation}]\}.
\]

(iii) We say that a stage \( s \) is \( \alpha \)-expansionary if \( \alpha \) is accessible at stage \( s \) and \( l(\alpha)[s] > l(\alpha)[v] \) for all \( v < s \) at which \( \alpha \) was accessible.

**Definition 3.2.** Given a node \( \alpha \), and \( e, y \in \omega \), let \( m(\alpha, e, y) \) be the number of times that \( \alpha \) has defined \( \Gamma(A, W; e, y) \), and define \( d(\alpha, e, y) = \max\{b(\alpha), e, y, m(\alpha, e, y)\} \).

We say that \( \alpha \) is ready to define \( \Gamma(A, W; e, y) \) if for each \( R \)-strategy \( \alpha' \) and each \( i \in \omega \) and if the node \( \alpha'^-\langle g_i \rangle \subseteq \alpha \), then \( d(\alpha, x, y) \in C_i'^\uparrow \), the clearing set of \( \alpha' \) for strategy \( S_{\epsilon, i} \), where \( R_{\epsilon'} \) is the requirement associated with \( \alpha' \).

Suppose that \( \alpha \) is a \( R \)-strategy. If \( \alpha \) is initialised, then all of its nodes \( \alpha^-\langle g_i \rangle \) are initialised, and both \( B_\alpha \) and \( \Delta_\alpha \) are set to be \( \emptyset \).

Suppose that \( \alpha'^-\langle g_i \rangle \) for some \( i \in \omega \) is a node coming out of \( \alpha \) and associated with \( S_{\epsilon, i} \). If it is initialised, then both \( C_i'^\uparrow \) and \( \Omega_i'^\uparrow \) are set to be \( \emptyset \), any gap is set to be closed and \( r^\alpha(i) \) is set to be undefined.

Without loss of generality, we suppose that the c.e. set \( W \) is enumerated by \( \{W_s\}_{s \in \omega} \) such that, for all \( n \), both \( W_{2n+1} = W_{2n} \) and \( |W_{2n+2} - W_{2n+1}| = 1 \) hold. Finally we require that the \( \Gamma \)-rules and \( \Delta \)-rules described in section 2 be followed automatically.

### 3.3. Stage-by-stage construction.

The construction proceeds in stages as follows:

**Stage** \( s = 0 \). Set \( A = \Gamma = \emptyset \).

**Stage** \( s = 2n+1 \). We first describe a node \( \alpha \) being accessible at stage \( s \) inductively on substages \( t \) of stage \( s \). First we allow the root node \( \emptyset \) to be accessible at substage \( t = 0 \).

**Substage** \( t \). Let \( \alpha \) be accessible at substage \( t \) of stage \( s \) and let \( \alpha \) be labelled \( R_e \) for some \( e \in \omega \). If \( t = s \), then initialise all \( \xi \) with \( \alpha <_L \xi \) and go to stage \( s + 1 \); if \( t < s \), then proceed as follows.

**Case 1.** \( s \) is not \( \alpha \)-expansionary.

Let \( \alpha^-\langle 1 \rangle \) be accessible. Let \( y \) be the least \( z \) such that \( \Gamma(A, W; e, z) \) is undefined.

**Case 1a.** \( \alpha \) is ready to define \( \Gamma(A, W; e, y) \). Then

- for any \( m, n \), if \( \Gamma(A, W; m, n) \) is defined and located at some \( \xi \) with \( \alpha^-\langle 1 \rangle <_L \xi \) and \( \gamma(m, n) > R(\alpha^-\langle 1 \rangle) \), then enumerate \( \gamma(m, n) \) into \( A \), and initialise all \( \xi \) with \( \alpha <_L \xi \);

- define \( \Gamma(A, W; e, y) = 0 \) with \( \gamma(e, y) \) fresh, and locate it at \( \alpha^-\langle 1 \rangle \);

- go to substage \( t + 1 \).

