PROMPT SIMPLICITY, ARRAY COMPUTABILITY AND CUPPING

ROD DOWNEY, NOAM GREENBERG, JOSEPH S. MILLER, AND REBECCA WEBER

Abstract. We show that the class of c.e. degrees that can be joined to $\mathbf{0}'$ by an array computable c.e. degree properly contains the class of promptly simple degrees.

1. Introduction

The main class examined in this paper is the class of *array computable* degrees introduced by Downey, Jockusch and Stob in [11, 12]. We recall from [12] that a c.e. degree $\boldsymbol{a}$ is array noncomputable iff for all $g \leqslant_{wtt} 0'$ there is a function $f \leqslant_T \boldsymbol{a}$ that is not dominated by g; that is, for infinitely many x we have $f(x) > g(x)$. Whilst the original definition was in terms of "very strong arrays", the given characterization highlights the fact that being array noncomputable is akin to being non-low$_2$, where $\boldsymbol{a}$ is non-low$_2$ using the same definition, but replacing $\leqslant_{wtt}$ by $\leqslant_T$. Indeed in [12], the authors showed that the array noncomputable degrees share many of the properties of the non-low$_2$ degrees with respect to cupping, lattice embeddings and the like.

The importance of the notion of array noncomputability has been highlighted by recent work on randomness and domination/tracing properties in computability theory. (For examples, see Cholak, Coles, Downey and Herrmann [3], Downey, Hirschfeldt, Nies, Terwijn [10], Downey and Hirschfeldt [8], Kummer [15], Schaeffer [21], Stephan and Wu [18], Terwijn and Zambella [24].) For instance, Kummer [15] shows that the c.e. degrees containing c.e. sets of high Kolmogorov complexity are exactly the array noncomputable degrees. Nies [19] shows that the K-trivial degrees are all array computable. Ng, Stephan and Wu [18] prove the interesting result that a c.e. degree is array computable if and only if the degree consists of only reals in the field generated by the left c.e. reals. Ismukhametov [14] proves the remarkable result that the array computable c.e. degrees are precisely the c.e. degrees that have strong minimal covers in the Turing degrees, and hence the array computable degrees are definable in the Turing degrees if the computably enumerable degrees are definable.

In this paper we plan to add to our understanding of the lowness concept of array computability.

Two of the most influential concepts in the computability theory of the computably enumerable sets are the concepts of *lowness* and *prompt simplicity*.

The first, second and fourth author's research was supported by the Marsden Fund of New Zealand, via postdoctoral fellowships. The second and third author were supported by a grant from the Singapore Institute for Mathematical Sciences. Most of this research was carried out whilst the first, second and third authors were supported by the Singapore IMS during the *Computational Prospects of Infinity* programme, in 2005.

The former concerns the intrinsic information content of a set; to say that a set is low for some class would indicate that it has little information content and resembles a computable set relative to the class in question. For instance, the classical notion of lowness, that $A' \equiv_T 0'$, says that the jump operator cannot distinguish between a low set and the empty set as oracles using Turing reducibility. There is a huge literature on low (c.e.) sets and how they resemble computable sets. For example, Soare [22] proved that if a c.e. set A is low and coinfinite then its lattice of c.e. supersets is effectively isomorphic to the lattice of all c.e. sets, and Robinson [20] proved that Sacks's splitting theorem can be carried out above any low c.e. degree, whereas Lachlan [16] demonstrated that this fact is not true for a general incomplete c.e. degree.

Prompt simplicity was introduced by Maass [17] in connection with automorphisms of the lattice of c.e. sets. Recall that a co-infinite c.e. set A is called promptly simple iff there is a computable function p, and an enumeration of $A = \cup_s A_s$, such that for all $e < \omega$, if the c.e. set W_e (the e^{th} c.e. set in the canonical indexing) is infinite, then

$$\exists^\infty x, s \;\; [x \in W_{e,\,\text{at}\,s} \;\;\&\;\; x \in A_{p(s)}].$$

Thus, prompt simplicity is a *dynamic* property which expresses *how fast* elements can enter a set. It turns out that this concept, and variations, was the key to the solution of many longstanding questions about the lattice of c.e. sets as discussed in Harrington and Soare [13].

Roughly speaking, a promptly simple set resembles the halting problem in its dynamic properties. In a beautiful paper, Ambos-Spies, Jockusch, Shore and Soare [1] showed that lowness and promptness are intimately related. In particular, they showed that whilst promptly simple sets might not be Turing complete, they did indeed resemble complete sets in that they were *low cuppable.*

Theorem 1.1 ([1], Theorem 1.17)**.** *If A is promptly simple then there is a low c.e. set B such that*

$$A \oplus B \equiv_T 0'.$$

In fact, the main result of [1] says a lot more. The promptly simple degrees coincide with the low cuppable ones; and the promptly simple degrees and the cappable degrees (i.e. those $\mathbf{a}$ for which there is a $\mathbf{b} \neq \mathbf{0}$ with $\mathbf{a} \cap \mathbf{b} = \mathbf{0}$) form an algebraic decomposition of the c.e. degrees into a strong filter and an ideal.

In this paper we will re-examine Theorem 1.1 with respect to array computability. Recent work on concepts like K-triviality (e.g., Nies [19]) and almost deep degrees (Cholak, Grozsek, Slaman [4]) as well as older work of Bickford and Mills [2], shows that the low degrees have a deep and poorly understood structure.

We prove the following two theorems (everything here is c.e.).

Theorem 1.2. *Every promptly simple degree is cuppable to $\mathbf{0'}$ by an array computable degree.*

The proof of this result uses a new technique involving certain priority rearrangements which is of independent technical interest. This result might lead the reader to believe that AC cupping could be used to characterize the promptly simple degrees, but this hope fails.

Theorem 1.3. *There is a degree that is cuppable to $\mathbf{0'}$ by an array computable degree but which is not promptly simple.*

One question we have not been able to answer is Nies' question of whether every promptly simple degree is *superlow* cuppable (recall that A is *superlow* if $A' \equiv_{tt} 0'$). A positive answer would supersede our result because all superlow c.e. degrees are array computable (Schaeffer [21]).[1] We remark that in some sense an affirmative answer would be the best one could hope for, in terms of the concepts mentioned. At first glance, other possibilities include almost deep degrees, shown to exist in [4], and K-trivial degrees. A degree $\mathbf{a}$ is almost deep if for all low $\mathbf{b}$, $\mathbf{a} \cup \mathbf{b}$ is also low, and hence, since there are low promptly simple degrees, they certainly cannot all be cupped by an almost deep degree. The K-trivial degrees are bounded above by an incomplete (in fact, low_2) degree, and hence again (using the fact that no upper cone of the c.e. degrees can avoid the promptly simple degrees [1]) there are promptly simple degrees that cannot be cupped to $\mathbf{0'}$ by a K-trivial degree.

