On the Universal Splitting Property*

Rod Downey
Mathematics Department
Victoria University
P. O. Box 600
Wellington
New Zealand

February 9, 1996

Abstract

We prove that if an incomplete computably enumerable set has the the universal splitting property then it is low_2. This solves a question from Ambos-Spies and Fejer [1] and Downey and Stob [7]. Some technical improvements are discussed.

1 Introduction

Two computably enumerable sets A_1 and A_2 are said to *split* A if $A = A_1 \cup A_2$ and $A_1 \cap A_2 = \emptyset$. We write $A_1 \sqcup A_2 = A$ in the case that A_1 and A_2 split A. Splitting theorems for computably enumerable sets have played a central role in the history of classical computability theory. For instance, Sack's splitting theorm [14], demonstrated that every nonzero computably enumerable degree could be

*Downey's research supported by Cornell University, an IGC grant from Victoria University and the New Zealand Marsden Fund via grant 95-VIC-MIS-0698 under contract VIC-509. Some of these results were obtained whilst Downey was a Visiting Professor at Cornell University in fall 1995.

decomposed into a pair of incomparible nonzero computably enumerable degrees. Moreover the technique introduced for the proof of this result was the key to the widespread use of the infinite injury method. We refer the reader to Downey and Stob [7] for a survey of results on splitting theorems in classical computability theory.

Because of the natural relationship between the splittings of a computably enumerable set and the $\mathcal{R}$ the upper-semilattice of computably enumerable degrees, it is natural to explore the possible structure of the following sets of computably enumerable degrees.

$$S(A) = \{\deg(A_1) : (\exists A_2)[A_1 \sqcup A_2 = A]\}, \text{ and,}$$

$$S_2(A) = \{\langle \deg(A_1), \deg(A_2)\rangle : A_1 \sqcup A_2 = A\}.$$

In this paper, our concerns are two questions implicit in Lerman and Remmel [11, 12] and Ambos-Spies and Fejer [1], about the the property known as the *universal splitting property.* By Sacks splitting theorem, we know that $S(A)$ and $S_2(A)$ have infinitely many elements. The largest possible sets $S(A)$ and $S_2(A)$ can be are $S(A) = \{\mathbf{b} : \mathbf{b} \leq \deg(A)\}$, and $S_2(A) = \{\langle \mathbf{b}_1, \mathbf{b}_2\rangle : \mathbf{b}_1 \cup \mathbf{b}_2 = \deg(A)\}$. Following Lerman and Remmel [11, 12], we shall say a set A has the *universal splitting property* if $S(A)$ is as large as possible: A has a splitting in each possible computably enumerable degree below $\deg(A)$, and following Ambos-Spies and Fejer we say at A has the *strong universal splitting property* if $S_2(A)$ is as large as possible.

There have been quite a number of results concerning the (strong) universal splitting property. For instance, Downey [3] proved that every nonzero computably enumerable degree degree contained a computably enumerable set without ithe universal splitting property, and indeed, in [4], proved that no hypersimple computably enumerable has the universal splitting property. (This is a kind of splitting analogue to Stob's result [16] that a computably enumerable set is simple iff it does not have computably enumerable supersets of each nonzero degree. The Downey hypersimple set result says that the position of a computable enumerable set in the lattice of computably enumerable sets similarly affects the possible degrees of splittings.) On the other hand, Lerman and Remmel [11, 12] constructed computable enumerable sets with the universal splitting property and later Ambos-Spies and Fejer improved this to the strong universal splitting property, and provided a partial characterization of the degrees with such sets.

The original construction in Lerman and Remmel [12] of a set with the universal splitting property resembled the construction of a contiguous degree in Ladner and Sasso [10]. (Recall that a degree is contiguous Turing degree is one containing a single computably enumerable wtt-degree.) The connection between the wtt-degree structure of a T-degree was futher noted by Ambos-Spies and Fejer who observed that if is A has the property that for all $B \leq_T A$, $B \leq_{wtt} A$ then $A \times \omega$ has the universal splitting property. (Following Downey and Jockusch [5], we call deg(A) wtt-topped with top A.) It is known that all incomplete *wtt*-topped degrees are low$_2$, and all contiguous degrees are low$_2$. Recently Downey and Lempp [6] demonstrated that the contiguous degrees are definable in $\mathcal{R}$. By the work of Ambos-Spies and Fejer, [1], Proposition 2.1, the Downey-Lemmp definition also proved that a computably enumerable degree is contiguous iff it contains a set with the strong universal splitting property. Naturally this also means that all sets with the strong universal splitting property are low$_2$.