**Case 1b.** Otherwise, then initialise all \( \xi \) with \( \alpha <_L \xi \), and go to substage \( t + 1 \).

**Case 2.** \( s \) is \( \alpha \)-expansionary.

Choose the least \( i \) such that the strategy \( S_{\epsilon, i} \) is either ready to open or ready to close an \( A \)-gap at stage \( s \), or such that \( \Delta_\alpha(B_\alpha, W; i)[s] \) is undefined, and act accordingly as described below. (Such an \( i \) exists because \( \Delta_\alpha(B_\alpha, W; j)[s] \) is undefined for all sufficiently large \( j \).) Initialise all \( \xi \) with \( \alpha^-\langle g_i \rangle <_L \xi \). If \( \Delta_\alpha(B_\alpha, W; i)[s] \) is undefined, define it to be \( B_\alpha^\prime(i)[s] \), and go to stage \( s + 1 \). Otherwise, open or close a gap as described below.

We say that the \( S_{\epsilon, i} \)-strategy is ready to open an \( A \)-gap at stage \( s \), if the gap is closed at the beginning of stage \( s \), \( \Psi_i(B_\alpha; i)[s] \downarrow \), and either \( \Delta_\alpha(B_\alpha, W; i) = 0[s] \) or
for some $j \geq i$, $\delta_s(j) < \psi_i(i)$. If this holds, we open an $A$-gap as follows:

- set $r^\alpha(i) = -1$;
- define $p^\alpha(i) = \delta_\alpha(i)$ and for each $z \leq p^\alpha(i)$ define $\Omega^\alpha_z(z) = W(z)[s]$ if it is currently undefined;
- for any $m, n$, if $\Gamma(A, W; m, n) = 0$ via an axiom located at some $\eta$ with $\alpha^-\langle g_i \rangle < L \eta$ and $\gamma(m, n) > R(\alpha^-\langle g_i \rangle)$, then enumerate $\gamma(m, n)$ into $A$;
- (Building $\Gamma$) let $y$ be the least $z$ such that $\Gamma(A, W; e, z) \uparrow$. If $S_{e,i}$ is ready to define $\Gamma(A, W; e, y)$, then define $\Gamma(A, W; e, y) = 1$ with $\gamma(e, y)$ fresh, and locate it at $\alpha^-\langle g_i \rangle$;
- (Updating $C^\beta_j$) for each node $\beta^-\langle g_j \rangle$ on $T$ properly extended by $\alpha^-\langle g_i \rangle$ and working on, say, $S_{e,j}$, if there exists $x \in C^\beta_j$ with $\Phi_c(A, x) \uparrow$, let $x_0$ be the least such $x$ and remove from $C^\beta_j$ all $y \geq x_0$ currently in $C^\beta_j$.
- go to substage $t + 1$.

We say that the $S_{e,i}$-strategy is ready to close an $A$-gap at stage $s$ if a gap is currently open. Let $s^-$ be the stage at which the current gap was opened; we close the $A$-gap in a manner depending on the two cases below.

Case 2a. (Successful Closure) $W_{s^-} \uparrow (p^\alpha(i) + 1) \neq W_s \uparrow (p^\alpha(i) + 1)$. Then:

- for every $j \geq i$, set $\Delta_\alpha(B_\alpha, W; j)$ to be undefined, and we say that $\delta_\alpha(j)$ is lifted at stage $s$;
- cancel the partial function $\Omega^\alpha_{s^+}$;
- initialise all $\xi$ with $\alpha^-\langle g_i \rangle < L \xi$ and go to stage $s + 1$.

Case 2b. (Unsuccessful Closure) Otherwise, then

- let $x$ be the least number $\not\in C^\alpha_i$ and define $r^\alpha(i) = \varphi_c(A; x)$ (note that $\Phi_c(A; x) \downarrow$ via an $\alpha$-believable computation as stage $s$ is $\alpha$-expansionary);
- enumerate $\delta_\alpha(j)$ into $B_\alpha$ for every $j \geq i$ with $\delta_\alpha(j) \downarrow$ and say that $\delta_\alpha(j)$ is lifted at stage $s$;
- initialise all $\xi$ with $\alpha^-\langle g_i \rangle < L \xi$ and go to stage $s + 1$.