A related question is whether the low c.e. degrees and the super-low c.e. degrees are elementarily equivalent. In a later paper, Downey, Greenberg and Weber [7] show that this is not the case by showing that no array noncomputable degree can bound a 1-3-1, whereas there are low embeddings of 1-3-1. We also remark that the present paper led the authors to examine other permitting notions relating to array computability and domination. It turns out that another related notion (of being totally ω-c.e.) corresponds exactly to embeddability of certain upper semilattices in the c.e. degrees. These results and further generalizations can be found in Downey, Greenberg and Weber [7] and Downey and Greenberg [6]. These notions of permitting arise as a modification of the following characterization of array computability:

Definition 1.4. Let $f\colon \omega \to \omega$. A Δ^0_2 function g is *f-c.e.* if there is an effective approximation $g(x,s)$ of g (that is, $g(\cdot) = \lim_s g(\cdot, s)$) such that for all x,

$$|\{s : g(x,s) \neq g(x,s+1)\}| \leq f(x).$$

Lemma 1.5 ([11]). *Let $f\colon \omega \to \omega$ be strictly increasing and computable. Then a c.e. degree $\boldsymbol{a}$ is array computable iff every $g \leqslant_T \boldsymbol{a}$ is f-c.e.*

In the sequel we use either the identity function or $x \mapsto x+1$ for f.

1.1. **Notation and Conventions.** Notation is standard and follows Soare [23, XIV, s.4].

Theorems 1.2 and 1.3 are priority constructions done on trees, and certain standard conventions will apply to both.

First, the stage number bounds everything. In particular, we assume $W_e[s] \subset s$. Likewise, we assume the domains of Turing functionals are closed downwards at every stage. If $\operatorname{dom}\Phi$ is used in an inequality, it means $|\operatorname{dom}\Phi|$, or equivalently, $\max\{\operatorname{dom}\Phi\}+1$. Stage s ends whenever a string of length s becomes accessible. Finally, all uses are nondecreasing in argument.

2. PS $\subseteq$ AC CUPPABLE

In this section we prove Theorem 1.2.

We are given a computably enumerable set A that permits promptly. In response, we enumerate a set B that will be array computable and join A up to $0'$.

---

[1] An intermediate question that we do not know the answer to is whether every promptly simple degree is cuppable by a *low* array computable degree.

We define moving markers $\gamma(m)$. At any given time, a marker may be defined or undefined; when we *erase* a marker it means that we make it undefined.

Let $\langle\Psi_e\rangle$ be an effective enumeration of all Turing functionals. To ensure that B is array computable, the requirement Q_e will, in the case that $\Psi_e(B) = g$ is total, construct an id-c.e. approximation for g. The strategy for Q_e will not be to impose restraint (since we cannot restrain the global join requirement) but rather to remove potentially dangerous $\Psi_e(B)$ computations by "disengaging" γ-markers from computations.

The construction is done on a tree of strategies — the strategy for Q_e measures whether or not $\Psi_{e'}(B)$ is total, for all $e' < e$. The e^{th} level is devoted to Q_e, and each node has two outcomes, the infinite and the finite (the infinite is stronger). As is customary, if α is a node working for Q_e then we let $\Psi_\alpha = \Psi_e$.

By the recursion theorem (and the slowdown lemma), let p be a computable function that witnesses that A permits promptly with respect to an array of c.e. sets which are enumerated during the construction (an enumeration occurs when a number is tested for prompt permission; in a flashback-like narrative, we will specify the sets we need during the verifications).

General instructions about erasure are: if $\gamma(m)$ is enumerated into B, then it is immediately erased; if $\gamma(m)$ is erased, then all $\gamma(n)$ for $n > m$ are also immediately erased.

For any Q_e-node α and any input k we determine, at stage s, whether to accept a $\Psi_\alpha(B, k)$ computation or not. A computation will not be accepted if there are too many γ-markers below its use. For each k we will determine a number n (which decreases with s, until α is initialised) such that $\Psi_\alpha(B, k)$ is believed only if for $m \geqslant n$, there are no $\gamma(m)$-markers below the use $\psi_\alpha(k)$. The way this is done during the construction is to define, at every stage, a (non-decreasing) sequence $\langle k_\alpha(m)\rangle$ such that $k_\alpha(m)$ is the least input k that does not want to disengage m. So a computation $\Psi_\alpha(B, k)\downarrow$ is *α-confirmed* if for all m, $\gamma(m)\downarrow < \psi_\alpha(k)$ implies $k \geqslant k_\alpha(m)$.

2.1. **Construction.** At stage 0, we begin by defining, for every node α of level e on the tree, $k_\alpha(m) = 0$ for $m \leqslant e$ and $k_\alpha(m) = m - e$ otherwise.

At stage s, if m enters $0'$, then we put $\gamma(m)\,[s]$ into B (and $\gamma(m)$ remains undefined forever). Note by our action at the end of the stage $\gamma(m)\,[s]$ will be defined for such m.

Next, we construct the path of nodes accessible at stage s. We also describe which nodes are *expansionary* at stage s.

The empty node is accessible at every stage. Suppose that a node α is accessible at s; let t be the last α-expansionary stage (0 if there is no such stage). Then s is α-expansionary if $\operatorname{dom}\Psi_\alpha(B)[s] > t$.

If s is not α-expansionary then the finite outcome of α is accessible at s; α does not act at s.

Suppose s is α-expansionary. The computation $\Psi_\alpha(B)\,[s]$ is *unconfirmed due to* m if for some $k < \operatorname{dom}\Psi_\alpha(B)$ we have both $k < k_\alpha(m)$ and $\gamma(m) < \psi_\alpha(k)$ (in particular, $\gamma(m)$ is defined). Let m be the smallest number such that $\Psi_\alpha(B)$ is unconfirmed due to m. (What to do if $\Psi_\alpha(B)$ is confirmed? Not likely, but then $\alpha^\frown\infty$ is accessible and α does nothing).

First, α asks for prompt permission from A: it looks for a change in $A \upharpoonright \gamma(m)$ between stages s and $p(s)$. If permission is granted, then α erases $\gamma(m)$, and the infinite outcome of α is accessible.