The goal of this paper is to delineate the connections between the universal splitting property, the strong universal splitting property, jump classes, and the wtt-degree structure of computably enumerable degrees. Our theorems are the following.

Theorem 1.1 *If A is incomplete and has the universal splitting property, then A is low$_2$.*

Ambos-Spies and Fejer proved that a low set A has the universal splitting property implies that A is a wtt-top of a wtt-topped degree. We deomonstate that the Ambos-Spies and Fejer result cannot be extended to all sets, and hence in some sense Theorem 1.1 is optimal.

Theorem 1.2 *There are computably enumerable sets with the universal splitting property which are not $wtt-$tops.*

At this stage we do not know if Theorem 1.2 can be improved to say that there is a non *wtt*-topped *degree* containing a set with the universal splitting property.

Our notation is standard and follows Soare [15]. We remind the reader that all uses etc at stage s are bounded by s and are nondecreasing in both argument and stage number. The hat convention applies throughout.

2 Proof of Theorem 1.1

Recall that given a computably enumerable set E we can construct the *dump set* $D(E)$ uniformly from E as follows. (Downey [3]) Let $E = \cup_s E_s$ be a canonical enumeration of E. We assume that at most one element a_s enters $E_{s+1} - E_s$ for each stage s. Define $D(A) = \cup D_s$ as follows.

Stage 0. $D_0 = \emptyset$ and $d_{i,0} = i$ for all i. (Here $\{d_{e,s} : e \in \mathbb{N}\}$ will list the complement of D_s in order of magnitude.)

Stage $s+1$. $D_{s+1} = D_s$ if $E_{s+1} = E_s$, and otherwise if $d_{j,s} = a_s$, we let $D_{s+1} = D_s \cup \{d_{j,s}, ..., d_{j+s,s}\}$, putting $d_{i,s+1} = d_{i,s}$ if $i < j$ and $d_{i,s+1} = d_{i+s+1,s}$ if $i \geq j$.

Then it is not difficult to establish that $D(E)$ is hypersimple if E is non-computable, $D(E) \leq_{wtt} E$ and $D(E) \equiv_T E$. Naturally, there is a computable function f such that for all e, $W_{f(e)}$ is the dump set of W_e.

We will need the following Lemma.

Lemma 2.1 *Suppose that A and B are computably enumerable sets with $B \leq_T A$ and suppose that A has the universal splitting property. Then $D(B) \leq_{wtt} A$.*

Once we have Lemma 2.1, Theorem 1.1 follows because it will prove that if A has the universal splitting property then $L = \{e : W_e \leq_T A\}$ will be a Σ^0_3 set. ($e \in L$ iff $W_{f(e)} \leq_{wtt} A$. And $\{j : W_j \leq_{wtt} A\}$ is Σ^0_3.) But then A is low$_2$ since $\{e : W_e \leq_T A\}$ is Σ^A_3. (Jockusch [9].)

Proof of Lemma 2.1. Let B be a given dump set. Suppose that $B \leq_T A$ via a reduction $\Phi^A = B$, yet $B \not\leq_{wtt} A$. We construct a set $C \leq_T A$ via a reduction $\Delta^A = C$ to meet the requirements below.

$$R_{\Gamma,\Xi,W,V} : [\Gamma^C = W \wedge \Xi^W = C \wedge W \sqcup V = A] \rightarrow (\exists wtt - \text{reduction } \Lambda)(\Lambda^A = B).$$

Note that the construction of Λ depends upon $\langle \Gamma, \Xi, W, V \rangle$. Assuming that $B \not\leq_{wtt} A$, the argument is finite injury. Thus it will suffice to describe the action of a single requirement $R = R_{\Gamma,\Xi,W,V}$. (Action by the R module simply initializes all lower priority modules.)