Stage $s = 2n + 2$. Let $w_s$ be the number $w \in W_s - W_{s-1}$. For each $S_{e,i}$-strategy associated with a node, say $\alpha^-\langle g_i \rangle$, such that $r^\alpha(i) \downarrow \neq -1$ and $p^\alpha(i) \geq w_s$, from left to right (in fact, the order is not essential as there are no conflicts) execute the following cogap permitting action of $\alpha^-\langle g_i \rangle$ provided that $S_{e,i}$ is not in a gap:

1. set $r^\alpha(i) = -1$;
2. for all $y$, if $\Gamma(A, W; e, y)$ is defined and located at some node to the right of $\alpha^-\langle g_i \rangle$ and $\gamma(e, y) > w_s$, then set $\Gamma(A, W; e, y)$ to be undefined;
3. enumerate the least number $x$ which is not in $C^\alpha_i$ into $C^\alpha_i$, and cancel $\Omega^\alpha_{s^+}$. Note that $\Phi_c(A; x) \downarrow$ via an $\alpha$-believable computation because of the restraint $r^\alpha(i)$ imposed the last time that $S_{e,i}$ unsuccessfully closed an $A$-gap.

This completes the description of the construction.

4. The verification

Let the true path $TP$ be the leftmost path through $T$ consisting of nodes which are accessible infinitely often.

**Lemma 4.1** (Existence of the true path). Suppose that there is no c.e. set $B$ such that $B' \leq_T B \oplus W$. Then the true path $TP$ is an infinite path through the tree $T$.
Proof. We show by induction on $e$ that there is a leftmost node $\alpha_e$ of length $e$ which is accessible infinitely often. It then follows easily that $\alpha_{e+1}$ extends $\alpha_e$ for all $e$, and clearly $TP = \{ \alpha_e : e \in \omega \}$. Clearly we may let $\alpha_0$ be the empty node. Now suppose inductively that $\alpha_e$ exists and let it be denoted $\alpha$. Let $s_0$ be a stage after which no node to the left of $\alpha$ is accessible. We consider two cases.

**Case 1.** There are only finitely many $\alpha$-expansionary stages.

Clearly $\alpha^-(1)$ is accessible infinitely many times, and is initialised only finitely many times and so is the leftmost node of length $e + 1$ which is accessible infinitely often.

**Case 2.** There are infinitely many $\alpha$-expansionary stages.

**Claim.** If for all $i \in \omega$ the node $\beta = \alpha^-(g_i)$ is accessible only finitely often, then for all $i$, $\Delta_\alpha(B_\alpha,W;i)[s]$ is defined and equal to $B'_\alpha(i)$.

**Proof of the Claim.** First note that, by induction on $i$, $\Delta_\alpha(B_\alpha,W;i)[s]$ is defined for all sufficiently large stages at which $\alpha$ is accessible. Now fix $i$. By hypothesis there is a stage $s_1$ after which no node $\alpha^-(g_i')$ is accessible for any $i' \leq i$. We may also assume that $s_1$ is so large that $\Delta_\alpha(B_\alpha,W;j)[s]$ is defined for all $j \leq i$ and $s \geq s_1$ such that $\alpha$ is accessible at $s$. First, suppose that $i \in B'_\alpha[s_2]$ for some stage $s_2 > s_1$ at which $\alpha$ is accessible. Then $\Delta_\alpha(B_\alpha,W;i)[s_2] = 1$ and $\delta_i(j) \geq \psi_i(i)[s_2]$ for all $j \geq i$ such that $\delta_i(j) \downarrow$, since otherwise $\alpha^-(g_i')$ would be accessible at $s_2$ for some $i' \leq i$, contrary to the choice of $s_1$. It follows that both the computations $i \in B'_\alpha[s_2]$ and $\Delta_\alpha(B_\alpha,W;i)[s_2] = 1$ are permanent. To see that the first is permanent, note that no marker position $\delta_i(i')$ for $i' \leq i$ can enter $B_\alpha$ after $s_2$ since no node $\alpha^-(g_i')$ for $i' \leq i$ is accessible after $s_2$. The marker positions $\delta_i(j)$ for $j \geq i$ are greater than the use $\psi_i(i)$ of the computation showing $i \in B'_\alpha[s_2]$ and remain greater than this use. To see that the second computation $\Delta_\alpha(B_\alpha,W;i)[s_2] = 1$ is never injured after stage $s_2$, let $s_3$ be the stage at which it was originally defined with the same use as at $s_2$. At stage $s_3$ all nodes $\alpha^-(j)$ for $j > i$ were initialised, and marker positions $\delta_i(j)$ for $j > i$ are chosen greater than its use and remain greater than its use. Since no node $\alpha^-(g_i')$ for $i' < i$ is accessible between stages $s_3$ and $s_2$ (or else $S_{e,i}$ would be initialised and the use of $\Delta_\alpha(B_\alpha,W;i)$ would change between $s_3$ and $s_2$), no marker position $\delta_i(i')$ for $i' < i$ enters $B_\alpha$ between $s_3$ and $s_2$. Finally, $W$-changes do not affect the computation by the $\Delta$-rules and the fact that no node $\alpha^-(g_i')$ for $i' < i$ is accessible after $s_2$. Thus, if $i \in B'_\alpha[s_2]$ for some stage $s_2 > s_1$ at which $\alpha$ is accessible, $\Delta_\alpha(B_\alpha,W;i) = 1 = B'_\alpha(i).