If permission is not granted, then α enumerates $\gamma(m)$ into B (and erases it). In this case, we end the stage — there are no more accessible nodes. Also, we update k_β for all nodes β such that $\beta^\frown\infty \subseteq \alpha$. We redefine $k_\beta(m) = s$ (and to keep things in order, redefine $k_\beta(m+i) = s+i$ for all $i < \omega$). [This is because $\Psi_\beta(B)$ computations which may have been confirmed were nonetheless injured by $\gamma(m)$ entering B. To avoid a repeated injury (by the same m), the injured computations need to get less tolerant; from now, they too wish to disengage m.]

At the end of the stage, all nodes β which are of length $\geq s$ or which *lie to the right* of the path of accessible nodes are *initialised*. This means that we redefine $k_\beta(m) = 0$ for all $m < s$.

Finally, for every $m \geq s$ which is not yet in $0'$ and such that $\gamma(m)$ is not defined, we redefine $\gamma(m)$ with large value (keeping the sequence $\langle\gamma(m)\rangle$ increasing).

2.2. **Verifications.**

Define the *true path* to be the leftmost path of nodes that are accessible infinitely often.

Lemma 2.1. *The true path is infinite; each node on the true path is eventually never initialised.*

Proof. Note that the empty node is on the true path. Let α be a node on the true path. If there are finitely many α-expansionary stages then the finite outcome of α is on the true path and is eventually never initialised.

Suppose that there are infinitely many α-expansionary stages. If at infinitely many of those stages we find $\Psi_\alpha(B)$ to be confirmed then $\alpha^\frown\infty$ is on the true path. Otherwise, we enumerate an auxiliary set $U = U_\alpha$; if at an expansionary stage s, α asks for permission from A to erase $\gamma(m)$, then we enumerate $\gamma(m)$ into U at s. Then U is infinite. By the properties of p, there are infinitely many stages at which A gives α permission to erase the marker; at each such stage, $\alpha^\frown\infty$ is accessible; so $\alpha^\frown\infty$ is on the true path. □

Claim 2.2. *Let α be a node and let $m < \omega$. Suppose that at some stage t, $k_\alpha(m) = 0$. Then at no stage $s \geqslant t$ does α erase $\gamma(m)$.*

Proof. Denote by $k_\alpha(m)\,[s]$ the value of $k_\alpha(m)$ at the beginning of stage s.

By induction on $s < \omega$, we show that for all α such that $k_\alpha(m) = 0\,[s]$,

(1) For all β extending α, $k_\beta(m) = 0\,[s]$.
(2) α does not erase $\gamma(m)$ at s.
(3) $k_\alpha(m) = 0\,[s+1]$.

First note that (1) holds at stage $s = 0$ by the initial definition of k_α and k_β. (2) and (3) hold at 0 because at stage 0 nobody does anything.

Suppose that (1)-(3) hold for stage $s-1$. Suppose that $k_\alpha(m) = 0\,[s]$.

Let β be any node extending α. If $k_\alpha(m) = 0\,[s-1]$ then by $(1)(\alpha, s-1)$, $k_\beta(m) = 0\,[s-1]$; and then by $(3)(\beta, s-1)$, $k_\beta(m) = 0\,[s]$. Otherwise, $k_\alpha(m)$ was set to be 0 during stage $s-1$, which means that α was initialised at stage $s-1$ and $m < s-1$. Then β was also initialised at stage $s-1$ and so $k_\beta(m) = 0\,[s]$. This establishes $(1)(s)$.

Now at s, α only wants to disengage from m if for some $k < k_\alpha(m)\,[s]$ we have $\gamma(m) < \psi_\alpha(k)$. Since there is no such k, α does not erase m at s and so (2)(s) holds.

$k_\alpha(m)$ is increased at stage s only if some node β extending α erases m at s. By (1)(s) and (2)(s), no node β extending α erases m at s. It follows that $k_\alpha(m)\,[s+1] = k_\alpha(m)\,[s] = 0$. □

Claim 2.3. *For every m, $\gamma(m)$ is erased only finitely often.*

Proof. We show this by induction on m. Assume the claim holds up to $m-1$.

Let $\alpha_0 \subset \alpha_1 \subset \cdots \subset \alpha_{m-1}$ be the first m nodes on the true path.

Let $s^* = s^*(m)$ be a stage as follows:

(1) α_{m-1} is never initialised after s^*;
(2) α_{m-1} is accessible at s^* (so $s^* > m$ and every node β which lies to the right of α_{m-1} is initialised at s^*);
(3) No $\gamma(n)$ for $n < m$ is ever erased after s^*.

The assumptions on s^*, together with the initial definitions of k_β and with Claim 2.2, show that after s^*, only nodes among $\alpha_0, \dots, \alpha_{m-1}$ may erase $\gamma(m)$. By *reverse* induction on $i < m$, we show that after some stage, α_i does not erase $\gamma(m)$.

Suppose that after some stage $s_i > s^*$, no node properly extending α_i erases $\gamma(m)$. Of course s_{m-1} exists.

At s_i, $k_{\alpha_i}(m)$ reaches a fixed value.

Suppose for contradiction that α_i erases $\gamma(m)$ infinitely often. At any stage s at which α_i erases $\gamma(m)$, we enumerate $\gamma(m)\,[s]$ into an auxiliary set $U = U_{\alpha_i,m}$. By our assumptions on p, there is some stage $s > s_i$ at which $\operatorname{dom}\Psi_{\alpha_i}(B) > k_{\alpha_i}(m)$ and at which α_i receives permission from A to erase $\gamma(m)$ without enumerating it into B (there are infinitely many stages at which permission is granted).

The computation $\Psi_{\alpha_i}(B) \upharpoonright k_{\alpha_i}(m)$ which we have at stage s is not injured at s and in fact will never be injured and always be confirmed. Thus after stage s, α_i never erases $\gamma(m)$. □

Corollary 2.4. *$0' \leqslant_T A \oplus B$.*

id-*c.e. approximations.* Suppose that $\Psi_e(B)$ is total. Let α be on the e^{th} level of the true path (so $\Psi_\alpha = \Psi_e$). We know that there are infinitely many α-expansionary stages and hence $\alpha^\frown\infty$ is on the true path.

Let r^* be the last stage at which α is initialised. Thus we have $k_\alpha(m) = 0$ from stage r^* onwards exactly for $m \leqslant e$ or $m \leqslant r^*$ (and we may assume that $r^* > e$). For $m \geqslant r^*$ we have $k_\alpha(m) \geqslant m - e$ from stage r^* onwards.

