SPLITTING THEOREMS AND LOW DEGREES

ELLEN CHIH AND ROD G. DOWNEY

ABSTRACT. We prove that if A is a non-low c.e. set, then there exists non-computable sets $A_1 \sqcup A_2 = A$ splitting A with A_1 non-low.

1. INTRODUCTION

Some of the earliest theorems in computability theory are those concerning splittings of computably enumerable (c.e.) sets. We say that $A_1 \sqcup A_2 = A$ is a *splitting of* A if A_1, A_2 are c.e., disjoint and $A_1 \cup A_2 = A$. One of the earliest results is Friedberg's splitting theorem [13] which asserts that each noncomputable c.e. set has a *nontrivial* split, meaning that neither A_1 nor A_2 are computable.

One of the reasons that splitting theorems have interest is their interactions with the c.e. degrees. If $A_1 \sqcup A_2 = A$, then $\deg(A_1) \vee \deg(A_2) = \deg(A)$ holds in the Turing (and in fact weak truth table) degrees. Thus Sacks famous splitting theorem [18] asserts that for each noncomputable c.e. set A and noncomputable Δ_2^0 set B there is a splitting $A_1 \sqcup A_2 = A$ with $B \not\leq_T A_i$ for $i \in \{1, 2\}$. Thus there is no least c.e. degree and all c.e. degrees are join reducible.

For more on the extensive interactions of splittings of c.e. sets with the c.e. degrees and other topics in classical computability theory, we refer to the somewhat dated but extensive paper Downey-Stob [12].

The current paper is concerned with interactions of splitting theorems and the jump operator. As observed by Soare, the standard proof of Sacks' splitting theorem automatically give *low* c.e. sets A_1, A_2, meaning that $A_i' \equiv_T \emptyset'$.

A natural question is to ask whether this is tight in some sense. Can c.e. sets always be nontrivially be split into non-low c.e. sets? Of course this question is only meaningful if A is non-low.

A strong counterexample would be provided by showing that each c.e. set A was *mitotic* meaning that it has a splitting $A_1 \sqcup A_2 = A$ where $A_1 \equiv_T A_2 \equiv_T A$. Unfortunately Ladner [16] proved that not every c.e. set was mitotic, but also proved [17] that there were nonzero c.e. degrees containing only mitotic sets. For more on this see Downey and Slaman [11], and Griffiths [14].

Perhaps a weak example could be provided by only looking at jumps. However, Ingrassia and Lempp [15] proved that there are c.e. sets A such that there is *no* splitting $A_1 \sqcup A_2 = A$ with $A_1' \equiv_T A_2' \equiv_T A'$.

Towards the other direction, Ambos-Spies [1] proved that there is a complete c.e. set A such that if $A_1 \sqcup A_2 = A$ is a splitting, then at *least one* of A_1 or A_2 is low. Can we have that both sides of every nontrivial splitting of a complete set always low? That is, the only splits with both sides noncomputable are essentially Sacks' splits? Downey and Stob [12] showed that the answer is no since if A is prompt

---

(i.e. of promptly simple degree) then there is a splitting $A_1 \sqcup A_2 = A$ with with $A_2 \not\equiv_T \emptyset$ and $A_1 \equiv_T A$.

Finally Downey and Shore [10] proved the following

Theorem 1.1 (Downey and Shore [10])**.** *There is a high c.e. set A (i.e. $A' \equiv_T \emptyset''$) such that if $A_1 \sqcup A_2 = A$ is a nontrivial splitting (i.e. neither A_1 nor A_2 are computable) of A then both A_1 and A_2 are* low$_2$*. (i.e. $A'' \equiv_T \emptyset''$)*

In terms of the jump operator, we show that Theorem 1.1 settles the matter since we prove the following definitive result.

Theorem 1.2. *If A is any non-low c.e. set, then there exists a non-low c.e. set A_1 and a non-computable c.e. set A_2 with $A_1 \sqcup A_2 = A$.*

The question we answer with Theorem 1.2 has attracted several conjectures. Theorem 1.2 certainly conjectured to be true around 1990, but no proof appeared; and in recent years conjectured to be not true.

Our proof is somewhat subtle (and surprisingly difficult) in that we will *essentially* use that fact that we can begin with A non-prompt. To the authors' knowledge, this is the first proof that nonuniformly splits into two cases, depending on whether A is prompt or tardy. Technically, we will use the tardy assumption to force certain "gaps" to remain open that we can argue that certain small numbers don't enter sets capriciously. This is done by trying to build noncomputable sets Q_α at nodes α on the true path, with the Q_α below a minimal pair. This must fail on some witness since they are a minimal pair, and it is here that we meet the requirement. The technique should be useful elsewhere.

We will work up to the proof by first establishing the result with the assumption that A is high. Once this is done we modify the proof for the non-low case, which involves significant surgery on the tree from the high case for purely technical reasons. Certain fundamental ideas from the high case remain, and there are most easily understood without the additional nonuniformity present in the non-high case.