We will assume that the reduction $\Phi^A = B$ is given so fast that every stage s is expansionary, so that the length of agreement exceeds s at every stage s. We need the following auxiliary functions.

$$\ell(s) = \ell_R(s) = \max\{x : \forall y < x[\Xi^W(y) = C(y) \wedge$$

$$(\forall q \leq \xi(y)(\Gamma^C(q) = W(q)) \wedge W \sqcup V(y) = A(y)]\}[s].$$

Naturally, $\ell(s)$ is the C-controllable length of agreement at stage s. Since we regard functionals as controlling the sets they compute. we will have that once $\ell(s) > z$ if we don't change $C_t - C_s$ on the use of this computation, then $W \upharpoonright \xi(z)$ and hence $W \upharpoonright z$ is fixed. That is, elements entering A after stage s must enter V.

We meet R via followers, of the form x_n. These are coding markers for coding "$n \in B$". They are defined via cycles.

The cycle for n.

Step 1. Assume that we have completed the cycle for $n-1$ and in particular defined $\lambda^A(j)[s]$ for $j \leq n-1$, and the stage is $\ell(s)$-expansionary with $\ell(s) > x_{n-1}$. Then we pick a fresh number x_n large. (This number is reset if any j-module acts for $j < n$ *before* we define $\Lambda^A(n)$.) Initialize all R' of lower priority than R. Define $\Delta^A = C(x_n) = 0[s]$ with a big use $\delta(x_n)[s]$ for which we will henceforth ensure that

$$\delta(x_n) > \varphi(x_n)[s']$$

for all $s' \geq s$.

Step 2. Wait till $\ell(t) > x_n$. If t occurs then initialize all R' of lower priority than R. Define $\Lambda^A(n) = (n)[t]$ with a big use $\lambda(n) > t$. R now asserts control of $C \upharpoonright t$. C is now fixed on this region unless some $j \leq n$-module acts via step 3 action below described for n.

Comment. *We use the following strategy. If some number $p \leq \lambda(n)$ enters A at stage $d > s$, then we make $\Lambda^A(n)$ undefined until the next $\ell(s)$ expansionary stage. This is okay since the reduction Λ is only predicated upon the hypotheses of R being fulfilled.*

Step 3. $j \leq n$ enters B at stage v. Let j be the least such. Since we have kept $\delta(x_j) > \varphi(x_j)[s']$ for all $s(j) \leq s' \leq v$, we know that A has permitted $\delta(x_j)[v-1]$

and hence we can put $x_j, ..., x_n$ into C at stage v. Note that

(*) since B is a dump set, $j, ..., n$ have all entered B by stage v.

(The reader should note that our speeding up of the various enumerations does not really affect this part of the dump property, since we can delay the definition of $\Lambda^A(n)$ until B's enumeration of some $j \leq n$ will cause n to enter too.) Our action has the following effect. Since we have frozen C on $\gamma(\xi(x_j))[s(j)]$, if $\Lambda^A(n)[v-1] \downarrow = 0$, then we must have had $\ell(v-1) > x_n$. Hence, if $A \upharpoonright \lambda(n) = A \upharpoonright \lambda(n)[v-1]$, in particular, since $\lambda(n) > \gamma(\xi(x_n))[v-1]$, (by preservation), it must be that $\Xi^W(x_n) \neq C(x_n)$. In this case, the R module can only act j further times and only for $j' < j$.

Finally, if we get a stage $u \geq v$ where $\Lambda^A(n) \downarrow$ it can only be that for all $j \leq n$ we have correctly computed $B(j)$ up to stage u. Of course the module would restart for $n+1$ at this stage.

If we assume that $B \not\leq_{wtt} A$, then R can thus only act finitely often, and this observation completes the proof of the Lemma. □

In passing we note the proof above has the following corollary.

Corollary 2.2 *Suppose that A and B are any computable enumerable sets with $B \leq_T A$ but $D(B) \not\leq_{wtt} A$. Then there is a computably enumerable set $C \leq_{wtt} B$ such that if $W \sqcup V = A$ is a splitting of A, then $W \not\equiv_T C$.*

3 Proof of Theorem 1.2

We need to construct a computable enumerable set A that is not a *wtt*-top, yet A has the universal splitting property. While technically the more difficult, this result is the least interesting of all the results. Rather than simply quoting the result (which is our inclination) we include a proof for completeness. However, in view of its marginal interest, we will only sketch the proof.