Now suppose there is no such stage $s_2$. Since $\alpha$ is accessible at infinitely many stages, it follows that $B'_\alpha(i) = 0$. Recall that $\Delta_\alpha(B_\alpha,W;i)[s]$ is defined for all $s \geq s_2$. Since there are only finitely many stages $s$ at which $\alpha$ is accessible and $i \in B'_\alpha[s]$, there are only finitely many stages $s$ at which $\alpha$ is accessible and $\Delta_\alpha(B_\alpha,W;i)[s] = 1$. Also, there are only finitely many stages at which $\Delta_\alpha(B_\alpha,W;i)[s]$ is set equal to 0 since there are only finitely many stages at which $\alpha^-(g_i)$ is accessible. Let $s_4$ be the last stage at which $\Delta_\alpha(B_\alpha,W;i)$ is set equal to 0. Then no marker position $\delta_i(i')$ for $i' < i$ enters $B_\alpha$ after $s_4$ for any $i' < i$, since otherwise $\alpha^-(g_i')$ would be initialised after $s_4$. The marker positions $\delta_i(j)$ for $j > i$ become undefined at $s_4$ and, when defined at any stage after $s_4$, take values larger than the use of $\Delta_\alpha(B_\alpha,W;i)$ at $s_4$. The marker position $\delta_i(i)$ does not enter $B_\alpha$ after $s_4$ because otherwise $\Delta_\alpha(B_\alpha,W;i)[s]$ would again be set equal to 0 after $s_4$, contrary to the choice of $s_4$. 


Thus there is an (least) $i$ such that the strategy $S_{e,i}$ associated with $\alpha^-(g_i)$ acts infinitely often. In particular, the $A$-gap will be opened infinitely often. Consequently the node $\alpha^-(g_i)$ is accessible infinitely often.

**Lemma 4.2.** For any node $\alpha^-(g_i)$ on $TP$, if there is a stage after which there is no cogap permitting action for $\alpha^-(g_i)$, then $W$ is computable.

*Proof.* Let $\alpha^-(g_i)$ be a string on $TP$ such that after some stage, say $s$, there is no cogap permitting for $\alpha^-(g_i)$. We may assume also that $\alpha^-(g_i)$ is never initialised after $s$. Observe that the associated $S_{e,i}$-strategy must open and close the $A$-gap infinitely often, and each closure after $s$ must be an unsuccessful one. Thus $\Omega^s_\alpha$ is never cancelled after stage $s$, and $p^\alpha(i)[t] = \delta_g(i)[t]$ goes to infinity as $t$ goes to infinity. For each $z$, once $\Omega^s_\alpha(z)$ is defined after $s$, $W \upharpoonright z$ never changes: It cannot change during an $A$-gap since otherwise the closure would be successful; it does not change during an $A$-cogap by the choice of $s$. Therefore $\Omega^s_\alpha = W$. 

**Lemma 4.3.** Suppose that there is no c.e. set $B$ such that $B \leq_T B \oplus W$ and $W$ is not computable. Then if $\alpha^-(g_i)$ is on $TP$, $C_i^\alpha = \omega$; more precisely, for any $x \in \omega$ there is a stage $s_x$ after which $x \in C_i^\alpha$. Hence $\Phi_e(A)$ is total, where $\alpha$ is associated with $R_e$.