We remark that Downey and Ng [9] have shown that if we consider splitting with low replaced by *superlow*[1] then the splitting result fails more spectacularly, even for *degrees*. Downey and Ng showed that there is a c.e. degree $\mathbf{a}$ such that if $\mathbf{a_1} \cup \mathbf{a_2} = \mathbf{a}$ in the c.e. degrees, then one of $\mathbf{a_1}$ or $\mathbf{a_2}$ is not *totally ω-c.a.*. Being totally ω-c.a. is a weakening of being superlow (and array computable) and means that if $f \leq_T \mathbf{a_i}$ then f has a computable approximation $f(n) = \lim_s f(n, s)$ with a computable function h, such that $|\{s \mid f(n, s+1) \neq f(n, s)\}| \leq h(n)$. Downey and Ng also showed that every high c.e. degree is the join of two totally ω-c.a. c.e. degrees. This extends a classical theorem of Bickford and Mills [4] who showed that $\mathbf{0}'$ is the join of two superlow c.e. degrees. However, in [9] it is shown that there are (super-)high c.e. degrees that are not the joins of two superlow degrees. Finally, Ambos-Spies, Downey and Monath have proven

Theorem 1.3 (Ambos-Spies, Downey, and Monath [2])**.** *Every c.e. set can be split into a pair of c.e. sets which are totally ω^3-c.a..*

Here $\mathbf{a}$ is totally ω^n-c.a. means that every $f \leq_T \mathbf{a}$ has an ω^n-approximation, given by effective Cantor Normal Form. (See Downey and Greenberg [7].) It is unknown if Theorem 1.3 can be improved to ω^2, even for *degrees*.

---

[1]That is, $A_i' \equiv_{tt} \emptyset'$.

We refer to Soare [19] or the computability section of Downey-Hirschfeldt [6] as a general reference to our notation and terminology. We tend to use the Lachlan convention that appending $[s]$ to a parameter, indicates its state at stage s. We denote the jump function $J^A(e) = \{e \mid \Phi_e^A(e) \downarrow\}$. The use of the jump function will be denoted by $j(e)$, and more generally, uses will be lower case letters of the functionals. Parameters don't change from stage to stage unless indicated otherwise.

2. The evolution of the present paper

Ellen Chih established a sketch of a proof of the result proven in this paper, but no details have appeared, or are likely to, for various reasons.

Independently, and thinking Chih had claimed the opposite direction, the second author also worked on the problem. Both authors independently split the proof into the non-prompt and prompt case, which seems essential. In an e-mail to Downey, Leo Harrington described Chih's proof which otherwise worked quite differently, and involved a lemma about lowness of independent interest. We give a proof of this Lemma in section 5. Chih's proof involved yet another level of nonuniformity, than the proof here. Our proof works through the Downey, Jockusch, Schupp non-lowness permitting machinery ([8]).

The point of all of this is that all errors in the present paper are due entirely to Downey. Given the long history of the problem, at least since 1990, it is important to get a proof in the literature.

3. The high case

We are given a c.e. set A of high degree and need to construct disjoint c.e. sets X and Y so that $X \sqcup Y = A$ and meet the requirements

$$R_e : J^X \neq \lim_s \varphi_e(\cdot, s).$$

$$P_e : \overline{Y} \neq W_e.$$

3.1. **Using Highness.** We will begin by proving the result for the more familiar case that A is of high degree. This proof has most of the elements we need and we later modify the proof to the situation of A only assumed non-low, in Section 4.

Since A is high we are given an A-computable function Γ^A such that for any computable function h,

$$(a.a.z)\Gamma^A(z) > h(z).$$

Of course, we will have a Δ_2^0 approximation $\Gamma^A(z)[s]$ with use function $\gamma(z)[s]$. This is sped up so that at every stage s, $\Gamma^A(p) \downarrow [s]$ for $p < s$.

For each requirement R or P we will build computable functions h and use them to force elements into A. Namely, if a function h is associated with a requirement and for its sake we wish to enumerate a element into A to put into one of X or Y, we will arrange things so that the relevant element needed is bigger than $\gamma(z)[s]$, and we have not as yet defined $h(z)$. (In fact it will be that this is the least such z with $h(z)$ undefined.) Note that $\gamma(z)[s]$ can only change if some element $a \leq \gamma(z)[s]$ enters $A - A_s$. Now, we will define $h(z) > \Gamma^A(z)[s]$.

The opponent only have two options. Either it gives up on z and admits that $\Gamma^A(z) < h(z)$, or it can change its stage s value to make $\Gamma^A(z) > h(z)$. In the latter case we are supplied with a permission.

If the requirement is infinitely active, then we can ensure that h is computable. Hence option 1 can only occur for finitely many z, as Γ^A is dominant. Option 2, will therefore occur almost always and hence almost always we get permissions from A.

