A HIERARCHY OF COMPUTABLY ENUMERABLE DEGREES

ROD DOWNEY AND NOAM GREENBERG

ABSTRACT. We introduce a new hierarchy of computably enumerable degrees. This hierarchy is based on computable ordinal notations measuring complexity of approximation of Δ_2^0 functions. This hierarchy unifies and classifies the combinatorics of a number of diverse constructions in computability theory. It does so along the lines of the high degrees (Martin) and the array noncomputable degrees (Downey, Jocksuch and Stob). The hierarchy also gives a number of natural definability results in the c.e. degrees, including a definable antichain.

1. INTRODUCTION

Ever since Post's original paper [62], two recurrent themes in computability theory have been the understanding the *dynamic* nature of constructions, and *definability* in the natural structures of computability theory such as the computably enumerable sets and degree classes. Beautiful example of this phenomenom are the definable solution to Post's problem of Harrington and Soare [44], and the definability of the double jump classes for c.e. sets of Cholak and Harrington [18].

The work reported here can be seen as contributing to both areas. The goal of this research announcement is to report on the current results of a program introduced by the authors and some co-authors which seeks to understand the fine structure of relationship between *dynamic* properties of sets and functions, their *definability*, and their *algorithmic complexity*.

Much of this announcement will report on the authors' new monograph [24]. In that monograph, along with the companion papers [27] and [25], we introduce a new hierarchy of computably enumerable (c.e.) degrees based on the complexity of approximations of functions in these degrees.

The reader might well ask why we need yet another hierarchy in computability theory. In this announcement, we also discuss three aspects of this work.

(i) A new methodology for classifying and unifying the combinatorics of a number of constructions from the literature.
(ii) New natural definability results in the c.e. degrees. These definability results are in the low_2 degrees and hence are not covered by the current metatheorems of Nies, Shore and Slaman [61]. Moreover they are amongst the very few natural definability results in the theory of the c.e. Turing degrees.
(iii) The introduction of a number of construction techniques which are injury-free and highly non-uniform. These would seem to have wider applications.

We will explain our results in relation to the above, but first we need to introduce the main player.

The authors' research was supported by the Marsden Fund of New Zealand.

2. α-COMPUTABLY APPROXIMABLE FUNCTIONS

Our main concern will be functions and sets which are Δ^0_2 (equivalently, computbale from the halting problem $\varnothing'$). Shoenfield's Limit Lemma states that these are the functions g that have a computable approximation: a uniformly computable sequence $\langle g_s \rangle$ of functions which converge to g — in the discrete topology, so this means that for all n, $g_s(n) = g(n)$ for all but finitely many s.

Our main idea is to classify Turing degrees of c.e. sets by the complexity of the possible approximations to functions in the degree. Intuitively speaking, an approximation is simple if we do not need to change our mind often about the value of the function. One way to quantify this is by seeking simple bounds on the "mind-change" function $\#\{s : g_{s+1}(x) \neq g_s(x)\}$. For example, we know that functions are weak truth-table reducible to the halting problem if and only if they have an approximation whose mind-change function is bounded by some computable function.

Ershov [38, 39, 40] developed and refined this idea and created a transfinite hierarchy of complexity. His idea is that a witness for the approximation $\langle g_s(n) \rangle$ converging is "counting down" some ordinal α. More formally, associated to the approximation $\langle g_s \rangle$ we have a computable sequence $\langle o_s \rangle$ of functions $o_s \colon \omega \to \alpha$; $o_s(n)$ can be thought of as the "number of times" we are still allowed to change our mind; the requirements are that $\langle o_s(n) \rangle$ is non-increasing in s, and that $o_{s+1}(n) < o_s(n)$ if $g_{s+1}(n) \neq g_s(n)$. Such an approximation is call an *α-computable approximation*, and a function which has an α-computable approximation we call *α-computably approximable* (or α-c.a.).

Note that a mind-change function with a computable bound is the same as an ω-computable approximation. In general, the greater the ordinal α, the more opportunities we have to change our mind, and therefore the more complicated the limit function g can be. One way to get an idea for the meaning of this notion is to iterate: an ω^2-computable approximation is one for which the mind-change function has an ω-c.a. bound. Namely, a value $o_s(n) = \omega k + m_k$ means that currently we allow ourselves m_k further mind-changes; but when we run out, at some stage $t > s$, we can choose a new number m_{k-1} (as large as we like) and decare $o_t(n) = \omega \cdot (k-1) + m_{k-1}$, giving us m_{k-1} further opportunities to change our mind. In short, this is an approximation for which we have a computable bound on how many times we are allowed to change our mind about how many times we are allowed to change our mind. Inductively this works for $\omega^3, \omega^4, \dots$ (the behaviour beyond ω^ω is different).

We remark that in [24] we also point out the relationship of these notions with *bounded* versions of the jump, as articulated by Coles, Downey and LaForte [19], and Anderson and Csima [6]. To wit, if $\hat{\Phi}_e$ for $e \in \omega$ is an enumeration of all partial wtt-procedures, we may define

$$A^\dagger = \{\langle x, y\rangle \mid \hat{\Phi}^A_x(y)\downarrow\}.$$

Notice that $\varnothing' \equiv_m \varnothing^\dagger$, but if Q is Δ^0_2, then $Q^\dagger$ is also Δ^0_2. Then, for example, in the same way that X being ω-c.a. is equivalent to $X \leqslant_{\text{wtt}} \varnothing'$, $Y \in 2^\omega$ being ω^2-c.a. is equivalent to $Y \leqslant_{\text{wtt}} \varnothing^{\dagger\dagger}$. This relationship is extended in the obvious way to the whole hierarchy.

2.1. **Canonical ordinals.**

The definition above of α-computable approximations requires the functions o_s to be uniformly computable. This makes sense only if α

is presented in a computable way: as a computable well-ordering of ω, or more specially, given by an ordinal notation in Kleene's complete Π^1_1-set $\mathcal{O}$ (more precisely, by the restriction of the partial ordering $\leqslant_{\mathcal{O}}$ to the notations below a given notation). An ordinal notation is essentially a computable copy of an ordinal for which the successor function and the set of limit ordinals are also computable.

But here we run into the problem of picking canonical representatives; not all computable well-orderings of some order-type α, nor all notations, are computably isomorphic. If we defined a function to be α-c.a. if it is a-c.a. for *some* notation a for α then the hierarchy becomes trivial; Ershov showed that every Δ^0_2 function is ω^2-c.a. according to this definition. The point is that the complexity of the approximation can be coded by the well-ordering of ω. This resembles the breakdown of Spector's theorem [67] for strong reducibilities: Kleene showed that if a and a' are two notations for the same ordinal α, then H_a and $H_{a'}$, the iterations of the Turing jump along these notations, are Turing equivalent; this allows us to define the *degree* $\mathbf{0}^{(\alpha)}$. However Moschovakis [58] showed that H_a and $H_{a'}$ may fail to be m-equivalent.

Ershov and his school often solve this problem by fixing a Π^1_1 path through $\mathcal{O}$; but there is nothing canonical about such a choice. For our purposes ordinals below ε_0 more than suffice. For such ordinals, we can put an extra condition on computable copies which makes the resulting subclass computably unique. The problem above with ω^2 was with copies in which we cannot tell in which copy of ω a given element is. That is, given some $\beta < \omega^2$ we want to know the m, n such that $\beta = \omega n + m$. In general, below ε_0 what we need is an *effective Cantor normal form.* For ordinals $\alpha < \varepsilon_0$, the exponents $\alpha_1, \dots, \alpha_n$ appearing in the Cantor normal form

$$\alpha = \omega^{\alpha_1} m_1 + \cdots + \omega^{\alpha_n} m_n$$

of α are strictly less than α, and so the function from an ω-copy of α giving the Cantor normal form (with the exponents again being elements of our copy) is well-defined, and can be asked to be computable. Note that from this normal form we can easily identify limit ordinals and successors, so each canonical ordinal is necessarily an ordinal notation. Any two canonical copies of an ordinal $\alpha < \varepsilon_0$ are computably isomorphic, and so the resulting notion of an α-computable approximation, and hence of α-c.a. functions and sets, is well-defined.

3. A degree hierarchy

Equipped with the robust notion of α-c.a. functions, we now turn to Turing degrees.

Definition 3.1. Let $\alpha < \varepsilon_0$. A Turing degree $\mathbf{d}$ is *totally α-c.a.* if every function $g \in \mathbf{d}$ is α-c.a.

Note that if $\mathbf{d}$ is totally α-c.a. then in fact every $f \leqslant_{\mathrm{T}} \mathbf{d}$ is α-c.a.; taking any $g \in \mathbf{d}$ we notice that f is α-c.a. if and only if $f \oplus g$ is α-c.a. We remark that the case $\alpha = \omega$ is of particular interest, and the definition of totally ω-c.a. degrees was first suggested by Joseph Miller. Throughout this announcement, unless otherwise mentioned, we concentrate on c.e. degrees.

If $\alpha < \beta < \varepsilon_0$ then there is a β-c.a. function which is not α-c.a. However, some of these differences collapse when we close under Turing reductions. For example, suppose that $\mathbf{d}$ is totally $(\alpha \cdot 2)$-c.a. and let $f \in \mathbf{d}$. Let $g(n) = f\restriction_n$. Since $g \leqslant_{\mathrm{T}} f$ we can find an $(\alpha\cdot 2)$-computable approximation (g_s, o_s) of g. Now either for infinitely many n, $o_\omega(n) = \lim_s o_s(n)$ is smaller than α; by waiting, we can use this to give an α-c.a. approximation of f. In the other case, ignoring finitely many inputs we live entirely in the second copy of α inside $\alpha \cdot 2$, and again we can translate that to an α-computable approximation of f. Thus, every totally $(\alpha\cdot 2)$-c.a. degree is actually totally α-c.a. Note the non-uniformity in this argument (it is necessary).

Recall that an ordinal is closed under (ordinal) addition if and only if it is a power of ω.

Theorem 3.2 ([24])**.** *Let $\alpha < \varepsilon_0$. There is a totally α-c.a. degree that is not totally γ-c.a. for any $\gamma < \alpha$ if and only if α is a power of ω.*

We thus get a proper hierarchy of classes of degrees, indexed by the powers of ω below ε_0.

Sketch of proof. In the easy direction, if α is not a power of ω then there is some β such that $\alpha \in (\omega^\beta, \omega^{\beta+1})$. The non-uniform argument above generalises to show that any totally α-c.a. degree is also totally ω^β-c.a.

For the main direction, we will sketch the priority argument. Assume that α is closed under addition. We enumerate a c.e. set D whose Turing degree is totally α-c.a., but not totally γ-c.a. for any $\gamma < \alpha$. To witness the last part, we enumerate a Turing functional Λ and ensure that $\Lambda(D)$ is not γ-c.a. for any $\gamma < \alpha$. For $\gamma < \alpha$, we can effectively enumerate all γ-c.a. functions in a list $\langle f^{e,\gamma}\rangle$ together with $(\gamma+1)$-computable approximations $\langle f^{e,\gamma}_s, o^{e,\gamma}_s\rangle$; and then we aim to meet the requirements:

$P^{e,\gamma}$**:** $\Lambda(D) \neq f^{e,\gamma}$.

Q_e**:** If $\Phi_e(D)$ is total, then it is α-c.a.

It turns out that the simplest construction one would hope work, does work. The strategy for meeting a $P^{e,\gamma}$ requirement is to pick a witness p (a "follower") and change the value of $\Lambda(D,p)$ whenever we observe that $f^{e,\gamma}_s(p) = \Lambda_s(D_s, p)$. This is of course done by enumerating the use $\lambda_s(p)$ of the old computation into D_{s+1}. This requirement is guaranteed to succeed and only act during finitely many stages; this is because $\langle f^{e,\gamma}_s(p)\rangle$ must stabilise.

The requirement Q_e follows $\Phi_e(D)$ and at various stages "certifies" observed computations $\Phi_{e,s}(D_s, x)$. Fix some x and let s_0 be the least stage at which a computation $\Phi_e(D,x)$ is certified. Of course only finitely many weaker-priority positive requirements $P^{d,\gamma}$ have chosen followers by stage s_0. All other weaker-priority positive requirements will be prohibited from injuring any certified $\Phi_e(D,x)$ computation in the future. Now we know that $o^{e,\gamma}_{s_0}(p) \leqslant \gamma$ for each "old" $P^{d,\gamma}$. This means that the "number of times" that $P^{d,\gamma}$ will ever act is bounded by γ. The fact that α is closed under addition allows us to tally up all the ordinals for "old" $P^{d,\gamma}$. Since we ensure that any injury to a certified computation indeed comes from the action of an old $P^{d,\gamma}$, this allows us to give a bound (strictly below α) on the "number of times" $\Phi_e(D,x)$ will be certified.