Let s^* be a stage after which the marker $\gamma(r^*)$ is never erased.

Definition 2.5. We *believe* a computation $\Psi_\alpha(k)\,[s]$ if $s > s^*$, $\alpha^\frown\infty$ is accessible at s, and if for all β such that $\beta^\frown\infty \subseteq \alpha$ and for all m such that $k \geqslant k_\alpha(m)\,[s]$ we have $\operatorname{dom}\Psi_\beta(B) > k_\beta(m)\,[s]$.

We can effectively recognise believable computations.

Claim 2.6. *The correct $\Psi_\alpha(k)$ computation is eventually believable at every stage at which $\alpha^\frown\infty$ is accessible.*

Proof. For all m such that the final value of $k_\alpha(m)$ is at most k and each β as above, $k_\beta(m)$ eventually stabilizes, and $\operatorname{dom}\Psi_\beta(B)$ goes to infinity on the stages at which $\alpha^\frown\infty$ is accessible. As there are only finitely many such m ($m < r^*$ or $m \leqslant e+k$), there is a stage at which all such $k_\beta(m)$ have stabilized and $\operatorname{dom}\Psi_\beta(B)$ has surpassed them. □

Claim 2.7. *Suppose that $\Psi_\alpha(k)$ is believable at s. Then it is confirmed at s (perhaps after α erased a marker).*

Proof. Because $\alpha^\frown\infty$ is accessible at s. □

Claim 2.8. *Suppose that $\Psi_\alpha(k)$ is believable at stage s, and that the computation is injured at stage $t \geqslant s$, by some $\gamma(m)\,[t]$. Then either m enters $0'$ at t, or $\gamma(m)$ is enumerated into B by some node extending $\alpha^\frown\infty$ (so $k < k_\alpha(m)$ from t onwards).*

This shows that different believable $\Psi_\alpha(k)$ computations are injured by $\gamma(m)$, for any given m, at most once: no number goes into $0'$ more than once; and if $k < k_\alpha(m)$ from t onwards then no computation $\Psi_\alpha(k)\,[s']$ with $\gamma(m) < \psi_\alpha(k)\,[s']$ is believable. Also, there are at most k such m's: suppose that $\Psi_\alpha(k)\,[s]$ is believable and that $\gamma(m) < \psi_\alpha(k)\,[s]$ injures the computation later. Then $k \geqslant k_\alpha(m)\,[s] \geqslant m-e$ (as $m < r^*$ is impossible), and $m \geq e$ because otherwise $k_\alpha(m)\,[s] = 0$ and there is no such k. Overall we see that the believable computations form an id-c.e. approximation for $\Psi_e(B)$.

Proof of Claim 2.8: Assume m does not enter $0'$ at t. We show that the node that enumerates $\gamma(m)$ into B at stage t must extend $\alpha^\frown\infty$.

Since the computation $\Psi_\alpha(k)$ is believable, by Claim 2.7 it is confirmed. Therefore $k \geq k_\alpha(m)[s]$ forever after, and α itself will never enumerate $\gamma(m)$ into B.

We have $m < \gamma(m)\,[t] < \psi_\alpha(k)\,[s] < s$. Every β which lies to the right of $\alpha^\frown\infty$ is initialised at stage s and so we set $k_\beta(m) = 0$ at s. By Claim 2.2, β never erases $\gamma(m)$ after s.

Of course, α is not initialised after s and so $\gamma(m)$ cannot be erased at t by some node β such that α extends the finite outcome of β. It remains to see that no β such that $\beta^\frown\infty \subseteq \alpha$ erases $\gamma(m)$ at t.

If it does, then there is some l such that $\Psi_\beta(l)\downarrow[t]$ and $\gamma(m) < \psi_\beta(l)\,[t]$ but $l < k_\beta(m)\,[t]$. Now no-one erased $\gamma(m)$ between stages s and t so $\gamma(m)\,[s] = \gamma(m)\,[t]$ and $k_\beta(m)\,[s] = k_\beta(m)\,[t]$. Thus $l < k_\beta(m)\,[s]$. The computation $\Psi_\alpha(k)\,[s]$ is believable so it is confirmed: $k \geqslant k_\alpha(m)\,[s]$. By the definition of believability we have $\operatorname{dom}\Psi_\beta(B)\,[s] > k_\beta(m)\,[s] > l$.

At s, the computation $\Psi_\beta(l)$ is confirmed. This means that we must have $\gamma(m)\,[s] \geqslant \psi_\beta(l)\,[s]$, for $k_\beta(m)\,[s] > l$. No markers $\gamma(n)$ for $n \leqslant m$ were erased between s and t; so B didn't change between s and t on numbers below $\psi_\beta(l)\,[s]$. Thus $\psi_\beta(l)\,[s] = \psi_\beta(l)\,[t]$ and altogether we get a contradiction to $\gamma(m)\,[t] < \psi_\beta(l)\,[t]$. □

3. AC cuppable $\neq$ PS

In this section we prove Theorem 1.3.

To do this, we construct a c.e. set A; we construct a c.e. set B such that **a** and **b** form a minimal pair (thus ensuring that **a** is not promptly simple — see Ambos-Spies, Jockusch, Shore and Soare [1]); and we construct a c.e. set C that is array computable, and a Turing functional Γ such that $\Gamma(A \oplus C) = 0'$.

We need to meet the following requirements:

$$\begin{aligned} P_e &: \quad \overline{B} \neq W_e \\ N_e &: \quad \Phi_e(A) = \Phi_e(B) = g \;\Rightarrow\; g \equiv_T 0 \\ Q_i &: \quad \Psi_i(C) \text{ total } \;\Rightarrow\; \Psi_i(C) \text{ is } (\mathrm{id}+1)\text{-c.e.} \end{aligned}$$

Note that if the requirements above are met, the fact that B is not computable implies that neither A nor C are complete or computable.

3.1. **The strategy.** We first discuss the basic strategy for meeting each requirement. The first two are familiar: P_e is met by picking a follower x and holding onto it until it enters W_e, in which case it is enumerated into B as well. N_e is met by following the Lachlan strategy of monitoring the length of agreement, and allowing only one side of the computations $\Phi_e(A)$ and $\Phi_e(B)$ be injured at a time; our construction will be done on a tree of strategies, and so the restraint will be implicit in the machinery of the tree.