Remark 3.1. We remark that with this methodology, as will also happen in the case that A is only non-low and not necessarily high, that although almost always A will permit, the opponent won't let this happen *promptly*. Naturally, while we await a permission, some new strategy is begun with the assumption that this z does not permit. For a single requirement this is of no consequence, but as we later see, the lack or promptness is the *essence* of difficulties we have when we consider interactions of requirements.

3.2. **The Basic R_e-Module.** Meeting R_e works as follows. We will pick an index $i = i_0$, potentially computable function h, and start argument 0 for h.

(i) We will keep $J^X(i_0) \uparrow [s]$ until a stage s is found where $\varphi_e(i_0, s) = 0$.

(ii) At this stage, we will declare that $J^X(i_0) \downarrow [s+1]$ with use $j(i_0)[s+1] > \gamma(p)[s+1]$. where p is the first argument for which we have not yet defined $h(p)$.

(iii) For stages $t \geq s+1$, until we see $\varphi_e(i_0, s) = 1$, we keep $j(i_0)[t] > \gamma(p)[t]$. To do this, whenever some number $q \leq \gamma(p)[t]$ enters $A_{t+1} - A_t$ we will put *at least one* such number into $X_{t+1} - X_t$. Notice for a fixed argument p of Γ^A, this will only effect a finite number of numbers entering A, essentially bounded by $\gamma(p)$.

(iv) Should we see a stage $u > s$ where $\varphi_e(i_0, u) = 1$, then we will try to make $J^X(i_0) \uparrow [u']$ for some $u' > u$.
The way we will implement this idea is to now define $h(p) > \gamma(p)[u]$.

(v) The strategy for i_0 now stalls until we see a stage $u' > u$ (if any) where $A \upharpoonright \gamma(p)[u']$ changes. If we see such a stage u' we would return to step (i). i_0 would now assert control of the strategy and we would not do further work until (ii) is again found for i_0.

(vi) While the i_0 strategy is stalled in (v), we'd begin the i_1-strategy for a newly chosen argument i_1, and whilst i_0 is stalled we would run through the same steps for i_1, which would assert control of the first unused argument for h. In general, the i_k strategy is initialized each time an i_{k-1} strategy is no longer stalled. We always ensure that $j(i_k)[s] > j(i_{k-1})[s]$.

There are several outcomes to the module above. The module could get stuck in (i) for some fixed i_k. This is outcome w which satisfies the requirement since $J^X(i_k) \uparrow \neq \lim_s \varphi_e(i_k, s) = 1$, the Σ_2^0-outcome. Otherwise we see that the R_e module is infinitely often active, and hence h is total. Because Γ^A is dominant, only finitely many i_k can be permanently stuck in (v), and hence there is a least k with the outcome (k, ∞), saying that $i_k[s]$ has come to a limit, and the whole cycle repeats infinitely often for this least i_k. We would conclude that $\lim_s \varphi_e(i_k, s)$ fails to exist. We would organize the outcomes as $(0, \infty) <_L (1, \infty) <_L \cdots <_L w$, an $\omega + 1$ list on the tree.

Remark 3.2. In the case of outcome w, the R_e-module has finite effect on the construction. In the case of outcome (k, ∞), i_d for $d < k$ never again active, from some stage s' onwards, but thereafter there will be infinitely many stages where $J^X(i_k) \uparrow [s]$. At such stages s, the R_e-module does not care if a number enters A,

in the sense that we would not be forced by (iii) or (v) to put such a number into X.

3.3. The Basic P_f-Module. This is essentially Friedberg Strategy.

    (i) Pick some d_0.

    (ii) We see that $Y \restriction d_0 = W_f \restriction d_0[s]$. Declare d_0 as *active*. We would begin a strategy on $d_1 > d_0$

    (iii) We see some number $a < d_p$ for some active p enter A. In this case we put a into Y and win P_f.

This strategy has two outcomes $s <_L w$. (More thematically, we could use $1 + \omega^* + 1$, where the left 1 represents s and the order type ω^* the number of times we play (ii). Using this, each time we play (ii) we can initialize on the priority tree to the right.) The second means that we get stuck waiting for (ii) on some d_k, the first means we invoke (iii).

This module is *finitary*. The strategy wins because A is noncomputable, and hence some a must so enter.

3.4. The Conflicts and Their Resolution. The first thing to note is that there are *no* conflicts between R_e strategies. They want to put numbers into X. So we concentrate on P_f vs R_e. A P_f above a R_e is finitary and this simply works by initialization (by moving left on the tree).

So we will concentrate upon P_f below an R_e. As we said in Remark 3.1, the conflicts between the strategies above all revolve around the fact that it is the opponent who chooses *when* to enumerate elements into A to witness the domination of Γ^A. Consider a version of P_f guessing the outcome of R_e is (k, ∞). It would *like* to see it's witnessing element a enter A at a stage where $J^X(i_k) \uparrow [s]$. That is because at such a stage s, an element entering A would be free to enter Y to meet P_f.