It is not actually easy to see why we need an infinite-injury construction; the reason is an intricate interplay between two negative requirements affecting a third positive requirement. However once we organise everything on a tree of strategies in the usual way, this problem goes away. A node working for Q_e will have Π_2/Σ_2 outcomes based on the totality of $\Phi_e(D)$. One thing to note is that we need to make $\Lambda(D)$ total; this means that when a node working for $P^{d,\gamma}$ enumerates $\lambda_s(p)$

into D, we need to immediately redefine a new (presumably large) value $\lambda_{s+1}(p)$. This is done before recovery is observed for the $\Phi_e(D,x)$ computation, and so the same node might indeed injure $\Phi_e(D,x)$ multiple times. The ordinal computation takes this into account. □

3.1. **Refinements of the hierarchy.** Definition 3.1 is not the most general we could make.

Definition 3.3. Let $\alpha < \varepsilon_0$. A Turing degree $\mathbf{d}$ is *totally* $< \alpha$*-c.a.* if every function $g \in \mathbf{d}$ is γ-c.a. for some $\gamma < \alpha$.

The non-uniform collapsing argument above can be used to show that the totally $< \omega^{\beta+1}$-c.a. degrees are precisely the totally ω^{β}-c.a. degrees, so we get nothing new in this case. So the only interesting cases are limits of powers of ω. The construction proving Theorem 3.2, as it constructs a single function $\Lambda(D)$, shows that for any limit β there is a degree which is totally ω^{β}-c.a. but not totally $< \omega^{\beta}$-c.a. At these limit levels we do indeed get a new level:

Theorem 3.4 ([24])**.** *Let $\alpha < \varepsilon_0$. There is a totally $< \alpha$-c.a. degree that is not totally γ-c.a. for any $\gamma < \alpha$ if and only if α is a limit of powers of ω.*

The first new level, that of the totally $< \omega^{\omega}$-c.a. degrees, is the main class investigated in [24]. The proof of Theorem 3.4 is an elaboration on the proof of Theorem 3.2. We cannot uniformly in $\gamma < \alpha$ define some $f \leqslant_{\mathrm{T}} D$ which is not γ-c.a. (or we would string them together to get a function which is not γ-c.a. for any $\gamma < \alpha$). In the construction, the problem arises when a requirement Q_e tries to give $\Phi_e(D)$ a γ-computable approximation for some $\gamma < \alpha$ (which γ, to be determined); but below γ are requirements $P^{d,\beta}$ for arbitrarily large $\beta < \alpha$. To resolve this, a "mother node" η attached to some $\beta < \alpha$ (a power of ω) starts a functional Λ_η with the aim of making $\Lambda_\eta(D)$ not β-c.a. Only nodes working for $P^{d,\beta}$ whose mother node lies above the node τ working for Q_e are allowed to injure the computations certified by τ; the ordinal bound on the injuries to τ is given by adding the finitely many mother nodes above τ; here we use the fact that α is a limit of powers of ω. The nodes $P^{d,\beta}$ whose mother node η lie below τ will not injure computations certified by τ because only η is responsible for making $\Lambda_\eta(D)$ total; η is only accessible during τ-expansionary stages at which we have seen the τ-computations recover, and so new $\Lambda_\eta(D)$ uses can be always set to be larger than the uses of the τ-computations.

Yet another refinement of our hierarchy is motivated by the class of *array computable degrees*. As we see below in Section 4, our new definability results will allow us to tie a number of natural constructions together in new degree classes in the same way as the array noncomputable degrees did in Downey, Jockusch and Stob [31, 32]. A c.e. degree $\mathbf{d}$ is array computable if and only if for some (equivalently, every) order-function h, every $f \in \mathbf{d}$ has a computable approximation whose mind-change function is bounded by h. In other words, this is like being totally ω-c.a., but with a uniform bound on the mind-change function.

This can be generalised to ordinals beyond ω as follows. Let $\alpha < \varepsilon_0$. An *α-order-function* is a computable function $h\colon \omega \to \alpha$ which is non-decreasing and unbounded in α. For such a function h, an *h-computable approximation* is an α-computable approximation $\langle f_s, o_s \rangle$ for which $o_0(n) < h(n)$ for all n; a function is *h-c.a.* if it has an h-computable approximation. We then define a degree $\mathbf{d}$ to be *uniformly totally α-c.a.* if for some (equivalently, all) α-order-function(s) h, every $f \in \mathbf{d}$ is h-c.a. Thus a c.e. degree is array computable if and only if it is uniformly totally ω-c.a.

Every totally ω^β-c.a. degree is uniformly totaly $\omega^{\beta+1}$-c.a., and so the new, uniform levels of our hierarchy slot in between the previous levels; indeed a theorem akin to Theorems 3.2 and 3.4 states that if α is a power of ω then there are totally α-c.a. degrees which are not uniformly so, and uniformly totally α-c.a. degrees which are not totally γ-c.a. for any $\gamma < \alpha$. (In particular, there are c.e. degrees which are totally ω-c.a. but not array computable.) And if α is a limit of powers of ω then every totally $< \alpha$-c.a. degree is uniformly totally α-c.a., and this implication is proper.

3.2. **Domination.** The original definition in [31] of array computability was restricted to c.e. degrees (and in that paper was shown to be equivalent to being uniformly totally ω-c.a.). This definition was extended in [32] to non-c.e. degrees but using domination instead. Indeed, a c.e. degree $\mathbf{d}$ is array computable if and only if some ω-c.a. function dominates every function in $\mathbf{d}$; and this was used as a general definition.

Such a characterisation holds for all levels of our hierarchy. Indeed, for all $\alpha \leqslant \varepsilon_0$, a c.e. degree $\mathbf{d}$ is...

(1) uniformly totally α-c.a. if and only if it is *uniformly α-c.a. dominated*: some α-c.a. function dominates all functions in $\mathbf{d}$ (Downey, Greenberg, McInerney, see [24]);
(2) totally α-c.a. if and only if it is *α-c.a. dominated*: every function in $\mathbf{d}$ is dominated by some α-c.a. function (Diamondstone, Greenberg, Turetsky [22]);
(3) totally $< \alpha$-c.a. if and only if it is *$< \alpha$-c.a. dominated*: for every $f \in \mathbf{d}$ there is some $\gamma < \alpha$ and a γ-c.a. function dominating f.

3.3. **Lowness.** The definition of array computability in terms of domination shows that it is a strengthening of being low$_2$: a degree $\mathbf{d}$ is low$_2$ if and only if some Δ^0_2 function dominates all functions in $\mathbf{d}$. In fact, the ability to list all α-c.a. functions effectively shows that for any $\alpha < \varepsilon_0$, every totally α-c.a. degree is low$_2$. In other words, our transfinite hierarchy is a (non-exhaustive) refinement of the low$_2$ degrees. As we hall later see, this shows that our definability results cannot be achieved using the Nies-Shore-Slaman metatheorems, as the latter concern classes that are invariant under the double jump.

We remark that being low$_2$ is often used in constructions involving a given a totally α-c.a. degree, using a Δ^0_3 decision procedure about totality of functions computable from that degree. Knowing that a degree $\mathbf{d}$ is totally α-c.a. gives us further information; once we have guessed that $\Gamma(D)$ is total (for some $D \in \mathbf{d}$), we can guess an α-computable approximation for $\Gamma(D)$. A somewhat stranger phenomenon occurs in constructions of totally α-c.a. degrees: often one first proves that the constructed set is low$_2$, then this fact is used to help show that the other requirements are met. These techniques come into play, for example, when investigating maximality in our hierarchy; see Section 7.

It is tempting to guess that all members of this hierarchy are low. For example if A is superlow (meaning that $A' \equiv_{tt} \varnothing'$), then A is certainly array computable and hence totally ω-c.a. Similarly, if $A' \equiv_{tt} \varnothing^{\dagger\dagger}$, then A is certainly totally ω^2-c.a. However, even the array noncomputable degrees contain non-low c.e. sets (Downey, Jockusch and Stob [31]), and as is shown in [24], all levels of the hierarchy contain low sets, but no level contains all low c.e. sets. Thus the hierarchy does not align itself with the low sets in any precise way.

4. Unifying classes

It is quite rare in computability theory to find a single class of degrees which capture precisely the underlying dynamics of a wide class of apparently similar constructions. A good example of this phenomenon is work pioneered by Martin [56] who identified the high c.e. degrees as the degrees of dense simple, maximal, hyperhypersimple and other similar kinds of c.e. sets. Another example would be the of the promptly simple degrees by Ambos-Spies, Jockusch, Shore and Soare [4]. Another more recent example of current great interest is the class of K-trivial sets (introduced by Solovay; see Downey, Hirschfeldt, Nies and Stephan [29] and Nies [60, 59]), which are known to coincide with many other "lowness" classes.

In each case, these classes quantify the necessary amount of "permitting" required to carry out constructions below such a degree. In a typical construction, a requirement makes infinitely many requests, and we quantify how often these requests are granted. Namely, we ask a c.e. oracle to change below some specified use. High degrees correspond to almost-always permitting; all but finitely many requests are granted. Promtly simple degrees correspond to prompt permitting; not all requests will be granted, but some will be granted quickly (within a computable time bound). Similarly, non-K-trivial degrees grant requests which globally have finite weight. The classificactions show that degrees in the class are sufficiently complicated so that they will permit as required; but also that degrees outside the class cannot give such permission, and so cannot bound the kind of object being constructed. In a strong way this says that the standard construction of that kind of object is the only way to build such an object.

As hinted above, an important class that falls in this scheme is the class of array noncomputable degrees; these are the degrees that provide "multiple permitting" [32]: roughly, the n^{th} instance of a request needs to be granted n times. Let us recall some of the results.

Theorem 4.1. *A c.e. degree* $\mathbf{d}$ *is array noncomputable if and only if...*

1. *it is the degree of a perfect thin* Π^0_1 *class* (Downey, Jockusch and Stob [31]; Coles, Downey, Herrmann and Jockusch [16]).
2. *it computes a separating* Π^0_1 *class (the class of separators of a pair of disjoint c.e. sets) which contains no element computing* $\varnothing'$*; it computes a pair of separating classes* $\mathcal{C}_1$ *and* $\mathcal{C}_2$ *such that any* $X \in \mathcal{C}_1$ *and* $Y \in \mathcal{C}_2$ *are Turing incomparable* (Downey, Jockusch, Stob [31])*;*
3. *it contains a c.e. set* A *which infinitely often has maximal (plain) Kolmogorov complexity:* $(\exists^\infty n)\, C(A\upharpoonright_n) \geqslant^+ 2\log n$ (Kummer [48]).
4. *it does not have a strong minimal cover in the Turing degrees* (Ishmukhametov [45]).
5. *it has effective packing dimension 1* (Downey and Greenberg [26])*; it computes a degree which has effective packing dimension 1 but contains no set of effective packing dimension 1* (Downey and Stephenson [36]).

(6) *it contains two left-c.e. reals with no common upper bound in the cL-degrees of left-c.e. reals*[1] (Barmpalias, Downey and Greenberg [12])*; it contains a left-c.e. real (equivalently, a set) which is not cL-reducible to any ML-random left-c.e. real* (Barmpalias, Downey and Greenberg [12]).
(7) *it contains a set which is not reducible to the halting problem with tiny use* (Franklin, Greenberg, Stephan and Wu [42]).
(8) *it computes an integer-valued random sequence* (Barmpalias, Downey and McInerney [10]).

The connection between domination properties, approximation properties, and permitting, exhibited by the array computable degrees, is naturally extended to the totally ω-c.a. degrees. Here multiple permitting is replaced by a non-uniform version: we specify during the construction, rather than in advance, how many times we need an instance of a request to be granted. And indeed, the class of totally ω-c.a. degrees captures the combinatorics of a number of constructions. We defer discussing an improtant one, that of critical triples, to Section 5, as it is related to definability results; here we discuss other results.

4.1. **Algorithmic randomness.** One area of interest in computability theory and theoretical computer science is algorithmic randomness. This is the programme of study which gives meaning to the notion of randomness for individual binary sequences, both finite and infinite. Another way to look at it is as effective measure theory.

4.1.1. *Presentations of left-c.e. reals.* Two related basic notions in this area are those of an effective open set and a left-c.e. real. An open set is effective (or c.e.) if one can effectively enumerate all of its basic clopen subsets (in the real line, the rational intervals it contains). This is the lightface version of the class of open sets. This is fundamental to randomness because the effectively null sets used to define notions of randomness are Π^0_2, i.e., uniform intersections of effective open sets. A real is left-c.e. if it is the limit of an increasing sequence of rational numbers; equivalently, if the left cut it defines is c.e. The left-c.e. reals are those which are the Lebesgue (fair coin) measure of effectively open sets. Random left-c.e. reals are of importance.