We meet the requirements below, building a set $Q \leq_T A$ via a reduction $\Xi^A = Q$.

$$R_{\Phi,B} : (\Phi^A = B \rightarrow (\exists W, V)[W \sqcup V = A \wedge B \equiv_T W].$$

$$P_\Lambda : \Lambda^A \neq Q.$$

Here, Λ denotes a partial wtt-reduction with computable use function λ.

We begin by discussing the satisfaction of a $P = P_\Lambda$ requirement in isolation. Let α be a node on the tree of strategies devoted to ${}_P\Lambda$. The basic module for P_Λ consists of the following steps.

Step 1. At an α stage s_1, pick a large follower x targeted for Q. Set $\xi(x)[s_1] = x$.

Step 2. Wait for a α stage $s_2 > s_1$ where the follower is *realized.* That is, wait till $\Lambda^A(x) = Q(x)[s_2]$.

Step 3. Put $\xi(x)[s_1]$ into A, and lift $\xi(x)[s_2]$ to some $s_3 > s_2$.

Step 4. Wait for *recovery* at α. That is, we see an α stage $s_4 > s_3$ where $\Lambda^A(x) = Q(x)[s_4]$. Freeze $A \upharpoonright \lambda(x)[s_4]$ to preserve the $\Lambda^A(x)\downarrow[s_4]$ computation.

Step 5. Put $\xi(x)[s_4]$ into $A[s_4+1]$ and x into $Q[s_4+1]$ meeting P_Λ by preservation and the fact that $\xi(x) > \lambda(x)[s_4]$. Hence $\Lambda^A(x)[s_4+1]$ is unchanged, but $Q(x)[s_4] = 0 \neq 1 = Q(x)[s_4+1]$.

The module above clearly succeeds and is really a finitary requirement despite the fact that we have written it in a complicated tree of strategies $\mathbf{0}''$ way. The machinery will be needed when we look at the interaction of the various requirements below.

But first we turn to the $R_{\Phi,B}$ requirements in isolation. On the tree of strategies, there will be a node β devoted to building reductions $\Theta^B = W$ and $\Omega^W = B$, based on the hypothesis that $\ell(s) \to \infty$, where

$$\ell(s) = \max\{x : \forall y \leq x(\Phi^A = B)[s]\}.$$

Let β be a node on the priority tree devoted to meeting $R_{\Phi,B}$. Of course, at β-expansionary stages it will be out responsibility to update the definitions of Θ and Ω. The usual requirements for axioms are here. If x enters B then we must change W below $\omega(x)[s]$ and similarly, if y enters W it is our responsibility to change B below $\theta(y)[s]$.

The $R_{\Phi,B}$ requirements have the potential to force us to add many numbers into A as we now see. Suppose that at some stage s_1 we happen to add some z to A which is relatively small. At the next β stage s_2 we see that B has changed upon a medium number z'. Because of this we will apparently need to put z into V (and not W) but need to put something into W to record z''s entry. Thus we can delay the definition of W and V's updates and first put $\omega(z')$ into A (noting that we can keep $\theta(x) > x$ for all x, so that $z' < \theta(\omega(z'))$ and hence we will be able to later update Θ too. Now this process can repeat itself at the next β stage. Thus β has the potential to put many relatively small numbers into A at relatively late stages, many stages removed from the initial enty of elements into A which was the first cause of Φ^A to potentially change B. However, note that since we will surely not extend the definitions of Θ and Ω this process will eventually catch up with itself, and we will add the relevant least $\theta(\widehat{z})$ to W. and put the rest into A.

Notice that we have not yet exploited the ability we have to move the uses of the reductions Θ and Ω. This is because the ideas above in isolation can be used to make a set of *contiguous degree*[1] with the strong universal splitting property (Downey [4]).

In our case we will need to explicitly kill many potential *wtt*-reductions, and hence as we see must actually explicitly use the full force of our potential to move the uses. This fact comes now when we consider the interactions with the other requirements below.

Interactions.