Certainly P_f could begin it's attack as per (ii) in R_e's cycle at a stage where $J^X(i_k) \uparrow [s]$. But there is no *a priori* reason that such an element would enter while R_e is not in (iii) or (v) for R_e, when $J^X(i_k) \downarrow [t]$. If such an a entered A_t when R_e is in that part of its cycle, we'd be forced to put a into X, otherwise the strategy for R_e would be invalidated. We would no longer have the ability to correct $J^X(i_k)$.

This conflict seems to be nearly fatal, and can be turned around to give a proof that there is no *uniform* method of constructing the split $X \sqcup Y = A$.

The method of solving the problem we use is rather unusual. We know that there is no problem is A is prompt by Downey and Stob [12], as mentioned earlier.

We desire that no small number like $a \leq d_p$ enters A while $J^X(i_k) \downarrow$. The plan is to try to impose restraint on A for stages while this is true. Of course, A is played by the opponent, so any restraint must be indirect. This restraint is imposed by utilizing the fact that since A is not prompt, it must be cappable. That is, if A is not prompt then there is a noncomputable c.e. set B such that $\deg_T(A) \wedge \deg_T(B) = \mathbf{0}$, by Ambos-Spies, Jockusch, Shore and Soare [3].

The idea is the following. We will build a c.e. set $Q \leq_T A, B$ and try to meet the requirements

$$M_m : Q \neq \overline{W_m}.$$

We will amalgamate these requirements with the outcomes of R_e, (k, ∞) met before. The outcomes will now read as of the form (k, m, ∞) of R_e. These are

ordered by regarding k, m as a pairing function, hence give a list, again of order type ω, below R_e.

Now each of these subrequirements of R_e will have associated with it a follower $i_{k,m}[s]$ (for the jump function $J^X(i_{k,m})$) serving the role of i_k discussed above. The rules of the basic module apply in that if something to the left acts then this witness is initialized.

The driver is that we don't want random small numbers entering A which should be usable for some P_f requirements being forced into X because $J^X(n)\downarrow$, and either we want to change the value of this ((v) of the basic module) or maintain the value ((iii) of the basic module). The plan is to use the fact that *for some m we must fail to meet M_m, since A and B form a minimal pair.*

The (k,m)-strategy for R_e.

The (k,m) strategy works by generating a stream of numbers $\{q_j^m : j \in \omega\}$ with (dropping the superscript) $q_0 < q_1 < \dots$. We will say that q_i is *realized* when we see $q_i \in M_m[s]$. q_{i+1} is defined as large and picked after q_i becomes realized. q_i is said to be *B-permitted* at stage s if it is realized and $B_s \upharpoonright q_i \neq B_{s-1} \upharpoonright q_i$.

We modify the basic module as follows. We do this while (k,m) is not yet *diagonalized*.

(i) We will keep $J^X(i_{k,m}) \uparrow [s]$ until a stage s is found where $\varphi_e(i_{k,m}, s) = 0$, and some q_i (the largest such) has been permitted since the last stage we visited the node representing R_e.

(ii) At this stage, we will declare that $J^X(i_{k,m}) \downarrow [s+1]$ with use $j(i_{k,m})[s+1] > \gamma(p)[s+1]$, where p is the first argument for which we have not yet defined $h(p)$.

(iii) For stages $t \geq s+1$, until we see $\varphi_e(i_{k,m}, s) = 1$, we keep $j(i_{k,m})[t] > \gamma(p)[t]$. To do this, whenever some number $n \leq \gamma(p)[t]$ enters $A_{t+1} - A_t$ we will put *at least one* such number into $X_{t+1} - X_t$. Notice for argument p, this will only affect a finite number of numbers entering A.

(iv) Should we see a stage $u > s$ where $\varphi_e(i_{k,m}, u) = 1$, then we will try to make $J^X(i_{k,m}) \uparrow [u']$ for some $u' > u$.
The way we will implement this idea is to now define $h(p) > \gamma(p)[u]$.
The following is *new*.

(ivb) While this is happening, (i.e. since we began (iii)) should some number $\leq \gamma(q_i)[s]$ enter A_t, then we will put q_i into Q_t and declare that all subrequirements of R_e of the form (k', m) for some k' are *diagonalized*. That would finish the stage, and $i_{k,m}$ would be discarded.

(v) The strategy for $i_{k,m}$ now stalls until we see a stage $u' > u$ (if any) where $A \upharpoonright \gamma(p)[u']$ changes. If we see such a stage u' we would return to step (i), with $J^X(i_{k,m}) \uparrow$. We would declare that q_i is no longer permitted.
As in the Basic Module,

(vi) While the $i_{k,m}$ strategy is stalled in (v), we'd begin the $i_{k',m'}$-strategy for the next not yet diagonalized (k', m) with a newly chosen argument $i_{k',m'}$. In general, the i_k strategy is initialized each time an i_{k-1} strategy is no longer stalled. We always ensure that $j(i_k)[s] > j(i_{k-1})[s]$.

So why does this work? First note that if we get stuck in (i) or (iii) waiting for a flip in the value of $\varphi_e(i_{k,m}, s)$ then this is a global win for R_e, and the outcome will be w. (Note, however, we could get stuck in (iv).)