To meet a Q_i requirement, we construct an $(\mathrm{id}+1)$-c.e. approximation for $\Psi_i(C)$ by preserving each computation $\Psi_i(C, x)$. To do this, we prevent most numbers from entering C.

As mentioned, we make use of a tree of strategies. The tree is used to arrange the restraints imposed by N_e and Q_i requirements, which are of infinitary type.

We order the requirements effectively and let each level of the tree be devoted to a single requirement. Each Q- or N-node on the tree has two children representing two possible outcomes. The outcomes are f (finite) and ∞ (infinite), the latter guessing totality of $\Psi_i(C)$ or that $\Phi_e(A) = \Phi_e(B)$ are total. The P-nodes do not impose any restraint and so have a single outcome.

The priority ordering on the tree is the lexicographic ordering generated by the ordering $\infty < f$. We say that a node α lies to the left of node β if α is stronger than β but they are not $\subset$-comparable.

The driving force behind the construction is numbers entering $0'$. At stage s we have an increasing sequence of markers (Γ-uses) $\gamma(0, s), \ldots, \gamma(s, s)$. When at some stage s, a number m enters $0'$, we must put $\gamma(m) = \gamma(m, s)$ into either A or C. Some nodes will have a preference between A and C: for example, Q-nodes may want to prevent $\gamma(m)$ from entering C, and so will favour A. Another possible scenario is that an N-node, currently during an expansionary stage, allows us to put numbers into either A or B but not both. A longer P-node might wish to enumerate a follower into B; it would thus prefer $\gamma(m)$ to go into C rather than A. The third possibility is that an N-node impose finite restraint on A while at a non-expansionary stage; again it will have an opinion as to where $\gamma(m)$ should go.

As expected, the final decision lies with the strongest node that has any preference.

The first scenario described (a Q-node α wants to protect a computation and so puts a marker into A) needs more elaboration. As described so far, it is possible that α acts infinitely many times, each time injuring some N-node $\beta \supseteq \alpha^\frown\infty$. To avoid this, α needs to take preventive action. Thus, instead of simply removing markers when they pose problems, whenever α sees a new computation (on x, say) that it wants to preserve, it puts all markers that are potentially dangerous in the future (that is, all markers between x and its $\Psi_i(C)$-use) into A straight away. Injury for β is avoided because at this stage (compared with the stage at which we really would have needed to put markers into A), the markers are still large relative to the uses of β-computations.

We note a further delicacy. As in the proof of Theorem 1.2, it is not enough to verify that the markers are "confirmed"; we need to take care of inputs between markers by actively confirming larger numbers. For this, a *test point* $d(\alpha)$ is appointed for a Q-node α; it is for convergence of $\Psi_i(C)$ on that test point that we wait, and this point may be larger than the next *marker* we would confirm. Naïvely applying this strategy may result in a single marker being driven to infinity. We need to find a compromise; thus the test point $d(\alpha)$ may at some stages be *mobile*, where we keep lifting its value to the next marker (to make sure that markers are not raised infinitely often); but when it really needs to act, it becomes *stationary.*

The rules for test points are as follows. A mobile test point $d(\alpha)$ must always have the value of some marker. Thus if the marker is enumerated into A or C, we raise the value of the test point to be the next marker (which will be the least one chosen at the end of the stage). However, if a marker *smaller* than the test point is enumerated, then the test point becomes stationary and will not point at a marker; indeed it will not move until α acts to confirm it.

3.2. **Construction.** At stage s, we define the path of nodes accessible at s; for each node α, we define how it acts. At stage s, a new number m is enumerated into $0'$. The stage always ends with some accessible node enumerating $\gamma(m)$ into either A or C. The usual conventions for working with markers apply; if $\gamma(n)$ is enumerated into a set then all of $\gamma(k)$ for $k \in (n, s)$ are also enumerated into the same set and are redefined with large values at the end of the stage.

For any node β, let $r(\beta)[s]$ be the last stage before s at which β was initialised.

The first node accessible at any stage is the root. Suppose that α is accessible at s; the requirement to which α is assigned determines its actions.

α is assigned to P_e. If α does not have a follower (this is the first time we visit α since it was initialised), we appoint a fresh (large) follower for α.

Let x be α's follower. If $x \in B$ or if $x \notin W_e$ then α does nothing; its single child is accessible next.

Otherwise, α enumerates x into B and $\gamma(m)$ into C. (The stage is now ended).

α is assigned to N_e. If $\Phi_e(A)(x)\downarrow = \Phi_e(B)(x)\downarrow$, then let

$$u_e(x) = \max\{\varphi_e(A)(x), \varphi_e(B)(x)\}.$$

If there is a Q-node β with $\beta^\frown\infty \subseteq \alpha$ and $d(\beta)[s] < u_e(x)[s]$, then α assumes markers in $(d(\beta), u_e(x))$ will be enumerated by β. Therefore we say a computation on input x is *believable* at s if there is no such Q-node β.

The modified *length of agreement*, $\ell(\alpha)[s]$, is the maximal y such that on every $x < y$ there is a believable computation on x at stage s.

Let $r = r(\alpha^\frown f)[s]$. If $\ell(\alpha, s) > \ell(\alpha, r)$ then we say that s is *expansionary* for α and let ∞ be α's outcome.

Suppose that s is not expansionary. If $\gamma(m) \leqslant r$ then α enumerates $\gamma(m)$ into C and halts the stage. Otherwise, the outcome is finite, and $\alpha^\frown f$ is accessible.

α is assigned to Q_i. If s is the first stage since $r(\alpha)[s]$ at which α is accessible, then $d(\alpha)[s]$ is not yet defined; we define it with value equal to the least marker greater than $r(\alpha)[s]$, and set it to a mobile state.

Let t be the last stage at which α was accessible. There are two cases.

(1) $t > r(\alpha)[s]$ and at t, α acted to confirm $d(\alpha)[t]$ (case 2b(i) below). In this case we let $\alpha^\frown\infty$ be accessible. [We believe totality as the "length of confirmation" has increased. We are not worried about any of the markers entering C because we already protected computations by removing dangerous markers. We now let weaker nodes act.]