The simplest scenario is that of $\beta\hat{\ }\infty \subset \alpha$. In this case α has the power to initialize β. The action of α is finite down the true path so this initialization will only occur finitely often.

The more difficult scenario is the situation that we have

$$\beta\hat{\ }\infty \subseteq \alpha.$$

The problem is that we must now update Ξ's reductions at each stage we perform step 3, which is before we get recovery in step 4 for α and, more importantly, before we get β correctness or even know φ's new value.

To illustrate this problem scenario, suppose that we blindly pursued the basic module. Thus at a α stage s, we perform step 3, and put $\xi(x)[s]$ into A. Thus we redefine $\xi(x)[s']$ for $s' = s+1$. When we hit β may well need some additional

[1] If a degree is contiguous and contains a set with the universal splitting property, note that all the reductions will be *wtt* reductions.

coding. We could even move $\xi(x)[s]$ again, but it does not really aid us. $\xi(x)[s']$ already exceeds $\lambda(x)$. The trouble is that we don't yet have the final value of $\varphi(z)[s'']$ for the next $\beta\hat{}\infty$ stage s'' for various z with perhaps $\omega(z)[s'']$ still quite small. (Perhaps $\omega(z)[s]$ is unchanged since the only thing that happened was that $\varphi(z)[s]$ has been changing but $B \upharpoonright z$ has remained fixed throughout.)

Naturally the opponent will make sure that for such small z, it will be that $\varphi(z) > \xi(x)[s'']$.

Therefore at the next α stage u where we get to enact step 4, $\xi(x)[u] = \xi(x)[s'']$'s entry into A will allow a change in $B \upharpoonright z$. But then at the next $\beta\hat{}\infty$ stage u', we may well see z enter B, and hence we will be forced by β to put $\omega(z)[u]$ which is still very small into A. But then it might well be that in fact $\omega(z)[u] < \lambda(x)[u]$. Therefore our opponent will no doubt be able to cause an A change correcting $\lambda^A(x) = Q(x)[u'']$ at some stage $u'' \geq u$.

To overcome this problem, we will use the idea of a backup strategy as we see below. In the situation of a *single* β as above, with $\beta\hat{}\infty \supseteq \alpha$, we proceed as follows.

Step 1. First we will work as in the basic module. At an appropriate α stage s_1 we will appoint a follower $x = x(\alpha)$.

Step 2. As with the basic module we wait for a α stage s_2 where x becomes realized. ($\Lambda^A(x) = Q(x)[s_2]$)

Step 2.1 Now begin the backup strategy at α. Define a new set of followers $\{b_1, ..., b_{s_2}\}$ which are targeted for Q and are very large. Note that there are s_2 of them and hence there are more of them than there are $\omega(z)$'s on the board at present.

Step 2.2 As with the primary strategy, we wait for a α stage $s_{2.2}$ where $\{b_1, ..., b_{s_2}\}$ all become realized. ($\Lambda^A(b_i) = Q(b_i)[s_{2.2}]$, for all $i \in \{1, ..., s_2\}$.) Again we will suppose that stage $s_{2.2}$ has β-correct computations.

Step 3. As with the basic module, we put $\xi(x)[s_2]$ into $A[s_2 + 1]$, and redefine $\xi(x)[s_3]$ and $\xi(b_i)[s_3]$ for each i, to be very large where $s_3 = s_2 + 1$.

Step 3.5. When we hit β and the stage is β-expansionary, see if β desires us to put $\omega(z)[s_3]$ into A. Perform such β coding if necessary making sure that we do not extend the definitions of Θ and Ω until we reach a $\beta\hat{}\infty$-correct stage. At a

$\beta\widehat{\ }\infty$-stage $s_{3.5}$, we get to redefine any $\omega_\beta(z)$ which have moved but we redefine them to be *large*. (And in particular, larger than $\lambda(b_i)$ for all i.)

Step 4 As with the primary strategy, we wait for a α stage s_4 where $\{b_1, ..., b_{s_2}\}$ all become realized again. $(\Lambda^A(b_i) = Q(b_i)[s_4])$

Step 5. As with the basic module, put $\xi(x)[s_4]$ into $A[s_4+1]$ and x into $Q[s_4+1]$ potentially meeting P_Λ by attempting to preserve and the fact that

$$\Lambda^A(x)[s_4] = 0 = Q(x)[s_4] \neq 1 = Q(x)[s_4+1].$$

Let α preserve $A \upharpoonright s_4$ as best it can.