Since there are infinitely many permitted q_i (as B is noncomputable), Thus if the φ_e behaves honestly, we will infinitely often play outcomes of the form (k, m, ∞). We first claim that there is a least such (k, m).

Suppose not. We can move to a stage s_0 where we are never again left of (k, m, ∞), and hence thereafter $i_{k,m}[s_0] = i_{k,m}$ is immortal. The stream of q_i associated with (k, m, ∞) is growing with time and hence there are infinitely many stages where (k, m, ∞) asserts control of the R_e-module. We cannot get stuck in (i) or (iii) by assumption, and hence the only way this can only act finitely often is either by being diagonalized (so that $Q \cap W_m \neq \emptyset$), or by being stuck in (v).

Lemma 3.3. *Assuming that we are dealing with an infinitely active version of R_e on the true path, $Q \leq_T B, A$.*

Proof. For the same reason as the basic module, only finitely many (k, m) modules can be stuck in (v), as Γ^A is dominant. If we assume that the node representing R_e is on the true path, then for any q_i^x, once it is B-permitted, either that permission will be initialized because of a higher priority $(r, z) < (w, x)$ acting at the next R_e-stage, or we will begin an attack for the outcome (w, x, ∞) associated with that q_i^x. Now either that attack is initialized, or for almost all q_i^x's the attack reaches (v), or a diagonalization occurs using q_i^x via a $\gamma(q_i^x)$-permission. Therefore $Q \leq_T B, A$. $\square$

It follows that for some least (k, m), (k, m, ∞) is on the true path and $W_m = \overline{Q}$, no q_i^m is ever enumerated into Q. Moreover, if we call this the *true* (k, m), then the following holds.

Lemma 3.4. *For the true (k, m), every $i_{k,m}$ cycle runs from (i) to (v). Suppose that a cycle is begun following a permission on q_i at stage s_0 and reaches (v) at stage s_1. Then no number enters A below $\gamma(q_i)[s]$ for any stage $s \in [s_0, s_1]$.*

So now we see how we can make no small numbers enter A whilst R_e would put them into X. *During a cycle*, Lemma 3.4 says no small numbers enter A during a cycle. *Between cycles*, $\Gamma^X(i_{k,m}) \uparrow$ so outcome (k, m, ∞) does not care about numbers entering A.

The P_f strategy remains the same. It does not care about B, only the noncomputabilty of A. The point is that some R_e strategy at left of a node τ representing P_f could ask that a number entering A go into X for the sake of (iii). Since Γ^A is total this can only happen finitely often and we can afford to initialize each time this request happens. Thus, in the case of A high, there is only finite injury from the left. In the later non-low case things are more complex.

The remainder of the argument is to put this on a tree in a standard way; and this is routine.

4. The case that A is only non-low

.

Being non-low resembles being "nonuniformly locally high". This has been made precise in, for example, Downey, Jockusch and Schupp [8] where the methodology is used to show that non-low c.e. degrees contain c.e. sets of density 1 with no computable subset of density 1.

The idea is to try to build a proof that A is low, and we see that for some argument k this proof will fail. It will be here that we succeed.

In more detail, for the *basic module* first, we again split R_e into infinitely many subrequirements $R_{e,k}$. This time subrequirement k is attempting to give a limit lemma definition of $\emptyset'(k)$, $\lim_s \Theta(k, s)$.

Initially, $\Theta(k, t) = 0$ indicating that we believe that $\Phi_k^A(k) \uparrow$. At some stage s we see $\Phi_k^A(k)[s] \downarrow$. The question is, *should we change* $\Theta(k, s) = 1$? What we will do in synchronise this belief with attempts to meet R_e. We will have a current $i_k[s]$ devoted to this subrequirement, as in the Basic Module of Subsection 3.2.

What we will do is now is to invoke the basic module; a modification of the one from the previous section. Currently we are keeping $\Theta(k, u) = 0$, until we hit (v) below. *If, during the execution of the below, before we reach (iii) we see* $\Phi_k^A(k) \uparrow [u]$ *then we abandon this attack. Note we are not yet initializing anything right of* (k, ∞).