In practice, effectively open subsets of Cantor space 2^ω are usually presented using *prefix-free* sets of strings: antichains in $2^{<\omega}$, i.e., sets which contain no comparable strings (under the relation of extension). This comes up in many arguments involving effectively null classes ("tests for randomness"); they also appear as the domains of prefix-free machines, those that are used to define prefix-free Kolmogorov complexity, which in turn gives an equivalent characterisation of ML-randomness, the most useful notion of randomness in this area. Indeed, the random left-c.e. reals are those which are the measures of the domains of universal prefix-free machines (these numbers are known as Chaitin's Ω, and are in a strong sense all equivalent). For more see [28, 60].

Every effectively open set is generated by a c.e. prefix-free set of strings; by padding (at stage s intead of enumerating a string σ, enumerate all of its extensions of length s), one can require the set to actually be computable. Bypassing the open

[1] X is computable Lipschitz reducible to Y ($X \leqslant_{cL} Y$) if it is Turing reducible to Y with use identity + constant.

sets, we say that a prefix-free set A is a *presentation* of a left-c.e. real α if α is the measure of the open set generated by A; directly, if α is the *weight* of A:

$$\alpha = \sum_{\sigma \in A} 2^{-|\sigma|}.$$

Thus, every left-c.e. real has a computable presentation [66]. On the other hand, bizarre things can happen. In [33], Downey and LaForte showed that there exist noncomputable left c.e. real α, all of whose c.e. presentations are computable, but that any real of promptly simple degree has a noncomputable presentation; so do all K-trivial left-c.e. reals, as was shown by Stephan and Wu [68]. We have the following:

Theorem 4.2 ([24])**.** *Let* $\mathbf{d}$ *be a c.e. Turing degree.*

(i) *If* $\mathbf{d}$ *is totally* ω*-c.a., then every left-c.e. real* $\alpha \in \mathbf{d}$ *has a presentation* $A \in \mathbf{d}$.
(ii) *If* $\mathbf{d}$ *is not totally* ω*-c.a., then there is a left-c.e. real* $\alpha \leqslant_{\mathrm{T}} \mathbf{d}$ *and a c.e. set* $B <_{\mathrm{T}} \alpha$ *which computes every presentation of* α.

This result extends the Stephan-Wu Theorem since every K-trivial degree is totally ω-c.a. The proof is uses an elaboration of the "drip feed" strategy used by Downey and LaForte for their result mentioned above.

4.1.2. *Finite randomness.* Another manifestation of total ω-c.a.-ness in algorithmic randomness is in the notion of finite randomness. Recall that a *Martin-Löf* (ML) null set is a Π^0_2 set which is effectively null. That is, it is the intersetion $\bigcap_n U_n$ of a sequence of uniformly effective open sets whose measure tends to 0 with a computable bound; one usually requires $\lambda(U_n) \leqslant 2^{-n}$ for simplicity. The ML-random sequences are those which are not elements of ML-null sets (in fact, there is a greatest ML-null set, a "universal ML-test").

In [14], Brodhead, Downey and Ng introduce a finite version of this notion: instead of the sets U_n being effectively open, they are required to be clopen (here we discuss randomness in Cantor space). It is important that despite being clopen, these sets are presented as open sets: as time passes we see more strings enumerated into U_n; there will be finitely many, but we never know for sure that the enumeration has stopped. (In the language of computability, we say that U_n is given by a c.e. index rather than a canonical index for a finite set.) If we further require that there is a computable bound on the number of strings in each U_n then we obtain the notion of *computably bounded finite randomness*, CB-randomness for short. (Here and below we wilfully identify a set of strings and the open set that it generates.)

Theorem 4.3 (Brodhead, Downey, Ng [14])**.** *The following are equivalent for a c.e. degree* $\mathbf{d}$*:*

(1) $\mathbf{d}$ *contains a CB-random sequence.*
(2) $\mathbf{d}$ *bounds a left-c.e., CB-random real.*
(3) $\mathbf{d}$ *is not totally* ω*-c.a.*

Sketch of proof. We first sketch the permitting direction (iii) $\to$ (ii). We are given $g \in \mathbf{d}$ which is not ω-c.a. We enumerate a left-c.e. real $\alpha \leqslant_{\mathrm{T}} \mathbf{d}$. To make α CB-random, a typical requirement will try to move α to the right as to avoid a given CB test $\langle U_n \rangle$: whenever we see some $\sigma \in U_{n,s}$ and $\alpha_s \in [\sigma]$, we want to move α_{s+1} sufficiently much to the right so as to avoid being in $[\sigma]$. Note that since

$\lambda(U_n) \leqslant 2^{-n}$, the total increase in α required to avoid U_n is bounded by roughly 2^{-n}.

To make $\alpha \leqslant_{\mathrm{T}} \mathbf{d}$ we need to seek permission from g before we move α. The rough plan is to stipulate that if $g_s(n)$ is correct and will never change again, then $\alpha - \alpha_s \leqslant 2^{-n}$. Since $\mathbf{d}$ is a c.e. degree, we can approximate g so that its modulus is computable from $\mathbf{d}$, and so α will be reducible to $\mathbf{d}$. (In fact, by standard manipulations, we may assume that g is lower semicomputable, essentially its own modulus.)

Therefore, if a randomness requirement wants to move α_{s+1} away from some $\sigma \in U_{n,s}$, it needs to await permission, in the form of a change in our approximation for $g_s(n)$. Since the CB-test $\langle U_n \rangle$ comes equipped with a computable function f bounding the number of strings in each U_n, we know how many permissions we need to avoid each U_n. More formally, we carry out the construction; during the verification, we show that if the requirement is not met then we can give an ω-computable approximation to g: we belive a value $g_s(n)$ each time we make a request to move away from a string in U_n. The computable bound on the number of changes is given by f.

The direction (ii) → (iii) tries to reverse the argument above, but there is a small trick. Suppose that α is left-c.e. and has totally ω-c.a. degree. We define a function f as follows: to calculate $f(n)$, first find the least s such that α_s is correct on the first n digits. If $\varphi_{n,s}(n)$ converges then we let $f(n) = \alpha\restriction_{\varphi_n(n)}$; otherwise, we let $f(n) = \alpha\restriction_n$.

Let $\langle f_t \rangle$ be an ω-computable approximation of f, say with bound h on the number of mind-changes. Let I be an infinite computable set of indices for h. We now capture α by a CB-test. For a typical component U of this test, choose some large $n \in I$. Find some t_0 such that $\varphi_{n,t_0}(n)\downarrow$. We let U consist of $\alpha_{t_0}\restriction_n$, and all strings of the form $f_t(n)$, provided that they have length $h(n)$.

The number of strings in U is bounded by $h(n) + 1$; and the measure of U is bounded by $2^{-n} + h(n)2^{-h(n)}$, which can be made small by increasing n and $h(n)$ if necessary. It remains to show that $\alpha \in U$. There are two cases. If $\alpha_{t_0}\restriction_n = \alpha\restriction_n$ then we are done. Otherwise, we know that $f(n) = \alpha\restriction_{h(n)}$, and so $\alpha\restriction_{h(n)} \in U$. □

As an aside, we remark that one can look at variations of CB-randomness as follows. If one removes the requirement on the computable bound on the number of strings in U_n, then we obtain a stronger notion of randomness which is less well-understood; it coincides with ML-randomness on the Δ^0_2 sequences. If one requires that U_n be given as a finite set, rather than enumerated, we get Kurtz randomness [49, 70]. If one requires that not only the number of strings in U_n is computably bounded, but their length is too, then we get the notion of granular randomness [11]. Indeed, the proof above gives a granular test, so in the c.e. degrees, bouding CB-randoms and granular randoms are equivalent, even though these notions of randomness do not coincide.

4.1.3. *DNC and cL reducibility.* A variant of Theorem 4.1(6) was proved by Ambos-Spies, Fang, Losert, Merkle and Monath [5, 2]. A sequence A is *complex* (see [47]) if $C(A\restriction_n) \geqslant h(n)$ for some order function h (computable, non-dcereasing, unbounded). There are several equivalent formalisations, including computing a diagonally noncomputable function with computable use on the bound (wtt). Every ML-random sequence is complex.

Theorem 4.4 (Ambos-Spies, Fang, Losert, Merkle and Monath). *A c.e. degree* $\mathbf{d}$ *is totally* ω*-c.a. if and only if every left-c.e. real* $\alpha \in \mathbf{d}$ *is cL-reducible to a complex left-c.e. real.*

4.2. **Ranked sets.** Barmpalias, Downey and Greenberg showed that the totally ω-c.a. degrees are related to strong reducibilities and the Cantor-Bendixson rank of reals in Π^0_1 classes. A set is *ranked* if it is an element of some countable Π^0_1 class (and so it has a Cantor-Bendixson rank). A linear ordering is *scattered* if it doesn't contain a copy of the rationals (and so repeating the Hausdorff derivative leaves an empty kernel at the end). A set $A \subset \omega$ is *hyperimmune* if it is infinite, and whenever $\langle F_n \rangle$ is a computable sequence of pairwise disjoint finite sets, there is some n such that $F_n \cap A$ is empty. Equivalently, the function which maps n to the n^{th} element of A (by magnitude) is not dominated by any computable function. Finally, a c.e. set is *hypersimple* if its complement is hyperimmune.

These concepts are related: any initial segment of a scattered, computable linear ordering is ranked [15]; If A is c.e. and non-computable, and is the ω-part of a computable linear ordering of order-type $\omega + \omega^*$, then A is hypersimple. The linear ordering $\omega + \omega^*$ is the simplest example of a comptable scattered linear ordering which may have non-computable proper initial segments.

Theorem 4.5 (Barmpalias, Downey and Greenberg [12]). *The following are equivalent for a c.e. degree* $\mathbf{d}$*:*

(1) *Every set in* $\mathbf{d}$ *is wtt-reducible to a ranked set.*
(2) *Every set in* $\mathbf{d}$ *is wtt-reducible to a hypersimple set.*
(3) *Every set in* $\mathbf{d}$ *is wtt-reducible to a proper initial segment of a computable, scattered linear ordering.*
(4) $\mathbf{d}$ *is totally* ω*-c.a.*

Moreover, the equivalence still holds if in any of (1), (2) or (3), "set" is replaced by "c.e. set".

This work extends work of Chisholm, Chubb, Harizanov, Hirschfeldt, Jockusch, McNicholl and Pingrey [15], who showed that every c.e. degree which is not totally ω-c.a. contains a c.e. set which is not wtt-reducible to any ranked set. Independently, Afshari, Barmpalias, Cooper and Stephan [1] showed that if $\mathbf{d}$ is totally ω-c.a. then every $A \leqslant_T \mathbf{d}$ is wtt-reducible to a hypersimple set; but Barmpalias [9] showed that not every c.e. set is wtt-reducible to a hypersimple set.

We remark that weak truth-table reducibility is exactly the right kind of reducibility which gives non-trivial results in this context. This is because every non-zero c.e. Turing degree contains a hypersimple set and every c.e. set is Turing reducible to a ranked set; but if $A \leqslant_{\text{tt}} B$ and B is ranked then so is A.

4.3. **Higher up.** At present there are few examples of theorems whose combinatorics involve ordinals above ω. An important example involving lattice embeddings into the c.e. degrees will be discussed in Section Section 5. Here we discuss three others.

4.3.1. *Indifference for Cohen genericity.* Let P be a property of sets like e.g. genericity or randomness. For sequences A, B and a set $I \subseteq \omega$ of positions, we will write $A =_I B$ to mean that for all $x \notin I$, $A(x) = B(x)$. Following Figueira, Miller and

Nies, [41], a set I is called *indifferent* for a sequence A relative to P, if for all $B =_I A$, B has property P. When P is clear from the context, we say that I is indifferent for A.

This notion has been mainly investigated in the context of ML-randomness. For every ML-random sequence Z there is a set indifferent for Z (with respect to ML-randomness) [41]. Barmpalias, Lewis and Ng [13] used indifferent sets to allow coding in their proof that every PA degrees is the join of two random degrees.

On the category side, recall the notion of Cohen 1-generic sequences: these are the sequences that are sufficiently generic (with respect to Cohen forcing) to decide all Σ^0_1 statements. In terms of computability, these are the sequences which either meet or avoid any Σ^0_1 set of strings. Jockusch and Posner [46] proved that some 1-generic has an indifferent set (with respect to 1-genericity); in fact, every 1-generic set has one (Figueira, Miller and Nies (unpublished); Day [21]).

Perhaps surprisingly, some 1-generic sets can actually compute indifferent sets for themselves.