Comment. *Notice that the disagreement will be preservable unless the entry of $\xi(x)[s_4]$ into $A[s_4+1]$ (but below $\varphi_\beta(z)[s_{3.5}]$ for various z) causes the entry of some such z into $B - B[s_{3.5}]$ with $\omega(z) < \lambda(x)[s_{3.5}]$. It is important here that the reader realize that the* only *z which can cause the potential win to go away were those z already present at stage $s_{3.5}$. Any $\omega(z')[t]$ appointed at stages after $s_{3.5}$ will be larger than $\lambda(x)[s_{3.5}] = \lambda(x)[s_4]$.*

Step 6. At the next β stage s_6, attend to any coding that is needed. *However, we will do the following. For any $\omega(y)[s_6]$ which enters $A[s_6]$ (because at some later stage we commit to putting some $c \leq \omega(y)[s_6]$ into W), we make sure that that we declare $\omega(y')[s_6+1]\uparrow$ for all $y' \geq y$, until the next $\beta\widehat{\ }\infty$ stage which is β-corrrect and hence there are no pending codings round.*

Step 7. Finally we reach a $\beta\widehat{\ }\infty$ correct stage s_7. Now we must redefine any $\omega(y)[s_7]$ which were defined but have become undefined during the process above. Notice that the only way that this undefining has happend is because we have put a unique $c \leq \omega(y)[s_f]$ some $f < 7$ into $W \cap A[s_7]$. Thus it is legal for us to redefine the $\omega(y)[s_7]$ to exceed $\lambda(b_j)[s_7]$ for all $j = 1, ..., s_2$, and $\varphi(w)[s_7]$ for all $w \leq \ell_\beta(s_7)$.

Now we get to the key point of this whole process. There are two cases.

Case 7.1 No β-coding has occurred below $\lambda(x)[s_{3.5}]$, because B did not change on any z with $\omega(z)[s_{3.5}] < \lambda(x)[s_{3.5}]$.

Action. We need do nothing except freeze the present situation by letting α assert control of $A \upharpoonright s_7$. There are no pending codings around and nothing has entered $A \upharpoonright \delta(\lambda(x))[s_{3.5}]$. Therefore we really did create an α preservable disagreement in step 5. (This is the import of the subsequent comment.)

Case 7.2 The disagreement created at step 5 has been killed because of the coding of some $\omega(z)[s_{3.5}]$ below $\lambda(x)[s_{3.5}]$ into $A[s_7] - A[s_3.5]$. *The main point is this. Since such a coding has occurred, we know that for all $z' > z$, $\omega(z')[t]$ became undefined and has only been redefined in step 7. For all such z', we have $\omega(z')[s_7]$ to exceed $\lambda(b_j)[s_8]$ for all $j = 1, ..., s_2$, and $\varphi(w)[s_7]$ for all $w \leq \ell_\beta(s_7)$. Now as usual act to freeze this situation and begin the backup modules for* $1, ..., s_2$. *Note well that there are fewer than s_2 numbers y with $\omega(z')[s_7] < \lambda(b_j)[s_7]$ for any $j \in \{1, ..., s_2\}$ and there are none in the interval $[\lambda(x), \lambda(b_{s_2})]$.*

Backup Strategy. We perform the following cycles $i = 1, ..., s_4$. We only begin cycle $i + 1$ if cycle i fails in step i.5.

Cycle i

Step i.1 Wait till b_i becomes realized. (cf. Step 2) That is we see an α stage $s_{i.1}$ where $\Lambda^A(b_i) = Q(b_j)[s_{i.1}]$. Note that we already know that $\xi(b_i)[s_{i.1}]$ is larger than $\lambda(b_j)[s_{i.1}]$ for all $j \in \{1, ..., s_2\}$ by step 3. Put $\xi(b_i)[s_{i.1}]$ into A and b_i into Q. This again makes a potentially preservable disagreement

$$\Lambda^A(b_1) \neq Q(b_i)[s_{i.1} + 1].$$

Again it can only be injured by numbers z with $\omega(z)[s_{i.1}]$ below $\lambda(b_i)[s_{i.1}]$ We know that such z are below s_2.