  (i) We will keep $J^X(i_k) \uparrow [s]$ until a stage s is found where $\varphi_e(i_k, s) = 0$.
  (ii) At this stage, we will declare that $J^X(i_k) \downarrow [s+1]$ with use $j(i_p)[s+1] > p$.
       Here p is some number chosen with $p > \phi_k(k)[s]$ (which denotes $\phi_k^A(k)[s]$).
 (iii) For stages $t \geq s+1$, until we see $\varphi_e(i_k, s) = 1$, we will monitor the situation, and if some number $\leq \phi_k(k)[s]$ enters A_t we declare $J^X(i_k) \uparrow [t]$ by putting said number into X.
  (iv) Should we see a stage $u > s$ where $\varphi_e(i_k, u) = 1$, then we will try to make $J^X(i_0) \uparrow [u']$ for some $u' > u$.
       New: The new way we do this is that we will finally define $\Theta(k, u') = 1$ (with big use, in particular $> \phi_k(k)[u]$), for $u' > u$ unless (v) below occurs.
   (v) The strategy for i_0 now stalls until we see a stage $u' > u$ (if any) where $A \restriction \phi_k(k)[u']$ changes. If we see such a stage u' we would return to step (i). i_0 would now assert control of the strategy and we would not do further work until (ii) is again found for i_0. Also making $\Theta(k, u'') = 0$, for $u'' \geq u'$ unless otherwise specified.
  (vi) While the i_0 strategy is stalled in (v), we'd begin the i_1-strategy for a newly chosen argument i_1, and whilst i_0 is stalled we would run through the same steps for i_1. It is here that initialization occurs. In general, the i_k strategy is initialized each time an i_{k-1} strategy is no longer stalled. We always ensure that $j(i_k)[s] > j(i_{k-1})[s]$.

Now the argument that the new R_e-module succeeds is similar. Suppose that for all k, the k-module fails, and the Σ_2^0 outcome is not on the true path. Then each k module can only reach (v) finitely often. We claim that A is low, and $\Theta(k, \cdot)$ is correct. The only time we change the value of $\Theta(k, s)$ is when we reach (v). This will be correct, unless a stage u' as above occurs, and then the value will be changed back to 0. The fact that each k-module only acts finitely often means that $\lim_s \Theta(k, s)$ exists for all s. The fact that w is not correct, means that each cycle of the basic module is either completed or the $\Phi_k^A(k) \downarrow [s]$ computation goes away before the cycle if completed. The fact that we only initialize to the right at step (v), means that no false initiation of k-modules can upset our actions.

Therefore, some k module must succeed. If the Σ_2^0-outcome is not correct, then this means that there is some least k which reaches (v) infinitely often and hence $\lim_s \varphi_e(i_k, s)$ has no limit, and cannot compute $J^X(i_k)$ where $i_k = \lim_s i_k[s]$.

Finally adding the B-permitting is *exactly* as before with the changes according to the module above.

So here is how the construction looks at a node ρ devoted to R_e. For undiagonalized k, m, if we hit ρ, then we will either play the outcome w or be in some attack as described below.

(a) If, during the execution of the below, before we reach (iii), we see $\Phi_k^A(k) \uparrow [u]$ then we abandon this attack at *any* stage u between the stage where the attack was begun and we hit ρ this time. Note we are not yet initializing anything right of (k, m, ∞). If this occurs, q_i is no longer permitted.

(b) We do not allow a k, m strategy to initialize outcome to the right unless we hit (iv) where we define $\Theta(k, s)$. i.e. q_i is no longer permitted.

A consequence of these two outcomes is that there may be many k', m'-strategies played in parallel, because of the capriciousness of the $\Phi_{k'}^A(k')$. This is of no consequence, because we do not play nodes extending $\rho\widehat{\ }(k, m, \infty)$ until we play (v) in the module.

(i) We will keep $J^X(i_{k,m}) \uparrow [s]$ until a stage s is found where $\varphi_e(i_{k,m}, s) = 0$, $\Phi_k^A(k) \downarrow [s]$, $\Theta(k, s) = 0$, and some q_i (the largest such) has been permitted since the last stage we visited ρ. (Note that the requirement $\Theta(k, s) = 0$ means that at most one k, m has defined $\Theta(k, s) = 1$ at a stage.)

(ii) At this stage, we will declare that $J^X(i_{k,m}) \downarrow [s+1]$ with use $j(i_{k,m})[s+1] > p$. Here p is some number chosen with $p > \phi_k(k)[s]$.

(iii) For any stages $t \geq s + 1$, until this module is resolved, until we see $\varphi_e(i_{k,m}, s) = 1$, we will monitor the situation, and if some number $\leq \phi_k(k)[s]$ enters A_t we declare $J^X(i_k) \uparrow [t]$ by putting said number into X.

(iv) Should we see a stage $u > s$ where $\varphi_e(i_{k,m}, u) = 1$, then we will try to make $J^X(i_{k,m}) \uparrow [u']$ for some $u' > u$. (i.e. using Θ.)

(iva) While this is happening, (i.e. since we began (iii)) should some number q_i enter A_t, then we will put q_i into Q_t and declare that all subrequirements of R_e of the form (k', m) for some k' are *diagonalized*. That would finish the stage, and $i_{k,m}$ would be discarded.

(ivb) Else we will finally define $\Theta(k, u') = 1$, for $u' > u$ unless (v) below occurs. It is at this stage that we would initialize (k', m') right of (k, m)

(v) The strategy for $i_{k,m}$ now stalls until we see a stage $u' > u$ (if any) where $A \upharpoonright \phi_k(k)[u']$ changes. If we see such a stage u' we would return to step (i), with $J^X(i_{k,m}) \uparrow$, putting the number entering A into X. We would decare that q_i is no longer permitted.