(2) Otherwise. There are two cases.
   (a) $\Psi_i(C)(d(\alpha)[s])\uparrow$. In this case there is nothing we can do on behalf of the requirement.
      (i) If $\gamma(m) > r(\alpha^\frown f)[s]$ then we let $\alpha^\frown f$ be accessible.
      (ii) If $\gamma(m) \leqslant r(\alpha^\frown f)[s]$ then α enumerates $\gamma(m)$ into C and halts the stage.
   (b) $\Psi_i(C)(d(\alpha)[s])\downarrow$. In this case we want to make progress.
      (i) If $\gamma(m) > d(\alpha)[s]$ then we *confirm* $d(\alpha)$: we enumerate all markers greater than $d(\alpha)[s]$ into A and halt the stage. Note that $d(\alpha)[s]$ itself (whether or not it is a marker), is not enumerated. As described later, we now pick new large markers; we raise the value of $d(\alpha)$ and redefine $d(\alpha)[s+1]$ to be the least marker among the new ones just chosen; we set it to be mobile.
      [At the next stage at which α is accessible, we follow case 1.]
      (ii) However, if $\gamma(m) \leqslant d(\alpha)[s]$ then there is no point in confirming $d(\alpha)[s]$ as smaller numbers will have to enter a set at this stage. We let $\alpha^\frown\infty$ be accessible and do nothing else.

At the end of the stage we initialise nodes that lie to the right of the accessible nodes (and all nodes of length $> s$), and reassign a large value (greater than s) to any marker $\gamma(k)$ ($k \leqslant s$) which is not defined ($\gamma(k)$ was just enumerated into a set or $k = s$). [Note we do not initialise *extensions* of the final accessible node.] Let $\gamma(k)$ be the least marker enumerated at stage s. If $\gamma(k) = d(\alpha)[s]$ for a Q-node α, define $d(\alpha)[s+1]$ to point to a new marker. On the other hand, if $\gamma(k) < d(\alpha)[s]$, then $d(\alpha)$ will no longer point to a marker value and hence has become *stationary* until α acts to confirm it.

3.3. **Verification.**

Claim 3.1. *Suppose that α is accessible at stage s, and that m enters $0'$ at s. Then $r(\alpha) \leqslant \gamma(m)$ $[s]$.*

Proof. This is by induction on α. For the root, we know that $r(\langle\rangle)[s] = 0$ for all s. Suppose that α is accessible at s. If α is a P-node then $r(\alpha) = r(\beta)$ $[s]$ where β is α's only child. If α is a Q- or N-node, then $r(\alpha) = r(\alpha^\frown\infty)$ $[s]$. If $\gamma(m) < r(\alpha^\frown f)$ $[s]$ then $\alpha^\frown f$ cannot be accessible at s: instead of going to $\alpha^\frown f$, α enumerates $\gamma(m)$ into C and halts the stage. □

Corollary 3.2. *If a node β is initialised at some stage r, then after r, β never enumerates a number smaller than r into any set.*

Proof. There are three kinds of enumerations: enumeration of $\gamma(m)$ (where m enters $0'$); enumeration of a follower into B (by a P-node); and enumeration of markers into A by a Q-node that is confirming its test point.

By claim 3.1, if at stage $s > r$, m enters $0'$ and β enumerates $\gamma(m)$, then (as β is accessible) $\gamma(m) \geqslant r(\beta)[s] \geqslant r$.

If β is a P-node, then after r it picks new followers, all greater than r.

And if β is a Q-node that is confirming $d(\beta)$, then by definition, $d(\beta) \geqslant r(\beta)$ $[s]$ (it is initially assigned to a larger value and can then only increase), and only markers greater than $d(\beta)$ are enumerated. □

Lemma 3.3. *For every m, the marker $\gamma(m)$ is eventually fixed.*

Proof. Suppose that after some stage s_0, the markers $\gamma(0), \dots, \gamma(m)$ do not change. Also suppose that after s_0, $m+1$ does not enter $0'$, so the only way $\gamma(m+1)$ can enter a set after stage s_0 is when some Q-node confirms a smaller $d(\alpha)[s]$ (necessarily $d(\alpha)[s] \geqslant \gamma(m)$), in which case $\gamma(m+1)$ will be enumerated into A.

Let α be such a node. After confirming at stage s, a new value is set for $\gamma(m+1)$, which is also the new value of $d(\alpha)$. At s, $d(\alpha)$ is set to be mobile; and it can never be reverted to be stationary (only $\gamma(m)$ enumeration can do that). So after s, we always have $d(\alpha) \geqslant \gamma(m+1)$ (even if they increase together). Therefore, after stage s_0, there is at most one stage s at which α's action causes $\gamma(m+1)$ to be enumerated into A.

Consider a Q-node of length $\geqslant s_0, \gamma(m)$. Any test point it ever appoints is at least $\gamma(m+1)$ and as in the previous paragraph, if it does appoint it to be $\gamma(m+1)$ it remains mobile until it is lifted to be greater than $\gamma(m+1)$. So such a node never enumerates any $\gamma(m+1)$ into A. This shows that after s_0, only finitely many nodes enumerate a $\gamma(m+1)$ into A.

[An alternative construction could have been: do not allow nodes of length k to enumerate $\gamma(k)$ (i.e., always set $d(\alpha) \geqslant \gamma(k)$). Then you would need this current lemma to show finite injury on the true path.] □

Corollary 3.4. $0' \leqslant_T A \oplus C$.

Proof. Whenever m goes into $0'$, $\gamma(m)$ enters either A or C, and is redefined with a large value. □

Lemma 3.5. *Suppose that α is a node on the tree that is accessible infinitely often and that is eventually never initialised. Then there is an immediate successor of α on the tree that is accessible infinitely often and is eventually never initialised.*

Since the root is always accessible and is never initialised, inductively applying the lemma shows the existence of an infinite true path.

Proof. Let $r_0 = r(\alpha)[\omega]$ be the last stage at which α is initialised. Of course, everything depends on the requirement to which α is assigned.

α is assigned to P_e. After stage r_0, α is assigned a follower x; that follower is never cancelled. After r_0, α acts at most once; after that, whenever α is accessible, so is its child. Also, the child is never initialised without α also being initialised.

α is assigned to N_e. Suppose that there is a last α-expansionary stage $r \geqslant r_0$. After r, $\alpha^\frown f$ is never initialised. After r, α only enumerates a marker $\gamma(m)$ into C if $\gamma(m) \leqslant r$, which is to say, finitely often. At other stages, whenever α is accessible, so is $\alpha^\frown f$.

If there are infinitely many expansionary stages, then $\alpha^\frown\infty$ is accessible infinitely often; it is not initialised after r_0.

α is assigned to Q_i. If there are infinitely many stages at which α confirms its test point, then $\alpha^\frown\infty$ is accessible infinitely often.