Theorem 4.6 (Day [21])**.** *Let $\mathbf{d}$ be a c.e. degree.*

(1) *If $\mathbf{d}$ is not totally ω^ω-c.a. then $\mathbf{d}$ computes a 1-generic sequence which can compute an indifferent set for itself.*
(2) *If $\mathbf{d}$ is totally ω-c.a. then $\mathbf{d}$ cannot compute a 1-generic sequence which can compute an indifferent set for itself.*

The reader might notice the rather large gap in the theorem above.

Question 4.7. *What is the correct level for constructing c.e. degrees which compute a 1-generic G which can compute an indifferent set for itself?*

Whilst we are mentioning indifferent sets, we mention the following which seems very hard.

Question 4.8. *Can any ML-random sequence compute an indifferent set for itself?*

Unlike indifferent sets for 1-generics, indifferent sets for ML-randoms have to compute $\varnothing'$.

4.3.2. *Variations on genericity.* Michael McInerney [57] has demonstrated other connections between genericity and our hierarchy of c.e. degrees. He studied notions of *multiple genricity* related to pb-genericity of Downey, Jockusch and Stob [32]. In particular, he defines a notion of ω-change genericity, a strengthening of pb-genericity. Now a Turing degree bounds a pb-generic sequence if and only if it is array noncomputable.

Theorem 4.9 (McInerney [57])**.** *A c.e. degree bounds an ω-change generic sequence if and only if it is not totally ω-c.a.*

Note though that the characterisation mentioned above of bounding pb-generics holds for all Turing degrees, not only the c.e. ones. Here we have a partial result, using the domination properties mentioned above in Section 3.2.

Theorem 4.10 (McInerney [57])**.** *Let $\mathbf{d}$ be a Turing degree.*

(1) *If $\mathbf{d}$ is not uniformly ω^2-c.a. dominated then $\mathbf{d}$ computes an ω-change generic sequence.*
(2) *If $\mathbf{d}$ is ω-c.a. dominated then it does not compute an ω-change generic sequence.*

Whether (1) can be improved to ω-c.a. domination remains open.

4.3.3. *m-topped degrees.* A c.e. degree $\mathbf{a}$ is called *m-topped* if $\mathbf{a}$ contains a c.e. set A such that for all c.e. sets $W \leqslant_T A$, $W \leqslant_m A$. That is, $\mathbf{a}$ contains a greatest c.e. m-degree (among all c.e. m-degrees in $\mathbf{a}$). Clearly $\varnothing'$ is an example of such a degree. Downey and Jockusch [30] showed that incomplete m-topped degrees exist, they are all low$_2$, and cannot be low. Later Downey and Shore [34] showed that every low$_2$ c.e. degree is bounded by an incomplete m-topped degree.

Theorem 4.11 ([25, 24])**.**

(1) *no totally $< \omega^\omega$-c.a. c.e. degree is m-topped.*
(2) *There is an m-topped, totally ω^ω-c.a. degree.*

The proofs for these results are rather complex and are therefore omitted. Note that the extra restriction regarding lowness means that bounding m-topped degrees is not possible purely in terms of our hierarchy.

Problem 4.12. *Classify the m-topped degrees, or the c.e. degrees bounding m-topped degrees.*

Being nonlow and not totally $< \omega^\omega$-c.a. seems a long shot.

5. Natural definability

Some of the constructions captured by two layers in our hierarchy yield objects, namely embedded lattices, that can be described in the c.e. degrees using the language of ordering. This shows that these two layers, the totally ω-c.a. degrees and the totally $< \omega^\omega$-c.a. degrees, are naturally definable in the structure of the c.e. degreees.

Natural definability results in degree theory are few. In terms of *abstract, general* results on definability, there has been significant success in recent years, culminating in the work of Nies, Shore and Slaman [61], where the following is proved.

Theorem 5.1 (Nies, Shore, Slaman [61])**.** *Any relation on the c.e. degrees, invariant under the double jump, is definable in the c.e. degrees if and only if it is definable in first order arithmetic.*

The proof of Theorem 5.1 involves interpreting the standard model of arithmetic in the structure of the c.e. degrees without parameters, and a definable map from degrees to indices (in the model) which preserves the double jump. The beauty of this result is that it gives at one time a definition of a large class of relations on the c.e. degrees. For example, it is used to show that the classes low$_n$ for $n \geqslant 2$ are definable (so are the high$_n$ for all $n \geqslant 1$; the case $n = 1$ needs an extra argument).

Theorem 5.1 has two shortcomings. One is the reliance on the invariance of the relation under the double jump. It follows that no set of c.e. degrees that contains some but not all low$_2$ degrees can be defined using the theorem; these are the kinds of sets we investigate here.

Another issue is that the definitions provided by the theorem are not *natural* definitions of objects in computability theory, as outlined by Shore [64]. Here we are thinking of results such as the following.

- A c.e. degree is promptly simple if and only if it is not cappable (Ambos-Spies, Jockusch, Shore, and Soare [4]).

- A c.e. degree is contiguous if and only if it is locally distributive (Downey and Lempp [33]) if and only if it is not the top of the pentagon (the non-modular, 5 element lattice N_5) (Ambos-Spies and Fejer [3]).
- A c.e. truth table degree is low$_2$ if and only if it has a minimal cover in the c.e. truth table degrees (Downey and Shore [34]).

5.1. **Lattice embeddings and critical triples.** Natural definitions are closely related to embeddings of finite lattices into the c.e. degrees; see for example Lempp and Lerman [53], Lempp, Lerman and Solomon [54], and Lerman [55]. The question of which finite lattices can be embedded into the c.e. degrees remains open. All distributive lattices can be embedded. The non-distributive lattices fall into two classes. Each non-distributive lattice contains either the pentagon, N_5; or the 1-3-1 lattice (also known as M_5 or M_3); see Fig. 1. The key diference between the two kinds of non-distributive lattices is the existence of a *critical triple*. In a lattice, a critical triple consists of three incomparable elements $\mathbf{a}_0, \mathbf{a}_1$ and $\mathbf{b}$ such that $\mathbf{a}_0 \vee \mathbf{b} = \mathbf{a}_1 \vee \mathbf{b}$ but $\mathbf{a}_0 \wedge \mathbf{a}_1 \leqslant \mathbf{b}$. We call $\mathbf{b}$ the *centre* of the critical triple. The middle three elements of the 1-3-1 lattice form a critical triple (with any of the elements serving as centre). The lattices which do not contain a critical triple are the join-semidistributive ones. It is known that critical triples present a serious impediment to embedding lattices into the c.e. degrees; for example, the lattice S_8 (Fig. 2) cannot be embedded (Lachlan and Soare [52]). However, the 1-3-1 itself can be embedded (Lachlan [50]). In particular in the c.e. degrees we can find critical triples. Note that the c.e. degrees do not always have meets; in an uppersemilattice, the definition of a critical triple does not require the meet to exist, rather we stipluate that any $\mathbf{c} \leqslant \mathbf{a}_0, \mathbf{a}_1$ must also be below the centre $\mathbf{b}$. A related concept is that of a *weak critical triple*, in which the meet condition is weakened to requiring the non-existence of any $\mathbf{c} \leqslant \mathbf{a}_0, \mathbf{a}_1$ such that $\mathbf{a}_0, \mathbf{a}_1 \leqslant \mathbf{b} \vee \mathbf{c}$.

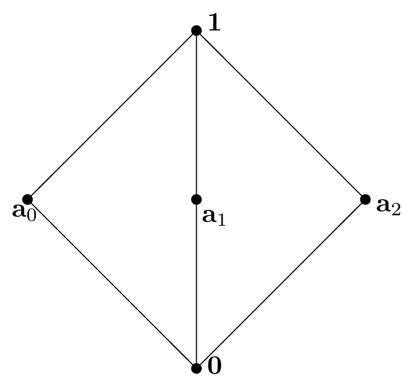

FIGURE 1. The 1-3-1 lattice

Bounding critical triples requires some computational power; Downey [23] showed that some nonzero c.e. degrees bound no critical triples; Weinstein [71] did the same for weak critical triples. On the other hand, some amount of permitting suffices. Downey noted that the embedding of critical triples seemed to be tied up with multiple permitting in a way that was similar to non-low$_2$-ness. Indeed this intuition was to some extent verified by Downey and Shore [35], who showed that any non-low$_2$ c.e. degree bounds a critical triple, indeed a copy of 1-3-1. The notion

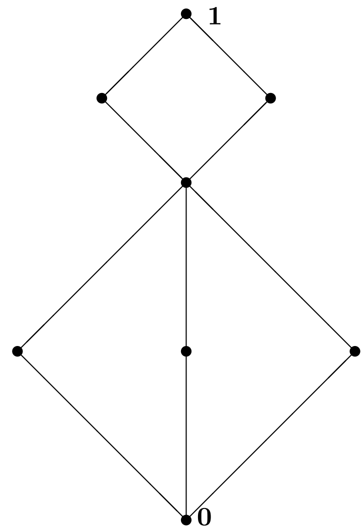

FIGURE 2. The lattice S_8

of non-low$_2$-ness seemed too strong to capture the class of degrees which bound the 1-3-1, but it was felt that something like that should suffice. On the other hand, Walk [69] showed that Weinstein's degree can be made array noncomputable, and hence it was already known that array non-computability was not enough for such embeddings.

The authors, together with Weber, showed the following.

Theorem 5.2 ([27]). *A c.e. degree bounds a critical triple if and only if it bounds a weak critical triple if and only if it is not totally ω-c.a.*

Thus, the totally ω-c.a. degrees are naturally definable in the c.e. degrees. Before we discuss the dynamics of this result, we observe that these results yield an answer to a question of Nies. Every superlow degree is array computable, but some low degrees are not totally ω-c.a. Hence:

Corollary 5.3 ([27]). *The low degrees and the superlow degrees are not elementarily equivalent.*

5.1.1. *Embedding critical triples: tracing and permitting.* Let us consider how to construct critical triples in the c.e. degrees. We wish to enumerate c.e. sets A_0, A_1 and B whose degrees will form the required triple. For the join requirements, we need to ensure that $A_0 \leqslant_T A_1 \oplus B$ and $A_1 \leqslant_T A_0 \oplus B$. To make the embedding non-trivial, we need to ensure that $A_0, A_1 \nleqslant_T B$.

The latter is done by usual Friedberg requirements. Say we want to ensure that $\Phi(B) \neq A_0$ for some Turing reduction Φ. We pick a witness (a *follower*) x and wait for $\Phi(B, x)\downarrow$. While we wait, we need to maintain a reduction of A_0 to $A_1 \oplus B$. This means that we promise that if we ever enumerate x into A_0, we will put another number x_1, called a *trace* for x, into either A_1 or B. But since we do not yet know the use $\varphi(B, x)$ of the potential computation $\Phi(B, x)$, it would be a bad idea to target x_1 for B. Now in turn, to maintain $A_1 \leqslant_T A_0 \oplus B$, we need to appoint an even larger number x_2, a trace for x_1, and promise to put it into A_0 if x_1 goes into A_1. And so we get a sequence of traces $x, x_1, x_2, \dots$ which keeps growing until we see $\Phi(B, x)\downarrow$. When we finally see the use, the next trace can be targeted for B, and then we can stop.

Now we would like to put all these numbers $x, x_1, x_2, \dots, x_k$ into the sets for which they are targeted. But we cannot do this all at once. For we also have minimal-pair-like requirements for the pseudo-meet: if $\Psi_0(A_0) = \Psi_1(A_1)$ equals some set C, then $C \leqslant_{\mathrm{T}} B$. Lachlan's strategy for meeting such a requirement is to allow, during an "expansionary" stage, to enumerate numbers into A_0 or A_1, but not both. This means that the computation giving one side of the observed agreement is fixed; we then wait for new agreement (the next expansionary stage; using the mechanism of pinball machines, we wait for a *gate* to open), and then again allow only one side of the computation to be destroyed.

We thus take our *entourage* of traces $x, x_1, \dots, x_k$ and "peel it back". Rename the last trace x_k to be b, to indicate that it is targeted for B. At an expansionary stage (for Ψ_0, Ψ_1, a minimality requirement stronger than $\Phi(B) \neq A_0$), we enumerate x_{k-1} into the set it is targeted for (A_i for $i = (k-1) \mod 2$) and b into B. We then appoint a new trace b' for x_{k-2}, this time targeted for B. At the next expansionary stage we enumerate x_{k-2} and b' into their target sets, and appoint a new B-trace for x_{k-3}. After k many such steps, we get to enumerate the original follower $x = x_0$ into A_0 and meet the requirement.

That is the sketch of the construction; of course in the general construction we need to deal with several requirements, and so we coordinate their action on a tree, or a pinball machine.