When (v) occurs it is now that we would allow the construction to play strategies at nodes ν with $\rho\widehat{\ }(k, m, \infty) \preceq \nu$.

(vi) While the $i_{k,m}$ strategy is stalled in (v), we'd begin the $i_{k',m'}$-strategy for the next not yet diagonalized (k', m') with a newly chosen argument $i_{k',m'}$. In general, the i_k strategy is initialized each time an i_{k-1} strategy is no longer stalled. We always ensure that $j(i_k)[s] > j(i_{k-1})[s]$.

Now, in the construction, as opposed to the basic module, it is (iii) that causes a problem. It could be that some $(k', m') < (k, m)$ is working with a fake $\Phi_{k'}^A(k')$ and during its cycle it sees a number entering A from the right and this number is below $\varphi_{k'}^A(k')[s]$, forcing this number into X, to correct $J^X(i_{k',m'})$. This can only occur finitely in the high case as $\gamma(k')[s]$ has a limit. But in the non-high case, it is conceivable it might happen infinitely often. Because of this, the P_f modules below need to know if this happens. Thus for each outcome of the form $\rho\widehat{\ }(k, m, \infty)$ we

will have a second layer of outcomes guessing this Π_2^0 behaviour. To wit, we would have finite sets F_n coding the $(k', m') <_L (k, m)$ for which this fake behaviour, and it would be enough to order them numerically.

The plan would be that if a P_f is guessing that $(k', m') \in F_n$ is acting in this way infinitely often, it would demand that its $d_i < q_{i'}$ where $q_{i'}$ is the permitted q_j associated with $\phi_{k'}(k')[s]$. This means that *if some small a from this version is pulled by $\rho\widehat{\ }(k', m', \infty)$, then it will be so* small *that $q_{i'}$ will succeed in diagonalizing* $M_{m'}$. Hence, this set-up allows for infinitely many numbes to enter A left of the true path for (k', m'), but conclude that they are so large that they are of no consequence for (k, m, ∞).

Being Π_2^0 behaviour, we can play this each time this event fires as the outcome of $\rho\widehat{\ }(k, m, \infty)$ by reaching (v).

Next suppose that τ is a P_f-node. The strategy is similar, but takes the comments above into account.

First if τ is infinitely often accessible, then it will be allowed to cycle through its definitions of d_i as desired. If it was infinitely often active, as A is noncomputable, this will produce infinitely many active a's, and by construction, these are *small*. On the assumption that all the nodes $\rho\widehat{\ }(k', m', \infty)$ above are on the true path, a cannot enter A at a stage where $J^X(i_{k',m'}) \downarrow [s]$ lest m' become diagonalized. Note that if this number a is forced into X, it can only be because of some strategy left of τ. In particular, it is some strategy of the form $\rho\widehat{\ }(p, m, \infty)$, with $\Phi_p^A(p) \downarrow [s]$, and $a < \varphi_p(p)[s]$. If we are dealing with the leftmost version of τ on the true path we would have $a < q_i^{p,m}$, since if this happens infinitely often, then $q_i^{p,m} \to \infty$ and $\varphi_p(p)[s] \to \infty$. But then we would put $q_i^{p,m}$ into M_m, diagonalizing M_m henceforth. Therefore, only finitely many such a can be pulled into X for version of P_f at τ, assuming it is on the true path. Thus after some stage, they enter when $J^X(i_{k',m'}) \uparrow$ and can successfully be put into Y and not X.

The result now follows.

5. Characterization of lowness

The proof sketch communicated to Downey by Harrington used a different strategy. This strategy involved a new characterization of lowness that is worth recording, and we do so here.

Lemma 5.1 (Chih-Harrington). *A computably enumerable set A is low iff there is a computable order (i.e. increasing computable function) such that for all e,*

$$\exists^\infty s \forall t \in [s, f(s)](\Phi_e^A(e)[s] \downarrow) \implies \Phi_e^A(e) \downarrow .$$

(Indeed, this will also hold for all orders f^* growing faster than f.)

Proof. First suppose that the hypothesis holds. Define $\Theta(e, s) = 1$ iff $\Phi_e(e) \downarrow [s]$ and $\forall t \in [s, f(s)](\Phi_e^A(e)[s] \downarrow$. Define $\Theta(e, s) = 0$ otherwise. Then by hypothesis and the Limit Lemma $J^A(e)$ is determined by $\lim_s \Theta(e, s)$, and hence $J^A \leq_T \emptyset'$.

Conversely, suppose that A is low. We define a computable function f. Since A is low we know that $X = \{j \mid (\exists n \in W_j)[D_n \subseteq \overline{A}\} \leq_T \emptyset'$. (See Soare [19], XI, Lemma 3.1.) Hence by the Limit Lemma, there is a computable function h, and an enumeration of the c.e. sets, such that, for each stage s $X_s = \{j \leq s \mid (\exists n \in W_j)[D_n \subseteq \overline{A}][s] = \mathrm{ra}(h(\cdot, s))$. We build uniformly for each e a c.e. set V_e whose index $W_{g(e)}$ is given by the Recursion Theorem. Initially, $V_e = \emptyset$. Thus, initially, $h(g(e), s) = 0$.