Suppose that there is a last stage $r_1 > r_0$ at which α confirms $d(\alpha)$. If there is no such stage, let $r_1 = r_0$. Suppose that at the end of r_1, we set $d(\alpha) = \gamma(k)$. Let $r_2 > r_1$ be a stage after which the value of $\gamma(k)$ is fixed. Then after r_2, the value of $d(\alpha)$ is also fixed (it may or may not have become stationary before r_2). After r_2, case (2b(ii)) only holds if $\gamma(m) \leqslant d(\alpha)$, so finitely many times. So there is a stage $r_3 > r_2$ after which $\alpha^\frown\infty$ is never accessible (so $\alpha^\frown f$ is not initialised after stage r_3). Later, case (2a(ii)) applies only if $\gamma(m) \leqslant r_3$; again, finitely many times. So after some stage $r_4 > r_3$, whenever α is accessible, so is $\alpha^\frown f$. □

Corollary 3.6. *Every P-requirement is met, so B is not computable.*

Proof. Standard. □

Lemma 3.7. *Every Q_i requirement is met. Hence C is array computable.*

Proof. Let α on the true path work for Q_i. Suppose that $\Psi_i(C)$ is total. There are infinitely many stages at which α confirms its test point. For otherwise, as argued in the proof above, there would be a final fixed value for $d(\alpha)$, and a stage after which case (2b(ii)) doesn't hold; but this contradicts $\Psi_i(C)$ converging permanently on that final value.

Definition 3.8. We say that a number $x < \omega$ is *confirmed* at stage s if $\Psi_i(C)(x)\downarrow\ [s]$ with use smaller than the least marker greater than x.

The point, of course, is that a confirmed computation can only be injured by a number not greater than the input. Note this is exactly what was accomplished in Case 2(b)(i) of the construction.

Suppose that α is last initialised at stage r_0. At the next stage r_1 at which α is accessible, α sets $d(\alpha) = \gamma(k_0)$ for some k_0. Let r_2 be the least stage $\geqslant r_1$ after which $\gamma(k_0)$ is always fixed. At the end of r_2 we have $x_0 := d(\alpha)[r_2 + 1] \leqslant \gamma(k_0)[r_2 + 1]$, because in between r_1 and r_2 either no markers below $\gamma(k_0)$ move, so $d(\alpha)$ and $\gamma(k_0)$ move together, or some marker smaller than $\gamma(k_0)$ moves, fixing $d(\alpha)$ below $\gamma(k_0)$. In fact we will have equality, because at a later stage α confirms its test point, an act which moves $\gamma(k_0)$ again.

We prove three claims that suggest a way to approximate $\Psi_i(C)$.

Claim 3.9. *Let $s > r_2$ be a stage at which α acts (to confirm its test point). Suppose that $x \in [x_0, d(\alpha)[s]]$ is confirmed at the end of s. Then it is confirmed at any later stage at which α acts.*

Proof. By assumption, α is not initialized at or after stage s. Suppose that at the end of stage s, $\gamma(m-1) < x \leqslant \gamma(m)$. We have $\gamma(m) \leqslant d(\alpha)$ $[s+1]$. By assumption, $\psi_i(C)(x) < \gamma(m)$ $[s+1]$. Let $t > s$ be the next stage at which α acts. If $\gamma(m-1)$ is not enumerated between s and t, then the computation is preserved and so it is of course still confirmed. Otherwise, let $\gamma(n)$ be the smallest marker enumerated between t and s, say at stage u. Then at the end of u we fix $d(\alpha)$ to be stationary and redefine a greater $\gamma(n)$. At t, $d(\alpha)$ is confirmed; since there is no marker in $(x, d(\alpha)[t]]$, x is also confirmed. □

Claim 3.10. *Let $s > r_2$ be a stage at which α acts and let $n \in [x_0, d(\alpha)[s]]$ be a marker. Then n is confirmed at the end of s.*

Proof. If $d(\alpha)[s] = n$, then n is confirmed at stage s. Otherwise, let $t < s$ be the least stage at the end of which $d(\alpha) > n$. Then $d(\alpha)[t] = n$ (n is already a marker at t) and α acts at t and confirms n. □

Claim 3.11. *Let $s > r_2$ be a stage at which α acts. Let $x \in [x_0, d(\alpha)[s]]$. Suppose that $\Psi_i(C)(x)[s]$ is injured at a later stage u by some $y > x$. Then at the end of the next stage t at which α acts, x is confirmed.*

Proof. Let $\gamma(m-1)[s] < x \leqslant \gamma(m)$ $[s]$. We must have $\gamma(m) \leqslant d(\alpha)$ $[s]$; otherwise, $\gamma(m)[s]$ is redefined at s to be larger than $\psi_i(C)(x)$ and then x is confirmed. Then $\gamma(m)$ is confirmed at the end of s and so $y = \gamma(m)[s]$. Also at the end of s we have $d(\alpha) > y$. It follows that at the end of u we have $\gamma(m-1) < x < d(\alpha) < \gamma(m)$ (since $d(\alpha)$ is stationary). We thus see that at t, the confirmation of $d(\alpha)$ is also a confirmation of x. □

The following is an approximation for $g = \Psi_i(C)$: at a stage $s > r_2$ at which α confirms $d(\alpha)$, guess $g(x) = \Psi_i(C)(x)$ for all $x \leqslant d(\alpha)[s]$. Then the claims ensure that a guessed computation can be injured by some $y > x$ at most once after s. It follows that this approximation is $\mathrm{id}+1$-c.e. □

Lemma 3.12. *Every N_e requirement is met. As $B >_T 0$, it follows that A and B form a minimal pair.*

Proof. Let α on the true path work for N_e. Suppose that $\Phi_e(A) = \Phi_e(B) = g$ are total and equal. We first argue that $\ell(\alpha)[s] \to \infty$ (and so $\alpha^\frown\infty$ is on the true path). This is because for all x, eventually there is a permanent $\Phi_e(A)(x) = \Phi_e(B)(x)$ with use $u_e(x)$; for Q-nodes β such that $\beta^\frown\infty \subseteq \alpha$ we know that $d(\beta)[s] \to \infty$; and there are finitely many such β.

Assume that after stage r_1, α is never initialised and no P-node $\beta \subset \alpha$ enumerates a follower into B.

The familiar argument of Lachlan's that shows that g is computable goes through, provided that we can show that:

(1) If $s > r_1$ is an α-expansionary stage, then at s numbers do not enter both A and B.
(2) If $s > r_1$ is an α-expansionary stage and $x < \ell(\alpha)[s]$, then numbers below $u_e(x)[s]$ do not enter either A or B until the next α-expansionary stage.