Now let us consider what happens when we want to build A_0, A_1 and B all below a given c.e. degree $\mathbf{d}$. The extra condition now is that whenever we enumerate any number into any of these sets, we need to obtain permission from $\mathbf{d}$ to do so. As in the proof of Theorem 4.3, this permission comes in the form of a change in the approximation of some function $g \in \mathbf{d}$ on an associated input, essentially the use of reducing A_0, A_1, B to g. And in the context of multiple permitting, we need to analyse *how many times* does a single positive requirement (of the kind $\Phi(B) \neq A_0$) need to receive permission in order for it to be met. If we can tell in advance, for each requirement, how many permissions we need, then this can be done below any array noncomputable degree. But here we see that the number k above of enumerations the requirement needed could not be computed in advance: we know it once we see $\Phi(B, x)$ converge but not before. Giving such permissions is the extra strength of non totally ω-c.a. degrees.

5.1.2. *Non-embedding of critical triples: certification.* The flip-side of this argument is adapting the construction of a degree which does not bound a critical triple and showing that it can be applied to any totally ω-c.a. degree.

Here we are given three incomparable sets A_0, A_1 and B, all below some c.e. set D whose degree is totally ω-c.a.; We assume that $A_i \leqslant_{\mathrm{T}} B \oplus A_{1-i}$ for $i = 0, 1$; our aim is to build some set $Q \leqslant_{\mathrm{T}} A_0, A_1$ but $Q \nleqslant_{\mathrm{T}} B$. To meet the latter, a typical requirement $\Psi(B) \neq Q$ appoints a follower x, waits for realisation, namely $\Psi(B, x)\downarrow = 0$, and then, remembering that we keep $Q \leqslant_{\mathrm{T}} A_0, A_1$, hopes for a *double change*, in A_0 and in A_1, that allows us to enumerate x into Q. Now there are two questions:

- Why would we get such a double change?

- If we do, how do we keep the computation $\Psi(B, x) = 0$ valid?

The overall idea is for us to define a function $g = \Lambda(D)$, and non-uniformly guess an ω-computable approximation $\langle g_s \rangle$ for g. The use of computing $g(n) = \Lambda(D, n)$ from D will bound the use of reducing a certain amount of A_0, A_1 and B to D. If the configuration of sets changes, this necessitates a change in D below that use; this allows us to redefine $\Lambda(D, n)$ with a new value. We then wait for the approximation $\langle g_s(n) \rangle$ to catch up and correctly guess the new value. This gives us a certification of sorts that D, up to the use, is correct. Since we know a bound on the number of changes on $\langle g_s(n) \rangle$, we know a bound on how many times a desirable configuration will be destroyed.

In the current situation, a basic idea is that of setting up layers to protect a computation. We know that $A_i \leqslant_{\mathrm{T}} B \oplus A_{1-i}$. This allows us, for any number x, to calculate an increasing sequence of numbers $x^{(1)}, x^{(2)}, \ldots$ such that any change in some A_i below $x^{(m)}$ necessitates a change in either B or A_{1-i} below $x^{(m+1)}$. See Fig. 3.

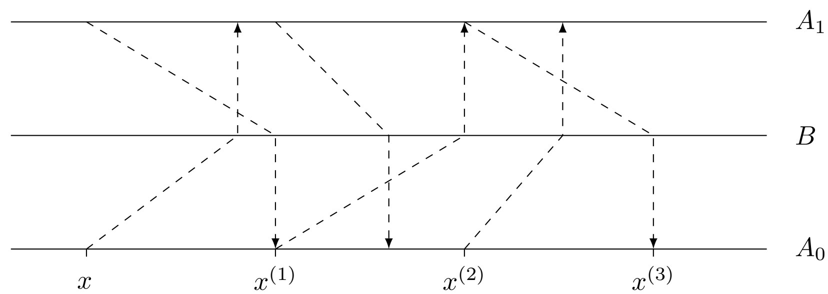

FIGURE 3. Three layers.

We now act as follows. Fixing n, we calculate a bound m on the number of changes in $g_s(n)$. We set up $m + 1$ many layers. The use for reducing $Q(x)$ to A_0, A_1 is $x^{(m+1)}$. After x is realised, two things can happen. If we get a double change (a change in both A_0 and A_1), we can enumerate x into Q and then meet the requirement. Otherwise, we claim that provided $B \restriction_{x^{(m+1)}}$ never changes, $A_0 \restriction_x$ is correct too. The point is that if $B \restriction_{x^{(m+1)}}$ is correct, then the next change in some A_i must happen above $x^{(m)}$: otherwise either a B-change or an A_{1-i}-change is guaranteed. After this happens, we will have lost a layer; but since we set the use of $\Lambda(D, n)$ to be sufficiently large, this peeling of the last layer allows us to change $\Lambda(D, n)$. The next change will happen above $x^{(m-1)}$, and so on. Overall, we see that if the requirement is never met (over infinitely many independent attempts to meet it), then $A_0 \leqslant_{\mathrm{T}} B$, which we assumed is not the case.

This sketch is necessarily rough, as there are several other delicate points to the argument. For details see [24].

5.2. **The 1-3-1.** What about embeddings of the 1-3-1 lattice itself? We note that previously, it was not known that there is a difference between bounding a copy of 1-3-1 in the c.e. degrees and bounding a critical triple. Our hierarchy results, together with Theorem 5.2 and the following result, show these are not equivalent.

Theorem 5.4 ([24]). *A c.e. degree bounds a copy of the 1-3-1 lattice if and only if it is not totally $<\omega^\omega$-c.a.*

Again, this shows that the totally $<\omega^\omega$-c.a. degrees are naturally definable.

5.2.1. *Embedding the 1-3-1 lattice.* We now wish to embed the 1-3-1 lattice into the c.e. degrees. How is the construction different from that of a critical triple? We now build A_0, A_1, A_2 and need to ensure that $A_i \leqslant_{\mathrm{T}} A_j \oplus A_k$ for $\{i, j, k\} = \{0, 1, 2\}$. This means that every number targeted for some A_i needs a trace targeted for either A_j or A_k. So even though the non-triviality requirements are just $A_i \neq \varphi$ for a partial computable function φ, without an oracle B this time, we still need the continuous tracing: at each stage we add another trace to the end of the entourage.

Something interesting starts happening when we consider two minimal-pair requirements, occupying two *gates* in a pinball machine used in this construction. An entourage consisting of numbers $x_0^0, x_1^1, x_2^0, x_3^1, x_4^0, x_5^1, \dots, x_m^0$ (with x_k^i targeted for A_i) arrives at a gate which works toward showing that A_0 and A_1 form a minimal pair. For a while the entourage waits for the gate to open. The gate will not allow numbers targeted for A_0 and numbers targeted for A_1 to pass at the same time. While the entourage waits for the gate to open, new traces must continually be added. We have a choice, though: a trace for x_k^0 can be targeted for either A_1 or A_2. So the rest of the entourage from x_m^0 onwards is targeted for $A_0, A_2, A_0, A_2, \dots$.

The gate opens, allowing the rest of the entourage: $x_m^0, x_{m+1}^2, x_{m+2}^0, \dots, x_k^2$ to pass. This second part of the entourage now arrives at a closed gate (of stronger priority), working for a minimal-pair requirement for A_0 and A_2. Again we redirect our targeting and target new traces to A_1 and A_2. When the gate opens, x_k^2 and the third part of the entourage pass and get enumerated into their sets. We then are left with x_{k-1}^0, to which we add a (A_0, A_1)-entourage, and repeat. After $(k-m)$ many such steps, all of the traces waiting at the lower (stronger) gate have been dealt with, and the entourage now consists of $x_0^0, x_1^1, \dots, x_{m-1}^1$. After appointing more traces, x_{m-1}^1 and the rest of the entourage passes the upper gate, lodges at the lower gate, and the process repeats.

How many permissions do we need? We see that the answer, in the case of 2 gates, is "roughly ω^2 many". When x_0^0, the original follower, is realised, we find out the number m. And then, for each $l < m$, when x_l passes the top gate, we know the length of the entourage at that time (roughly, the stage number). We will update the number of required permissions m times. And in general, to pass n gates we need ω^n-permission. If the function giving permissions to the entire construction is not ω^n-c.a. for any n, then the construction will work.

Again, we skip many important details; for example, why the bottom of the embedding cannot actually be $\mathbf{0}$, and a more careful analysis, involving more than one follower, which shows that actually we need ω^{2n} to deal with n gates; all is explained in [24].

5.2.2. *Non-embedding the 1-3-1.* How do we show, for example, that a totally ω^2-c.a. degree $\mathbf{d}$ does not bound a copy of the 1-3-1 lattice? After all, it may bound a critical triple. As pointed out above, any of the middle elements of the 1-3-1 lattice can serve as the centre of a critical triple, together with the other two elements. The plan is now, given B_0, B_1 and A below $\mathbf{d}$, to show that either B_0 is not the centre of a critical triple, flanked by B_1 and A; *or* B_1 is not the centre of a critical triple, flanked by B_0 and A. This adds a level of non-uniformity to the construction. We build $Q \leqslant_{\mathrm{T}} A, B_0$, and try to ensure that $Q \nleqslant_{\mathrm{T}} B_1$. We might fail. For each $e < \omega$ (representing a potential failure, i.e. $\Phi_e(B_1) = Q$), we build another c.e. set Q^e, this time reducible to A and B_1, and try to ensure that $Q^e \nleqslant_{\mathrm{T}} B_0$. Somewhere we must succeed, or we will have shown that $B_0 \leqslant_{\mathrm{T}} B_1$. Each instance will now consider two followers (for Q and for Q^e), and will set up two levels of layers, inner layers for Q^e and outer layers for Q. The inner layers reflect the coefficient of ω in an ordinal below ω^2 which bounds the number of changes in $g_s(n)$; when one inner layer is peeled (the number of changes left drops below a limit ordinal), we get a new constant coefficient, which tells us how many new outer layers we need to set up. If we guess that $g = \Lambda(D)$ is ω^m-c.a., then we need m steps of non-uniformity, constructing Q, Q^{e_1}, $Q^{e_1,e_2}, \dots,$ $Q^{e_1,e_2,\dots,e_{m-1}}$; at each step we alternate between treating B_0 or B_1 as the centre.

5.3. **A Question.**

Question 5.5. *Is there an $n > 1$ such that being totally ω^n-c.a. is (naturally) definable?*

We remark that if the methods we have used can also be used to answer this, what would be needed would be a lattice or partial ordering whose embedding needed "n-gates". For example, something that might define totally ω^2-c.a. could be a structure whose embedding needed 2 gates and could not be done with 1. In spite of concerted efforts on our part, we have not been able to find such a structure. On the other hand if natural is left off definability then perhaps there is some more delicate way to use the methods of Nies, Shore and Slaman [61], which may involve strong reducibilities. The reader should treat this last paragraph as speculation on our part.

6. Weak truth table triples

Downey and Stob [37] observed that there seemed to be a connection between lattice embeddings and the structure of the c.e. weak truth table degrees within a c.e. Turing degree. To wit, they showed that if a c.e. Turing degree $\mathbf{a}$ is the top of a 1-4-1 lattice with bottom degree $\mathbf{0}$ then $\mathbf{a}$ contains a pair of c.e. sets A_1 and A_2 such that $\deg_{\mathrm{wtt}}(A_1)$ and $\deg_{\mathrm{wtt}}(A_2)$ form a minimal pair. In fact, the original proof of the construction of a pair of noncomputable c.e. sets $A_1 \equiv_T A_2$ forming a wtt-minimal pair was a direct one, and it was only when the authors noticed that the combinatorics of the construction were similar to the embedding of 1-3-1 that the proof using 1-4-1 was found.

Mimicking critical triples, in [27] we gave the following definition.

Definition 6.1. Three c.e. sets A_0, A_1 and B form a *wtt triple* if $A_0 \equiv_{\mathrm{T}} A_1$, $A_i \nleqslant_{\mathrm{T}} B$, and for all $C \leqslant_{\mathrm{wtt}} A_0, A_1$ we have $C \leqslant_{\mathrm{wtt}} B$.

An analogue of weak critical triples was also discussed.

Theorem 6.2 ([27])**.** *A c.e. degree* $\mathbf{d}$ *is not totally ω-c.a. if and only if there are $A_0, A_1, B \leqslant_{\mathrm{T}} \mathbf{d}$ which form a wtt-triple.*

In fact, this theorem can be improved. Going back to the original idea of wtt minimal pairs inside a Turing degree, we strengthen the notion of a wtt triple.

Definition 6.3. Three c.e. sets A_0, A_1 and B form a *wtt infing triple* if $A_0 \equiv_{\mathrm{T}} A_1$, $B <_{\mathrm{T}} A_0, A_1$, and

$$\deg_{\mathrm{wtt}}(A_0) \wedge \deg_{\mathrm{wtt}}(A_1) = \deg_{\mathrm{wtt}}(B).$$

The following appears here for the first time.