Step i.2.5 Now repeat step 3.5 of the primary strategy. That is at the next β stage $s_{i.2.5}$, attend to any coding that is needed. For any $\omega(y)[s_{i.2}]$ which enters $A[s_{i.2.5}]$ (because at some later stage we commit to putting some $c \leq \omega(y)[s_{i.2}]$ into W), we make sure that that we declare $\omega(y')[s_{i.2} + 1] \uparrow$ for all $y' \geq y$, until the next $\beta\hat{}\infty$ stage which is β-corrrect and hence there are no pending codings round.

Step i.3 Now we reach the analogue of step 7 for b_i. That is, we reach a $\beta\hat{}\infty$-correct stage $s_{i.3}$. Now we must redefine any $\omega(y)[s_{i.3}]$ which were defined but have become undefined during the process above. Again, notice that the only way that this undefining has happened is because we have put a unique $c \leq \omega(y)[s_f]$ some $f < i.3$ into $W \cap A[s_{i.3}]$. Thus it is legal for us to redefine the $\omega(y)[s_{i.3}]$ to exceed $\lambda(b_j)[s_{i.3}]$ for all $j = 1, ..., s_2$, and $\varphi(w)[s_{i.3}]$ for all $w \leq \ell_\beta(s_{i.3})$.

Again, there are two cases.

Case i.3.1 No β-coding has occurred below $\lambda(x)[s_{i.1}]$, because B did not change on any z with $\omega(z)[s_{i.1}] < \lambda(b_i)[s_{i.1}]$.

Action. We need do nothing except α-freeze the present situation. There are no pending codings around and nothing has entered $A \upharpoonright \lambda(b_i)[s_{i.3.1}] = M \upharpoonright \lambda(b_i)[s_{i.1}]$. Therefore we really did create a preservable disagreement in step $i.1$. We don't begin cycle $i+1$ since we won in cycle i.

Case i.3.2 The disagreement created at step $i.1$ has been killed because of the coding of some $\omega(z)[s_{i.1}]$ below $\lambda(b_i)$ into $A[s_{i.3}] - A[s_{i.1}]$. Again, since such a coding has occurred, we know that for all $z' > z$, $\omega(z')[t]$ became undefined and has only been redefined in step $i.3$. For all such z', we have $\omega(z')[s_{i.3}]$ to exceed $\lambda(b_j)$ for all $j = 1, ..., s_2$, and $\varphi(w)[s_{i.3}]$ for all $w \leq \ell_\beta(s_{i.3})$. Begin cycle $i+1$.

The Punch Line. *By the hypothesis predicating the necessity of the backup strategy, cycle i, the z which entered B to cause this problem must have been below s_2, and in fact below the y which killed the b_{i-1} disagreement of cycle $i-1$. Therefore although we have lost cycle i another z below s_2 has been used up. All the other $\omega(z')[s_{i.3}]$'s are to large to cause coding injury in cycle $i+1$. Since there are only $s_2 - 1$ numbers below s_2, it follows that some b_i will succeed.*

More than one β.

The above finishes the process for α living below a *single* $\beta\widehat{\ }\infty$. The argument for more than one β is similar but even uglier. For simplicity, suppose that we have

$$\beta_1\widehat{\ }\infty \subseteq \beta_2\widehat{\ }\infty \subseteq \alpha.$$

Now each of the β_i can independently and out of phase decide to injure the attempt at diagonalization. It becomes a counting argument.

One can think of the strategy above as building a sequence $b_1, ..., b_{s_2}$ for the sake of β. Similarly, we would begin the first layer of backup strategies at an α stage building a sequence $b_1, ..., b_{s_2}$ as above at step 2.1 and wait for then to be realized at step 2.2. Suppose again that we get to this α stage at stage $s_{2.2}$.

To take care of the second β_i, we will interpolate new steps 2.2.1 and 2.2.2

Step 2.2.1 Pick a *new* set of followers $b'_1, ..., b'_{2s_{2.2}}$.