*Proof.* Consider $\alpha^-(g_i)$ on $TP$. Let $s_0$ be a stage after which no node to the left of $\alpha^-(g_i)$ is accessible.

We first prove that for each $x$, there are only finitely many stages at which $x$ is removed from $C_i^\alpha$. This is proved by induction on $x$. Fix $x$, and consider a stage $s$ after which no $x' < x$ is removed from $C_i^\alpha$. Since $\alpha^-(g_i) \in TP$ we may assume also that $s$ is so large that $\alpha^-(g_i)$ is not initialised after $s$. Suppose that $x$ is removed from $C_i^\alpha$ at stage $t > s$. Then some $\gamma(m,n) < \varphi_e(x)$ is enumerated into $A$ at stage $t$. Let $\gamma(m,n)$ be located at $\eta$. If $\eta$ is to the left of $\alpha^-(g_i)$ or $\eta$ is extended by $\alpha^-(g_i)$ or $\eta = \alpha^-(g_i)$, then $\alpha^-(g_i)$ is initialised at $t$, as some node to the left of $\eta$ acts at $t$. Thus $\eta$ is to the right of $\alpha^-(g_i)$, or $\eta$ extends $\alpha^-(g_i)$. We next prove that there are only finitely many stages $t$ such that $\eta$ (as described above) extends $\alpha^-(g_i)$. Suppose that $\eta$ extends $\alpha^-(g_i)$ and causes $x$ to be removed from $C_i^\alpha$ via the enumeration of $\gamma(m,n)$ (located at $\eta$) in $A$. Then $b(\eta) < x$ since otherwise $\eta$ must wait until $x \in C_i^\alpha$ to be ready to define $\gamma(m,n)$ and then must choose $\gamma(m,n) > \varphi_e(x)$. Similarly, $m < x$ and $n < x$. Thus, there are only finitely many choices for $n, m,$ and $n$. For fixed $n, m,$ and $n$ there are only finitely many stages at which the enumeration of $\gamma(m,n)$ located at $\eta$ into $A$ can cause $x$ to be removed from $C_i^\alpha$, since once $\gamma(m,n)$ has been defined by $\eta$ more than $x$ times, it must be chosen after $x$ has entered $C_i^\alpha$ and hence larger that $\varphi_e(x)$. This concludes the proof that there are only finitely many stages $t$ at which $x$ is removed from $C_i^\alpha$ by the enumeration of $\gamma(m,n)$ into $A$, where $\gamma(m,n)$ is located at some $\eta$ extending $\alpha^-(g_i)$. Let $s_1$ be a stage at which $S_{e,1}$ opens an $A$-gap and which is larger than all such $t$. At $s_1$ all markers $\gamma(m,n)$ located to the right of $\alpha^-(g_i)$ become undefined or take values greater than $p^\alpha(i)$. At any future stage at which $p^\alpha(i)$ increases, an $A$-gap is opened by $\alpha^-(g_i)$, and so again all such markers are undefined or greater than $p^\alpha(i)$. Thus, at all stages $s_2 > s_1$, all markers $\gamma(m,n)$ located to the right of $\alpha^-(g_i)$ are greater than $p^\alpha(i)$. If there is no stage $s_3 > s_1$ at which $x$ is added to $C_i^\alpha$, then clearly $x$ is removed from $C_i^\alpha$ at at most one stage $s_2 > s_1$. Suppose now that $x$ is added to $C_i^\alpha$ at stage $s_2 > s_1$. Then $\Phi_e(A; x)$ as defined at $s_2$, as remarked in the construction. At $s_2$, all $\gamma(m,n)$ located to the right of $\alpha^-(g_i)$ with $\gamma(m,n)$
greater than or equal to \( p^n(i) \) become undefined. As remarked above, these \( \gamma(m, n) \) are the only numbers which can injure the computation of \( \Phi_e(A; x) \). Hence \( x \) is never removed from \( C^\alpha_i \) after stage \( s_2 \). This completes the proof that \( x \) is removed from \( C^\alpha_i \) at most finitely often.