Theorem 6.4. *Every c.e. degree which is not totally ω-c.a. bounds three c.e. sets which form a wtt infing triple.*

It follows of course that bounding wtt triple and bounding wtt infinig triples are both equivalent to being not totally ω-c.a.

Proof. The proof is a modification of the permitting direction of Theorem 6.2. We give details here as this theorem is new.

We are given a c.e. degree which contains a function g which is not ω-c.a. We have a computable approximation $\langle g_s \rangle$ of g and we may assume that g itself is the modulus for this approximation: $g_s(n)$ is the least $t \leqslant s$ such that $g_r \upharpoonright_n$ is stable for $r \in [t, s]$.

We enumerate three c.e. sets A_0, A_1 and B, and ensure that they are all computable from g. The wtt infinig triple will consist of $A_0 \oplus B$, $A_1 \oplus B$ and B. Thus we need to ensure that $A_i \nleqslant_{\mathrm{T}} B$ and $A_i \leqslant_{\mathrm{T}} B \oplus A_{1-i}$; and of course the inf requirements:

N_Δ**:** If $\Delta(A_0, B) = \Delta(A_1, B)$ is total then it is wtt-reducible to B.

Here Δ ranges over all wtt functionals; we denote by δ the (possibly partial) computable bound on the use of Δ. To ensure that $A_i \nleqslant_{\mathrm{T}} B$ we meet requirements

$P_{\Phi,i}$**:** $\Phi(B) \neq A_i$,

where now Φ ranges over the Turing functionals.

To meet the negative requirements we use a tree of strategies. Nodes working for negative requirements N_e have two possible outcomes on the tree, ∞ and `fin`, with ∞ stronger. Nodes working for positive requirements only have one outcome.

To keep track of our reduction of these sets to g, we define moving markers $a_{x,s} < \omega$. The rules for the markers are that if $g_{s+1} \upharpoonright_{a_{x,s}} = g_s \upharpoonright_{a_{s,x}}$ then we must have $a_{x,s+1} = a_{x,s}$. Otherwise we are allowed to increase it. We increase it to “take over space” from cancelled (weaker) followers for the same strategy.

Here x denotes a potential follower for a strategy (node) σ working for a positive requirement $P_{\Phi,i}$. While a follower is waiting for *realisation*, namely $\Phi(B, x){\downarrow} = 0$, we keep appointing traces, gradually building an *entourage* of traces $x_0, x_1, x_2, \ldots$ with $x = x_0$ being the follower, and the targeting of followers alternating between A_i and A_{1-i}. When the follower is realised and then σ is visited, we stop this process, and instead appoint a last trace targeted for B. After that we aready to start enumerating traces into the sets for which they are targeted. When σ wants to enumerate traces, we say that we put them into a *permitting bin*. We then wait for the follower x to be *permitted*, which means that $g_{s+1} \upharpoonright_{a_{x,s}} \neq g_s \upharpoonright_{a_{x,s}}$.

Construction.

At any stage of the construction there are two options, depending on whether some traces in the permitting bin are permitted.

Option A: permission

If there is a pair x_k, b_k of traces of an entourage for some follower $x = x_0$ which are currently waiting at the permitting bin and are realised, then we choose the strongest such follower. We then:

(1) Enumerate the two traces into their target sets.
(2) If $k = 0$ (the follower has just been enumerated) then we declare the node σ which appointed the follower *satisfied*, and cancel all of its other followers.
(3) In that case, we cancel all followers for σ whcih are weaker than x; we redefine $a_{x,s+1} = s + 1$.
(4) In either case we initialise all nodes weaker than σ. This causes the cacellation of the followers of all these nodes; and none of these nodes is now satisfied.

We then end the stage.

Option B: no permission
We construct the path of nodes accessible at stage s.

First, suppose that a node τ which works for a negative requirement N_Δ is accessible at stage s. Let $\ell(\Delta, s)$ be the length of agreement between $\Delta(A_0, B)$ and $\Delta(A_1, B)$. Let t be the last $\tau\hat{}\infty$-stage (also known as a τ-expansionary stage); $t = 0$ if there was no such stage. If $\ell(\Delta e, s) > t$ then we let $\tau\hat{}\infty$ be next accessible. Otherwise, we let $\tau\hat{}\mathtt{fin}$ be next accessible.

Next, suppose that a node σ which works for a positive requirement $P_{\Phi,i}$ is accessible at stage s. If σ is satisfied, or if it has an unrealised follower, then σ does nothing and its only child is accessible. Otherwise:

- If all followers for σ have some traces waiting in the permitting bin (this includes the case that there are no followers appointed), then σ appoints a new (large) follower x, and defines $a_{x,s+1} = s + 1$.
- Suppose that there is a follower x that is realised, but no elements of x's entourage lie in the permitting bin (necessarily x will be σ's weakest follower.) Let the entourage be $x = x_0, x_1, \ldots, x_m$.
  - If this is the first time at which σ is accessible and x is realised, then we reserve a set of potential traces $b_0, b_1, \ldots, b_m$ (consisting of large numbers).

  We then drop the pair x_m, b_m into the permitting bin.

In either case, all weaker nodes are initialised and the stage is ended.

At the end of the stage, for any yet uncancelled and unrealised follower x with an entourage $x = x_0 \ldots, x_m$ (so with no B-traces b_k defined yet), we appoint a new, large trace x_{m+1}, targeted for the set A_i for which x_m is not targeted.

Verification.
First, let us observe that $A_i \leqslant_{\mathrm{T}} B \oplus A_{1-i}$. Let $y < \omega$. To see if $y \in A_i$, we look at stage y to see if $y = x_k$ is currently an element of some entourage $x_0, \ldots,$ and is targeted for A_i. If not, then $y \in A_i \iff y \in A_{i,y}$. If so, at stage y, y has a trace, either x_{m+1} or b_m. In the latter case, $y \in A_i \iff b_m \in B$. Suppose otherwise. If $x_{m+1} \notin A_{1-i}$ then $y \notin A_i$. If $x_{m+1} \in A_{1-i}$ then we can find the stage t at which z is enumerated into A_{1-i}. By that stage, the number b_m has been defined and we are back in the previous case.

Next, let us observe that all three sets A_0, A_1 and B are computable from g. For let y be any number. To see if y will enter any of these sets, we first go to stage y and see if $y = x_m$ or $y = b_m$ for some entourage which is already present at stage y. Again, if not then at that stage we can see y's fate. Otherwise, let x be the follower, the first number in that entourage. We observe that the marker $a_{x,s}$ can only be updated finitely many times: the number of times is bounded by the length of the entourage when x is realised. The rules of the markers show that g can find a stage t such that $g_t\restriction_{a_{x,t}}$ will never change, or by which time x was cancelled. Then y is enumerated into its target set if and only if this has happened by stage t.

Now let us consider positive requirements. By induction on the length of nodes, we show that the true path is infinite and that nodes on the true path act only finitely many times; and all positive requirements are met. Note first that option A cannot be taken cofinally many times; eventually we would run out of pairs in the permitting bin. So it remains to show that if a node σ, working for $P_{\Phi,i}$, is accessible infinitely often, but is not initialised infinitely often, then σ acts only finitely often.

It is a standard argument to show that if σ is satisfied (after the last stage at which it is initialised) then the requirement is met. The point as always is that the markers $b_0, \dots, b_m$ are chosen late and so are greater than the use $\phi(B, x)$.

We may assume that every follower that σ appoints is later either realised or cancelled. Every follower receives attention only finitely often. Suppose that σ is never satisfied (after the last stage it is initialised). Then σ has infinitely many followers that are never cancelled (for every stage, consider the strongest follower ever to receive attention after that stage). These followers eventually have traces stuck for ever in the permitting bin.

Under these assumptions, we argue that g is ω-c.a.

Let $n < \omega$. To approximate $g(n)$ we pick out stages during which we believe $g_s(n)$. Let r^* be the last stage at which σ is initialised. Find a stage $s^* = s^*(n) > r^*$ at which σ is accessible and has a follower $x^* = x^*(n)$ which is already realised and has some traces waiting at the permitting bin. We then let $S(n)$ be the set of stages $s > s^*$ at which σ is accessible. Suppose that $s < t$ are successive stages in $S(n)$; suppose that $g_s(n) \neq g_t(n)$. Let $x \leqslant x^*$ be the strongest follower for σ that receieved attention since stage s^* ($x = x^*$ if there is no such x). Then at the end of stage s, traces from x's entourage are waiting at the permitting bin, and $a_{x,s} > n$. Thus, between stages s and t, x, or a stronger follower for σ, is permitted, and traces are enumerated into sets. The number of times this can happen is bounded by the sum of the lengths of the entourages of all followers $y \leqslant x^*$ at stage s^* (essentially, bounded by $(s^*)^2$). Thus, restricting to the stages in $S(n)$, we can effectively put a bound on the number of changes in $g_s(n)$.

It remains to show that every negative requirement N_e is met. It is here where the construction diverges from that of a critical triple. Fix a wtt functional Δ, and let τ be the node on the true path which works for N_Δ; we assume that $\Delta(A_0, B) = \Delta(A_1, B) = Z$ is total. We know that $\tau\hat{}\infty$ also lies on the true path. Let r^* be a stage after which τ is never initialised.

Given $n < \omega$, let $s^* = s^*(n)$ be the least $\tau\hat{}\infty$-stage $s > r^*$ at which the length of agreement $\ell(\Delta, s)$ is greater than n. By convention, we assume that s^* bounds the size of all of the B-traces for all follower which have been appointed by stage s^*, and of course also the use $\delta(n)$. Let $t^* = t^*(n) \geqslant s^*$ be a $\tau\hat{}\infty$-stage at which

$B\restriction_{s*}$ is correct. We claim that $Z(n) = \Delta(A_i, B, n)[s]$. Note that $n \mapsto s^*(n)$ is computable, so this is a wtt reduction.

The basis of the argument is of course Lachlan's minimal pair argument of preserving one side of the computation from one expansionary stage to the next. We claim that if $s < t$ are successive $\tau\hat{}\infty$-stages, with $s \geqslant t^*$, then it cannot be that numbers y_0 and y_1, both smaller than $\delta(n)$, enter A_0 and A_1 respectively, both between stages s and t. Suppose for a contradiction that this happens. When and where do these traces originate? They have to have been appointed before stage s^*, and so they belong to followers x_0 and x_1 which belog to nodes extending $\tau\hat{}\infty$. On the other hand, they were not in the permitting bin at stage s^*, since then their B-trace would be smaller than s^*, and as $s \geqslant t^*$, that trace not being in B would mean that they cannot be enumerated into the sets A_i. Also, $x_0 \neq x_1$, as no two traces for the same follower for a node extending $\tau\hat{}\infty$ can be in the permitting bin between two successive $\tau\hat{}\infty$-stages.

Assume, without loss of generality, that x_0 is stronger than x_1. This means that x_1 was appointed after the $\tau\hat{}\infty$-stage t_0 at which y_0 was placed in the permitting bin. But we just argued that $t_0 \geqslant s^*$, which in turn implies that $y_1 > x_1 > \delta(n)$, a contradiction. □

Remark 6.5. Note that this proof used the special features of wtt reducibility. It is not the case that if Δ were a Turing functional, then we could argue that $Z \leqslant_{\mathrm{T}} B$. Once successive A_0 and A_1 changes occur, the use of a Turing computation goes up, and then a B-change, too large to be previously comprehended even by B itself, destroys both sides of the computation. This is why we cannot prove the existence of a paradoxical critical triple: Turing degrees $\mathbf{a}_0, \mathbf{a}_1$ and $\mathbf{b}$ such that $\mathbf{a}_0 \vee \mathbf{b} = \mathbf{a}_1 \vee \mathbf{b}$, but $\mathbf{b} < \mathbf{a}_0, \mathbf{a}_1$ and $\mathbf{b} = \mathbf{a}_0 \wedge \mathbf{a}_1$.

On the other hand, note that when we embed the 1-3-1 lattice, it seems that we exactly do run this hypothetical, paradoxical construction using Turing reductions. The difference, the reason it works, is the existence of the third set A_2. This third set allows us to retarget traces not to B but to A_2. This means that B-traces are now appointed only when traces already arrive in the permitting bin, not immeidately after the last trace of the entourage is enumerated (and we are still waiting for the gate to open). This allows B to catch its own tail and correctly certify computations.

7. Maximality in the new hierarchy

Remarkably, it turns out that the hierarchy we introduced gives new non-continuity results in the c.e. degrees.

Definition 7.1. We say that $\mathbf{a}$ has *maximal* totally α-c.a. degree if

- $\mathbf{a}$ is totally α-c.a., and
- For all $\mathbf{b} > \mathbf{a}$, $\mathbf{b}$ is not totally α-c.a.