Step 2.2.2 Wait for a α stage t where all of x, $b_1, ..., b_{s_2}$ and $b'_1, ..., b'_{2s_{2.2}}$ are realized.

Now we perform step 3 as before and put $\xi(x)[t]$ into A kicking all of the

$\xi(b_j)[t+1]$ and $\xi(b'_j)[t+1]$ to be large. After we have finished all the β_1 and β_2 codings, say at stage t', either we will have an α preservable win as in Case 7.1, or some number $\omega_{\beta_i}(z)[t] < \lambda(x)$ has entered A. Now it does not matter if this is $i = 1$ or $i = 2$ but we certainly know that *there are only* $s_2 - 1$ *numbers of the form* $\omega(z)[t']$ *below any* $\lambda(b_j)$ *or* $\lambda(b'_k)$. Thus we would begin the backup strategy on $\{b_1, ..., b_{s_2}\}$. The point is that either that will win or it must be that $\beta_{i'}$ for $i' \neq i$, did not code between stages t and t' but decides to code while we are attempting the backup strategy. In that case, some number $\omega_{\beta_{i'}}(z')[t]$ below $\lambda(b_q)$ must be coded into A by $\beta_{i'}$. But now we will using the backup to the backup strategy, using the numbers $b'_1, ..., b'_{2s_{2.2}}$. *All of* both β_1 *and* β_2*'s* $\omega_{\beta_i}(z)$*'s have ben cleared from the interval* $[s_{2.2}-1, t]$*, and now we have more entropy than both of the opponents combined. (Each* β_i *has only* $\leq s_{2.2} - 1$ *numbers to work with and we have* $2s_{2.2}$*.)*

Naturally with more than 2 β_i one has as many layers of backup strategies as necessary. The argument now follows with no new insights but much detail. $\square$.

References

[1] K. Ambos-Spies and P. A. Fejer, Degree theoretical splitting properties of recursively enumerable sets, J. Symbolic Logic **53** (1988), 1110-1137.

[2] P. Cholak, *Automorphisms of the Lattice of Recursively Enumerable Sets,* Memoirs American Math. Soc., No. 541, 1995.

[3] R. G. Downey, The degrees of r.e. sets without the universal splitting property, Trans Amer. Math. Soc. 291 (1985) 337- 351

[4] R. G. Downey, Subsets of hypersimple sets, Pacific J. Math. 127 (1987) 299-319.

[5] R. G. Downey and C. G. Jockusch, Jr., T- degrees, jump classes and strong reducibilities, Trans. Amer. Math. Soc. 301 (1987) 103-136.

[6] R. G. Downey and S. Lempp, Contiguity and distributivity in the enumerable degrees, submitted.

[7] R. G. Downey and M. Stob, Splitting theorems in recursion theory, Ann. Pure Appl. Logic **65** (1993), 1-106.

[8] Leo Harrington and Robert Soare, "Post's program and incomplete recursively enumerable sets," Proceedings of the National Academy of Science, U. S. A., Vol. 88, 1991, 10242-10246.

[9] C. G. Jockusch Jr., Relationships between reducibilities, Trans. American Math. Soc., vol. 142 (1969), 229-237.

[10] R. E. Ladner and L. P. Sasso, Jr., The weak truth table degrees of recursively enumerable sets, Ann. Math. Logic **8** (1975), 429-448.

[11] M. Lerman and J. B. Remmel, The universal splitting property I, in D. van Dalen, D. Lascar, and T. Smiley, eds, *Logic Colloquium '80*, (North Holland, Amsterdam, 1982), 181-208.

[12] M. Lerman and J. B. Remmel, The universal splitting property, II, J. Symbolic Logic **49** (1984), 137-150.

[13] P. Odifreddi, *Classical Recursion Theory*, North-Holland, Amsterdam, 1990.

[14] G. E. Sacks, On the degrees less than $\mathbf{0}'$, Ann. Math. (2) **77** (1963), 211-231.

[15] R. I. Soare, *Recursively Enumerable Sets and Degrees*, Springer-Verlag, New York, 1987.

[16] M. Stob, Index sets and degrees of unsolvability, J. Symbolic Logic, **47** (1982), 445-471.