It is now easy to see that, for all \( x, x \in C^\alpha_i \) from some stage on. Assume this is false, and let \( x_0 \) be the least \( x \) for which this is false. Choose \( s_0 \) so large so that, for all \( s \geq s_0 \) no \( x \leq x_0 \) is added to or removed from \( C^\alpha_i \) at stage \( s \). Thus, at the beginning of each stage \( s \geq s_0 \), \( x \) is the least number not in \( C^\alpha_i \). Let \( s_1 \) be a stage \( s > s_1 \) at which cogap permitting action occurs for \( \alpha^\-helper(g_i) \). Such a stage \( s_1 \) exists by Lemma \ref{4.2}. Then \( x_0 \) is added to \( C^\alpha_i \) at \( s_1 \), which gives us the desired contradiction.

It remains to show that \( \Phi_e(A; x) \) is total. When \( x \) enters \( C^\alpha_i \) for the last time, \( \Phi_e(A; x) \) is defined. It remains defined forever thereafter with the same computation, since otherwise it would be removed from \( C^\alpha_i \).

\begin{lemma}
Suppose that \( W \notin C \emptyset \), and that for any c.e. set \( B, B' \notin C B \oplus W \).
Suppose that \( \alpha \) is a node on \( TP \) and \( i \in \omega \). Then:
\begin{enumerate}
  \item If \( \alpha^\-helper(1) \in TP \), then
    \begin{enumerate}
      \item \( (\lambda y)\Gamma(A, W; e, y) \) is total,
      \item \( \lim_y \Gamma(A, W; e, y) = 0 \), and
      \item \( \Phi_e(A) \) is not total.
    \end{enumerate}
  \item If \( \alpha^\-helper(g_i) \in TP \), then
    \begin{enumerate}
      \item \( (\lambda y)\Gamma(A, W; e, y) \) is total,
      \item \( \lim_y \Gamma(A, W; e, y) = 1 \), and
      \item \( \Phi_e(A) \) is total.
    \end{enumerate}
\end{enumerate}
\end{lemma}

\begin{proof}
For (i). Let \( s_0 \) be minimal after which \( \alpha^\-helper(1) \) is never initialised. Let \( r_0 = \max\{r(\xi)[s_0] \mid \xi < L \alpha^\-helper(1)\} \). By the construction, for any \( y \), if \( \Gamma(A, W; e, y) \) is defined and located at some node to the right of \( \alpha^\-helper(1) \) and \( \gamma(e, y) > r_0 \), then \( \gamma(e, y) \in A \). Thus for almost every \( y \), \( \Gamma(A, W; e, y) = 0 \) is defined by \( \alpha \) and located at \( \alpha^\-helper(1) \) eventually. Also, by the choice of \( s_0 \), once \( \Gamma(A, W; e, y) = 0 \) is defined after \( s_0 \) and located at \( \alpha^\-helper(1) \), it will never be destroyed, and hence it will be kept permanently. Since we always define \( \Gamma(A, W; e, y) \) for the least \( y \) for which \( \Gamma(A, W; e, y) \) is undefined, \( (\lambda y)\Gamma(A, W; e, y) \) is total. Hence, both (a) and (b) hold.
\( (c) \) follows from the fact that there are only finitely many \( \alpha \)-expansionary stages and that, for \( \alpha \) on the true path, if \( \Phi_e(A; x) \) is defined, it is eventually defined by an \( \alpha \)-believable computation.

For (ii). (a) and (b) follow from the proof for (i) above. (c) follows from Lemma \ref{4.2}.
\end{proof}

This completes the proof of Theorem \ref{4.1}

\begin{thebibliography}{9}
\bibitem{1} M. Bickford and C. F. Mills, Lowness properties of r.e. degrees, Unpublished preprint, 1982.
\bibitem{3} S. Barry Cooper, On a theorem of C. E. M. Yates, Handwritten notes, 1974.
\bibitem{4} S. Barry Cooper and Angsheng Li, Splitting and nonsplitting, II: A low \( e \) c.e. degree above which \( 0' \) is not splittable, \textit{J. Symbolic Logic}, 67:1391–1430, 2002. MR \texttt{2003k:03055}
\end{thebibliography}


Department of Mathematics, University of Illinois, 1409 W. Green St., Urbana, Illinois 61801

E-mail address: jockusch@math.uiuc.edu

Institute of Software, Chinese Academy of Sciences, P. O. Box 8718, Beijing, 100080, People’s Republic of China

E-mail address: angsheng@gcl.iscas.ac.cn

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

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