Cholak, Downey and Walk [17] constructed maximal *contiguous* degrees. This result hints at the following.

Theorem 7.2 ([24])**.** *Let $\alpha < \varepsilon_0$ be a power of ω. There exists a maximal totally α-c.a. degree. Indeed, there is such a degree which is uniformly totally α-c.a.*

On the other hand, maximality has its limits. No degree is maximal for the next level:

Theorem 7.3 ([24])**.** *Let $\beta < \varepsilon_0$. For any c.e. degree* $\mathbf{a}$ *which is totally ω^β-c.a. there is a c.e. degree* $\mathbf{b} > \mathbf{a}$ *which is totally $\omega^{\beta+1}$-c.a.*

And limit classes have no maximal elements:

Theorem 7.4 ([24])**.** *If $\alpha < \varepsilon_0$ is a limit of powers of ω, then there is no maximal $< \alpha$-c.a. degree.*

Applying Theorem 7.2 to $\alpha = \omega$, and Theorem 7.4 to $\alpha = \omega^\omega$, yields:

Corollary 7.5. *There are c.e. degrees, maximal with respect to not bounding critical triples; but there are no degrees which are maximal with respect to not bounding copies of the 1-3-1 lattice.*

Our definability results show:

Corollary 7.6. *The maximal totally ω-c.a. degrees form a naturally definable antichain in the c.e. degrees.*

Standard lower-cone avoiding techniques show that this antichain is infinite.

Sketch of proof of Theorem 7.2. We enumerate a c.e. set D whose degree should be maximal totally α-c.a. As stated, by a small modification we can in fact make $\deg_{\mathrm{T}}(D)$ uniformly totally α-c.a. computable, but we do not discuss this here.

To understand the construction, first think of what goes wrong if we try to make D both totally α-c.a. and not totally α-c.a. For the former, we meet negative requirements Q_Φ which measure the length of convergence of $\Phi(D)$ and at various stages "certify" observed computations $\Phi(D,x)$. These requirements live on a tree of strategies and have Σ_2/Π_2 outcomes.

For the latter, we build a functional Λ and try to diagonalise $\Lambda(D)$ against all possible α-c.a. functions, which we can list. If $\langle f_s, o_s\rangle$ is an α-computable approximation, then a requirement $P_{f,o}$ appoints a follower p, and each time we observe that $\Lambda(D,p) = f_s(p)$, it enumerates the use $\lambda(p)$ into D, redefines $\Lambda(D,p)$ to have large value, and repeats. (In the language of one of the authors, we "beat $f_s(p)$ to death".) It will not need to do this more than $o_0(p)$ "many times". (As usual, "number of times" is in terms of ordinal counts; this term is literally correct in the case $\alpha = \omega$.)

So what indeed does go wrong? The issue is of uniformity and timing. We cannot effectively enumerate all α-c.a. functions, each with a *total* α-computable approximation (exactly as we cannot list all total computable functions). In the arguments above (Theorem 3.2), we used $(\alpha+1)$-computable approximations. These are essentially potentially partial approximations: we wait for $o_0(p)\downarrow$. If it never converges then we needn't do a thing. When it does converge we can start the diagonalisation process.

Now's the timing problem: a Q-requirement stronger than the P-requirement tries to certify some computation $\Phi(D,x)$ and give some bound on the number of changes it will allow, i.e., the number of times it anticipates weaker requirements, such as P, will injure a certified computation by their positive action. To do so, we need to see the value $o_0(p)$. But it is possible that P has apointed p but is still waiting to see $o_0(p)$ converge (in other words, to see $o_s(p) < \alpha$). On the other hand, upon appointing p, P must declare a use $\lambda(p)$; regardless of anything, we need to ensure that $\Lambda(D)$ is total. Really, what we would like is to only define $\lambda(p)$ when $o_0(p)\downarrow$, or at least to be able to change the use $\lambda(p)$ to something large at that stage — sufficiently large so that P-action will not affect any certified computations, whose use would be smaller.

Well, we can't, and therefore mathematics is still apparently consistent. But essentially, we do get our wish when we construct a maximal totally α-c.a. degree.

In this construction, for each c.e. set W we define a functional Λ_W, and the aim is to show one of the two: either $\Lambda(D, W)$ is not ω-c.a. (and so $D \oplus W$ is not totally ω-c.a.); or $W \leqslant_{\mathrm{T}} D$. A typical positive requirement $P_{W,(f,o)}$ appoints a follower and tries to show that $\Lambda_W(D, W, p) \neq f(p)$. What we do, once we see that $o_0(p)\downarrow$, is wait for a future change in W below the use $\lambda(p)$. This future change allows us to lift the use $\lambda(p)$ to something large; then we can start diagonalising against $f(p)$ by enumerating uses into D. $\Phi(D, x)$ computations certified prior to the W-change are protected from this diagonalisation action, because their uses are smaller than the new $\lambda(p)$. Computations which are certified later, have already seen the value of $o_0(p)$ and can take into account how many times P will act. If, on the other hand, we keep appointing followers p but never get the W-change that we ask for, then we show that W is computable from D.

There are some several other delicate issues, such as explaining why we need D to compute W – it would appear that no changes would make W computable; the problem is timing of permissions, compared to when nodes are accessible, and which computations we see at such stages. It turns out that during verification, this complication means that we first have to show that D is low_2, and only after can we show that positive requirements are met, and that D is totally α-c.a. $\square$

Recent work of the authers together with Katherine Arthur [8, 7] has explored the relationship of maximal α-c.a. degrees and the rest of the hierarchy. One basic question is understanding what happens to our hierarchy when we restrict it to upper cones. Say a degree is *properly* totally β-c.a. if it is totally β-c.a., but not totally γ-c.a. for any $\gamma < \beta$.

Question 7.7. *Let $\alpha < \beta < \varepsilon_0$ be powers of ω. Let $\mathbf{a}$ be a totally α-c.a. degree. Must there be a degree $\mathbf{b} > \mathbf{a}$ which is properly totally β-c.a.?*

In light of Theorem 7.3, this question is related to the question of bounding by maximal degrees. Namely, suppose that α, β and $\mathbf{a}$ witness the failure of an instance of Question 7.7: say that no $\mathbf{b} \geqslant \mathbf{a}$ is properly totally β-c.a. Then $\mathbf{a}$ is bounded by no maximal totally α-c.a. degree (and in fact, by no maximal totally γ-c.a. degree for any $\gamma \in [\alpha, \beta]$). This motivates the following.

Question 7.8. *For which pairs $\alpha < \beta < \varepsilon_0$ of powers of ω is it the case that every totally α-c.a. degree is bounded by a maximal totally α-c.a. degree?*

Here we have partial answers.

Theorem 7.9 (Arthur, Downey and Greenberg [7, 8])**.** *Let $\alpha < \varepsilon_0$ be a power of ω.*

1. *There are totally α-c.a. degrees which are not bounded by any maximal totally α-c.a. degrees.*
2. *For any $\beta \geqslant \alpha^\omega$ which is a power of ω, every totally α-c.a. degree is bounded by a maximal totally β-c.a. degree.*

We note that there is quite a gap there. The following would be nice.

Conjecture 7.10. *There is a totally ω-c.a. degree which is bounded by no maximal totally ω^n-c.a. degree, for any $n < \omega$.*

Theorem 7.9(2) implies that if $\beta \geqslant \alpha^\omega$, then every totally α-c.a. degree is bounded by a properly totally β-c.a. degree. The question remains open for lower levels; note that in Theorem 7.3, we cannot ensure that $\mathbf{b} > \mathbf{a}$ is properly totally $\omega^{\beta+1}$-c.a. Again we have a partial result:

Theorem 7.11 (Arthur, Downey and Greenberg [7, 8])**.** *Every totally ω-c.a. degree is bounded by a totally ω^3-c.a. degree which is not totally ω-c.a.*

We do not know whether the degree constructed for Theorem 7.11 is totally ω^2-c.a. or not. Note that this result shows that a naive plan for resolving Conjecture 7.10 cannot work. That plan would construct a totally ω-c.a. degree $\mathbf{a}$ such that for all n, every totally ω^n-c.a. degree $\mathbf{b} \geqslant \mathbf{a}$ is totally ω-c.a.

Further results concern degrees which are maximal not only withrespect to themselves, but with respect to smaller degrees:

Theorem 7.12 ([7, 8])**.** *There are c.e. degrees $\mathbf{a} < \mathbf{b}$ such that $\mathbf{b}$ is totally ω-c.a., and such that if $\mathbf{c} \geqslant \mathbf{a}$ is totally ω-c.a., then $\mathbf{c} \leqslant \mathbf{b}$.*

This is related to the proof of Theorem 7.9(1). Indeed, paradoxically, the proof of this theorem is made by adapting the *construction* of a maximal totally ω-c.a. degree and creating a "maximal ideal": a sequence $\mathbf{a}_0 < \mathbf{a}_1 < \mathbf{a}_2 < \dots$ of totally ω-c.a. degrees, with the propery that every $\mathbf{b} \geqslant \mathbf{a}_0$ which is totally ω-c.a. lies below $\mathbf{a}_n$ for some n.

Finally, we see that non-bounding by maximal degree requires at least some complexity.

Theorem 7.13 ([7, 8])**.** *Every superlow degree is bounded by a maximal totally ω-c.a. degree.*

We note that the proofs of some results in this section, for example Theorem 7.11, are of technical interest, since they involve infinitary positive activity at nodes along the true path; this does not occur in [24].

8. Promptness

One fundamental characterisic of our results is that in our lattice embedding results, the bottom degree cannot always be $\mathbf{0}$. For example, consider the bottom degree in the embedding of the 1-3-1 lattice.

The classical proof of Lachlan [50] embeds 1-3-1 with bottom $\mathbf{0}$. The situation is akin to the difference between minimal pairs and branching degrees. Lachlan [51] proved that there are nonzero c.e. degrees that do not bound minimal pairs, whereas Slaman [65] proved that the branching degrees are dense.

The two natural classes of c.e. degrees which bound minimal pairs are the high degrees, as proven by Cooper [20], and the promptly simple degrees, as proven by Ambos-Spies, Jockusch, Shore and Soare [4]. The situation here is similar. Every high degree bounds a copy of the 1-3-1 lattice. But for an analogue of promptly simple degrees, we need a notion of prompt multiple permitting, at the correct level. Roughly speaking, what we need is that when we attempt to give an α-computable approximation to a function in the degree, not only do we fail to do so, but this failure is witnessed promptly; each time we make a guess, the opponent's function changes within a number of steps which is computably bounded. The details are a wee bit messy, but relatively straightforward; and at the end we get a reasonably robust definition of what it means for a degree to be *promptly not totally α-c.a.*, and similar definitions for promptly not totally $< \alpha$-c.a. and so on. See [24].

Here are two representative results.

Theorem 8.1 ([24]). *Every degree which is promptly array noncomputable computes a pair of Π^0_1 separating classes $\mathcal{C}_1, \mathcal{C}_2$ such that any $X \in \mathcal{C}_1$ and $Y \in \mathcal{C}_2$ form a minimal pair.*

(Compare with Theorem 4.1(2)).

Theorem 8.2 ([24]). *Every degree which is promptly not $< \omega^\omega$-c.a. bounds a copy of the 1-3-1 lattice with bottom* **0**.

We remark that sometimes having the bottom not be **0** significantly simplifies the dynamics of a construction. This is why we suspect the following (which should be straighforward but has not been written down).

Conjecture 8.3. *Every degree which is promptly not $< \omega^\omega$-c.a. bounds a non-computable left c.e. real, all of whose presentations are computable.*

Compare with Theorem 4.2. The dynamics of the original construction [33] of a non-computable left-c.e. real, all of whose presentations are computable, has similar dynamics to the embedding of the 1-3-1, namely $< \omega^\omega$-c.a. permitting; once we allow a non-computable $B <_{\mathrm{T}} \alpha$ bounding the presentations, the dynamics simplify to the level of ω-c.a. permitting.

Similarly, the construction of a wtt minimal pair inside a Turing degree, as mentioned above, strongly resembles the construction of a 1-4-1 lattice; hence we surmise:

Conjecture 8.4. *Every degree which is promptly not $< \omega^\omega$-c.a. bounds two Turing equialent, non-computable c.e. sets whose wtt-degrees form a minimal pair.*

Again, in contrast with Theorem 6.4, we suspect that being promptly not totally ω-c.a. is insufficient.

As with embeddings of 1-3-1, all of these constructions can be performed below any high c.e. degree. It would be interesting to formulate a common generalization of prompt non-low$_2$-ness, and highness.