Stage s. We are ready to define $f(s)$.

$e \leq s$ will either be *active* or *inactive*. If e is inactive, then $\Phi_e^A(e) \downarrow [t]$ for all $t \in [u,s]$ where u was the stage where it was declared inactive. If e was inactive at stage $s-1$ but now $\Phi^A(e) \uparrow [s]$ we will declare that e is active.

Suppose we see a stage s where $\Phi_e^A(e)[s] \downarrow$, for some active e.

Then for all active $e' \leq s$, we do the following. While the active requirements e' are *unconfirmed* for all $s' > s$ if we see $\Phi_{e'}^A(e') \downarrow [s']$ (so that $s' = s$ for e), we will put $\varphi_{e'}^A(e') \cap \overline{A}[s']$ into $V_{e'}[s']$, and hence $W_{g(e')}[t(s')]$ at some stage $t(s')$ given by the s-m-n Theorem, and we wait for a stage $u > s'$ where either (i) $h(g(e'), u) = 1$ or (ii) $\Phi_{e'}^A(e') \uparrow [u]$ (Here we are using the hat convention, that if a computation changes on its use, it must diverge for at least one stage.) When we see this, we will declare that e' is confirmed. Once confirmed we do not allow a requirement active at stage s to become unconfirmed until stage $f(s)$ defined below, so that each $e \leq s$ will have at most one activity cycle before we define $f(s)$.

We then define $f(s)$ to be the first stage where all arguments n, active at stage s are confirmed, or for which during the confirmation process for active $e' \leq s$, $\Phi_n^A(n) \uparrow [s']$ for all s'.

Then f is clearly computable, and works. For suppose $\Phi_e^A(e) \downarrow [t]$ for all $t \in [s, f(s)]$. Then it must be that $h(g(e), u) = 1$ for some $u \in [s, f(s)]$. Then the only way that $\Phi_e^A(e) \uparrow [v]$ for some $v > f(s)$, is that $h(g(e), v) = 0$ by the conventions on X. By the Limit Lemma, this can happen only finitely often. The result follows. □

References

[1] K. Ambos-Spies, Antimitotic recursively enumerable sets, Z. Math. Logik Grundlagen Math. **31** (1985), 461-467.

[2] K. Ambos-Spies, Rod Downey, and Martin Monath, Notes on Sacks Splitting Theorem, in preparation.

[3] Ambos-Spies K., C. Jockusch, R. Shore, and R. Soare, An algebraic decomposition of recursively enumerable degrees and the coincidence of several degree classes with the promptly simple degrees, Trans. Amer. Math. Soc., Vol. 281 (1984), 109-128.

[4] M. Bickford and C. F. Mills, Lowness properties of r.e. sets, unpublished.

[5] E. Chih, Non-splitting of recursively enumerable sets, J. Symb. Logic, **80** (2015), 609-635

[6] R. G. Downey and D. R. Hirschfeldt, *Algorithmic Randomness and Complexity*, Springer-Verlag, 2010.

[7] R. Downey and N. Greenberg, A Hierarchy of Low Computably Enumerable Degrees, Unifying Classes and Natural Definability, 172pp, Submitted.

[8] R. Downey, C. Jockush, P. E. Schupp, Asymptotic density and computably enumerable sets, Journal of Mathematical Logic Vol. 13 (2013) (43 pages) `https://doi.org/10.1142/S0219061313500050`.

[9] R. G. Downey and Keng Meng Ng, Splitting into degrees with low computational strength, accepted, Annals of Pure and Applied Logic.

[10] R. G. Downey and R. Shore, Splitting theorems and the jump operator, Annals Pure and Applied Logic, **94** (1998), 45-52.

[11] R. G. Downey and T. A. Slaman, Completely mitotic r.e. degrees, Ann. Pure Appl. Logic **41** (1989), 119-152.

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

[13] R. Friedberg, Three theorems on recursive enumeration, J. Symbolic Logic, **23** (1958), 308-316.

[14] E. Griffiths, Completely mitotic c.e. degrees and non-jump inversion, Annals of Pure and Applied Logic, **132** (2005), 181-207

[15] M. Ingrassia and S. Lempp, Jumps of nontrivial splittings of r.e. sets, Z math. Logic. Grundlagen Math., **36** (1990), 285-292.

[16] R. Ladner, Mitotic recursively enumerable sets, J. Symbolic Logic, **38** (1973), 199-211.

[17] R. Ladner, A completely mitotic nonrecursive recursively enumerable degree, Trans. Amer. Math. Soc. **184** (1973), 479-507.

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

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

Rod G. Downey School of Mathematics and Statistics, Victoria University, P.O. Box 600, Wellington, New Zealand.

E-mail address: rod.downey@vuw.ac.nz

Ellen Chih Mathematics Department, University of California, Berkeley, CA 94720, USA.