The first is immediate from the construction; it holds at every stage, not just expansionary stages. We verify the second point. Now suppose that $s > r_1$ is α-expansionary and let $x < \ell(\alpha)[s]$.

- Nodes that lie to the right of $\alpha^\frown\infty$ are initialised at stage s; they never enumerate anything smaller than s (which is in turn greater than $u_e(x)[s]$) into any set.
- Nodes that lie to the left of α are not accessible after r_1.
- P-nodes $\beta \subset \alpha$ do not enumerate numbers into B after stage r_1.
- Nodes extending $\alpha^\frown\infty$ are not accessible between s and the next α-expansionary stage.

Thus the only possible culprits are Q-nodes $\beta \subset \alpha$ that confirm $d(\beta)$. If $\beta^\frown f \subseteq \alpha$ then β does not confirm $d(\beta)$ after r_1; if it does, then at the next stage at which it is accessible, α will be initialised. However, if $\beta^\frown\infty \subseteq \alpha$ and β confirms $d(\beta)$ at $t > s$, then it only enumerates numbers greater than $d(\beta)[t] \geqslant d(\beta)[s]$. But x is believable at s, which means $u_e(x)[s] \leqslant d(\beta)[s]$. □

References

[1] K. Ambos-Spies, C. Jockusch Jr., R.A. Shore, and R.I. 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.

[2] M. Bickford and C. Mills, *Lowness properties of r.e. sets*, typewritten unpublished manuscript.

[3] P. Cholak, R. Coles, R. Downey, and E. Herrmann, *Automorphisms of the lattice of Π^0_1 classes: perfect thin classes and anr degrees*, *Trans. Amer. Math. Soc.* Vol. 353 (2001), 4899-4924.

[4] P. Cholak, M. Groszek, and T. Slaman, *An almost deep degree,* J. Symbolic Logic, Vol. 66(2) (2001), 881-901.

[5] P. Cholak, R. Downey, and M. Stob, *Automorphisms of the lattice of recursively enumerable sets: promptly simple sets,* Trans. American Math. Society, 332 (1992), 555-570.

[6] R. Downey and N. Greenberg, *Domination and definability II : The ω^ω case*, in preparation.

[7] R. Downey, N. Greenberg and R. Weber, *Domination and definability I : The ω case*, in preparation.

[8] R. Downey and D. Hirschfeldt, *Algorithmic Randomness and Complexity*, Springer-Verlag to appear.

[9] R. Downey, D. Hirschfeldt, A. Nies, and F. Stephan, *Trivial reals*, extended abstract in *Computability and Complexity in Analysis* Malaga, (Electronic Notes in Theoretical Computer Science, and proceedings, edited by Brattka, Schröder, Weihrauch, FernUniversität, 294-6/2002, 37-55),July, 2002. Final version appears in *Proceedings of the 7th and 8th Asian Logic Conferences*, (R. Downey, Ding Decheng, Tung Shi Ping, Qiu Yu Hui, Mariko Yasuugi, and Guohua Wu (eds)), World Scientific, Singapore (2003) 103-131.

[10] R. Downey, D. Hirschfeldt, A. Nies, and S. Terwijn, *Calibrating randomness,* to appear, Bulletin Symbolic Logic.

[11] R. Downey, C. Jockusch, and M. Stob, *Array nonrecursive sets and multiple permitting arguments,* in *Recursion Theory Week* (Ambos-Spies, Muller, Sacks, eds.) Lecture Notes in Mathematics 1432, Springer-Verlag, Heidelberg, 1990, 141–174.

[12] R. Downey, C. Jockusch, and M. Stob, *Array nonrecursive degrees and genericity*, in *Computability, Enumerability, Unsolvability* (Cooper, Slaman, Wainer, eds.), London Mathematical Society Lecture Notes Series 224, Cambridge University Press (1996), 93–105.

[13] L. Harrington and R. Soare, *Post's Program and incomplete recursively enumerable sets*, Proc. Natl. Acad. of Sci. USA 88 (1991), 10242-10246.

[14] S. Ishmukhametov, *Weak recursive degrees and a problem of Spector,* in *Recursion Theory and Complexity*, (ed. M. Arslanov and S. Lempp), de Gruyter, (Berlin, 1999), 81-88.

[15] M. Kummer, *Kolmogorov complexity and instance complexity of recursively enumerable sets,* SIAM Journal of Computing, Vol. 25 (1996), 1123-1143.

[16] A. Lachlan, *A recursively enumerable degree which will not split over all lesser ones,* Ann. Math. Logic, Vol. 9, (1975), 307-365.

[17] W. Maass, *Characterization of the recursively enumerable sets with supersets effectively isomorphic to all recursively enumerable sets,* Trans. Amer. Math. Soc., Vol. 279 (1983), 311-336.

[18] Ng Keng Meng, F. Stephan and G. Wu, *The degrees of weakly computable reals*, in preparation.

[19] A. Nies, *Lowness properties and randomness*, Advances in Mathematics Vol. 197 (2005), 274-305.

[20] R.W. Robinson, *Jump restricted interpolation in the recursively enumerable degrees,* Annals of Math., Vol 93 (1971), 586-596.

[21] B. Schaeffer, *Dynamic notions of genericity and array noncomputability*, Ann. Pure Appl. Logic, Vol. 95(1-3) (1998), 37-69.

[22] R.I. Soare, *Automorphisms of the lattice of recursively enumerable sets Part II: Low sets,* Annals of of Math. Logic, Vol. 22 (1982), 69-107.

[23] R.I. Soare, *Recursively enumerable sets and degrees* (Springer, Berlin, 1987).

[24] S. Terwijn and D. Zambella, *Algorithmic randomness and lowness,* Journal of Symbolic Logic, Vol. 66 (2001), 1199-1205.

School of Mathematics, Statistics and Computer Science, Victoria University, P.O. Box 600, Wellington, New Zealand

E-mail address: Rod.Downey@vuw.ac.nz

School of Mathematics, Statistics and Computer Science, Victoria University, P.O. Box 600, Wellington, New Zealand

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

Department of Mathematics, 196 Auditorium Road, University of Connecticut, Unit 3009, Storrs, CT 06269-3009

E-mail address: joseph.miller@math.uconn.edu

Department of Mathematics, Dartmouth College, 6188 Bradley Hall Hanover, NH 03755-3551

E-mail address: rweber@math.dartmouth.edu