9. An application to admissible computability

Combined with results of the second author, our work has an application to admissible computability. This is a generalisation of traditional computability to ordinals beyond ω. In [43] it is shown that for any admissible ordinal α, the α-c.a. degrees are not elementarily equivalent to the c.e. degrees. This was done in cases, depending on the proximity of α to ω. In one case the separation between the theories is not natural but relies on coding models of arithmetic. However one result is:

Theorem 9.1 ([43]). *Let $\alpha > \omega$ be an admissible ordinal, and let* **a** *be an incomplete α-c.a. degree. The following are equivalent:*

(1) **a** *computes a cofinal ω-sequence in α.*
(2) **a** *bounds a copy of the 1-3-1 lattice.*
(3) **a** *bounds a critical triple.*

Again, it is the analysis of continuous tracing that underlies this result. The basic idea is the following. Consider again the embedding of a critical triple: as time goes by, a longer and longer entourage is build for a follower. When the follower is realised, the entourage is peeled back (from the end to the beginning), one member at a time. Trying to do this when time goes beyond ω presents a completely new problem: after ω many stages, we will have an entourage of order-type ω, that is, one without a last element. We cannot then peel it back, each step removing only the last element. It turns out that this blockage is fundamental. The only case it might be possible for a degree **a** to bound a copy of the 1-3-1 lattice is if it itself can see that α is far from being a regular cardinal — if it can essentially re-order time and space to order-type ω, so that the construction can be (at least after the fact) seen to have taken ω steps, avoiding infinite entourages. In one direction, effective closed and unbounded sets are used to show that this is necessary. In the other direction, a fine-structural result of Shore's [63] says that an incomplete degree of computable cofinality ω must be high, and can compute a bijection between α and ω. Working below such a degree, we can translate back to ω-computability, and use non-low$_2$ permitting to embed the 1-3-1 lattice (for a technical reason, we cannot quite use high permitting).

To sum, what this says is that once we go beyond ω, the fine distinctions between totally ω-c.a. degrees and totally $< \omega^\omega$-c.a. degrees completely disappear. Combined with the current work, this gives us a single, natural sentence which separates the elementary theory of the c.e. degrees from the theory of the α-c.a. degrees for any admissible $\alpha > \omega$.

Theorem 9.2. *Let $\alpha \geqslant \omega$ be admissible. The following are equivalent:*

(1) *There is an incomplete α-c.a. degree which bounds a critical triple but not the 1-3-1 lattice.*
(2) $\alpha = \omega$.

In closing we wonder if the classes we have introduced will have interesting connections with reverse recursion theory in the sense of understanding the proof theorical strength of constructions in computability theory in weak systems of arithmetic.

References

[1] Bahareh Afshari, George Barmpalias, S. Barry Cooper, and Frank Stephan. Post's programme for the Ershov hierarchy. *J. Logic Comput.*, 17(6):1025–1040, 2007.

[2] Klaus Ambos-Spies, Nan Fang, Nadine Losert, Wolfgang Merkle, and Martin Monath. Array noncomputability: a modular approach. In Preparation.

[3] Klaus Ambos-Spies and Peter A. Fejer. Embeddings of N_5 and the contiguous degrees. *Ann. Pure Appl. Logic*, 112(2-3):151–188, 2001.

[4] Klaus Ambos-Spies, Carl G. Jockusch, Jr., Richard A. Shore, and Robert I. Soare. An algebraic decomposition of the recursively enumerable degrees and the coincidence of several degree classes with the promptly simple degrees. *Trans. Amer. Math. Soc.*, 281(1):109–128, 1984.

[5] Klaus Ambos-Spies, Nadine Losert, and Martin Monath. Array noncomputable left-c.e. reals. In preparation.

[6] Bernard Anderson and Barbara Csima. A bounded jump for the bounded Turing degrees. *Notre Dame J. Form. Log.*, 55(2):245–264, 2014.

[7] Katherine Arthur. Maximality in the α-c.a. degrees. Master's thesis, Victoria University of Wellington, 2016.

[8] Katherine Arthur, Rodney G. Downey, Noam Greenberg, and Daniel Turetsky. The distribution of maximality in the totally α-c.a. degrees. In Preparation.

[9] George Barmpalias. Hypersimplicity and semicomputability in the weak truth table degrees. *Arch. Math. Logic*, 44(8):1045–1065, 2005.

[10] George Barmpalias, Rod G. Downey, and Michael McInerney. Integer valued betting strategies and Turing degrees. *J. Comput. System Sci.*, 81(7):1387–1412, 2015.

[11] George Barmpalias and Rodney G. Downey. Kobayashi compressibility. Submitted.

[12] George Barmpalias, Rodney G. Downey, and Noam Greenberg. Working with strong reducibilities above totally ω-c.e. and array computable degrees. *Trans. Amer. Math. Soc.*, 362(2):777–813, 2010.

[13] George Barmpalias, Andrew E. M. Lewis, and Keng Meng Ng. The importance of Π^0_1 classes in effective randomness. *J. Symbolic Logic*, 75(1):387–400, 2010.

[14] Paul Brodhead, Rod Downey, and Keng Meng Ng. Bounded randomness. In *Computation, physics and beyond*, volume 7160 of *Lecture Notes in Comput. Sci.*, pages 59–70. Springer, Heidelberg, 2012.

[15] John Chisholm, Jennifer Chubb, Valentina S. Harizanov, Denis R. Hirschfeldt, Carl G. Jockusch, Jr., Timothy McNicholl, and Sarah Pingrey. Π^0_1 classes and strong degree spectra of relations. *J. Symbolic Logic*, 72(3):1003–1018, 2007.

[16] Peter Cholak, Richard Coles, Rodney G. Downey, and Eberhard Herrmann. Automorphisms of the lattice of Π^0_1 classes: perfect thin classes and anc degrees. *Trans. Amer. Math. Soc.*, 353(12):4899–4924, 2001.

[17] Peter Cholak, Rodney G. Downey, and Stephen Walk. Maximal contiguous degrees. *J. Symbolic Logic*, 67(1):409–437, 2002.

[18] Peter A. Cholak and Leo A. Harrington. On the definability of the double jump in the computably enumerable sets. *J. Math. Log.*, 2(2):261–296, 2002.

[19] Richard J. Coles, Rodney G. Downey, and Geoff L. Laforte. Notes on wtt-jump and ordinal notations. Manuscript, 1998.

[20] S. Barry Cooper. Minimal pairs and high recursively enumerable degrees. *J. Symbolic Logic*, 39:655–660, 1974.

[21] Adam R. Day. Indifferent sets for genericity. *J. Symbolic Logic*, 78(1):113–138, 2013.

[22] David Diamondstone, Noam Greenberg, and Daniel Turetsky. Natural large degree spectra. *Computability*, 2(1):1–8, 2013.

[23] Rodney G. Downey. Lattice nonembeddings and initial segments of the recursively enumerable degrees. *Ann. Pure Appl. Logic*, 49(2):97–119, 1990.

[24] Rodney G. Downey and Noam Greenberg. A transfinite hierarchy of lowness notions in the computably enumerable degrees, unifying classes, and natural definability. Submitted.

[25] Rodney G. Downey and Noam Greenberg. Totally $< \omega^\omega$ computably enumerable and m-topped degrees. In *Theory and applications of models of computation*, volume 3959 of *Lecture Notes in Comput. Sci.*, pages 46–60. Springer, Berlin, 2006.

[26] Rodney G. Downey and Noam Greenberg. Turing degrees of reals of positive effective packing dimension. *Inform. Process. Lett.*, 108(5):298–303, 2008.

[27] Rodney G. Downey, Noam Greenberg, and Rebecca Weber. Totally ω-computably enumerable degrees and bounding critical triples. *J. Math. Log.*, 7(2):145–171, 2007.

[28] Rodney G. Downey and Denis R. Hirschfeldt. *Algorithmic randomness and complexity*. Theory and Applications of Computability. Springer, New York, 2010.

[29] Rodney G. Downey, Denis R. Hirschfeldt, André Nies, and Frank Stephan. Trivial reals. In *Proceedings of the 7th and 8th Asian Logic Conferences*, pages 103–131, Singapore, 2003. Singapore Univ. Press.

[30] Rodney G. Downey and Carl G. Jockusch, Jr. T-degrees, jump classes, and strong reducibilities. *Trans. Amer. Math. Soc.*, 301(1):103–136, 1987.

[31] Rodney G. Downey, Carl G. Jockusch, Jr., and Michael Stob. Array nonrecursive sets and multiple permitting arguments. In *Recursion theory week (Oberwolfach, 1989)*, volume 1432 of *Lecture Notes in Math.*, pages 141–173. Springer, Berlin, 1990.

[32] Rodney G. Downey, Carl G. Jockusch, Jr., and Michael Stob. Array nonrecursive degrees and genericity. In *Computability, enumerability, unsolvability*, volume 224 of *London Math. Soc. Lecture Note Ser.*, pages 93–104. Cambridge Univ. Press, Cambridge, 1996.

[33] Rodney G. Downey and Steffen Lempp. Contiguity and distributivity in the enumerable Turing degrees. *J. Symb. Logic*, 62(4):1215–1240, 1997.

[34] Rodney G. Downey and Richard A. Shore. Degree-theoretic definitions of the low_2 recursively enumerable sets. *J. Symbolic Logic*, 60(3):727–756, 1995.

[35] Rodney G. Downey and Richard A. Shore. Lattice embeddings below a nonlow_2 recursively enumerable degree. *Israel J. Math.*, 94:221–246, 1996.

[36] Rodney G. Downey and John Stephenson. Avioding effective packing dimension 1 below array noncomputable c.e. degrees. Submitted.

[37] Rodney G. Downey and Michael Stob. Structural interactions of the recursively enumerable T- and w-degrees. *Ann. Pure Appl. Logic*, 31(2-3):205–236, 1986. Special issue: second Southeast Asian logic conference (Bangkok, 1984).

[38] Yuri L. Ershov. A certain hierarchy of sets. I. *Algebra i Logika*, 7(1):47–74, 1968.

[39] Yuri L. Ershov. A certain hierarchy of sets. II. *Algebra i Logika*, 7(4):15–47, 1968.

[40] Yuri L. Ershov. A certain hierarchy of sets. III. *Algebra i Logika*, 9:34–51, 1970.

[41] Santiago Figueira, Joseph S. Miller, and André Nies. Indifferent sets. *J. Logic Comput.*, 19(2):425–443, 2009.

[42] Johanna N. Y. Franklin, Noam Greenberg, Frank Stephan, and Guohua Wu. Anti-complex sets and reducibilities with tiny use. *J. Symbolic Logic*, 78(4):1307–1327, 2013.

[43] Noam Greenberg. The role of true finiteness in the admissible recursively enumerable degrees. *Mem. Amer. Math. Soc.*, 181(854):vi+99, 2006.

[44] Leo Harrington and Robert I. Soare. Post's program and incomplete recursively enumerable sets. *Proc. Nat. Acad. Sci. U.S.A.*, 88(22):10242–10246, 1991.

[45] Shamil Ishmukhametov. Weak recursive degrees and a problem of Spector. In *Recursion theory and complexity (Kazan, 1997)*, volume 2 of *de Gruyter Ser. Log. Appl.*, pages 81–87. de Gruyter, Berlin, 1999.

[46] Carl G. Jockusch, Jr. and David B. Posner. Double jumps of minimal degrees. *J. Symbolic Logic*, 43(4):715–724, 1978.

[47] Bjørn Kjos-Hanssen, Wolfgang Merkle, and Frank Stephan. Kolmogorov complexity and the recursion theorem. *Trans. Amer. Math. Soc.*, 363(10):5465–5480, 2011.

[48] Martin Kummer. Kolmogorov complexity and instance complexity of recursively enumerable sets. *SIAM J. Comput.*, 25(6):1123–1143, 1996.

[49] Stuart A. Kurtz. *Randomness and genericity in the degrees of unsolvability*. Ph.D. Dissertation, University of Illinois, Urbana, 1981.

[50] Alistair H. Lachlan. Embedding nondistributive lattices in the recursively enumerable degrees. In *Conference in Mathematical Logic—London '70 (Proc. Conf., Bedford Coll., London, 1970)*, pages 149–177. Lecture Notes in Math., Vol. 255. Springer, Berlin, 1972.

[51] Alistair H. Lachlan. Bounding minimal pairs. *J. Symbolic Logic*, 44(4):626–642, 1979.

[52] Alistair H. Lachlan and Robert I. Soare. Not every finite lattice is embeddable in the recursively enumerable degrees. *Adv. in Math.*, 37(1):74–82, 1980.

[53] Steffen Lempp and Manuel Lerman. A finite lattice without critical triple that cannot be embedded into the enumerable Turing degrees. *Ann. Pure Appl. Logic*, 87(2):167–185, 1997. Logic Colloquium '95 Haifa.

[54] Steffen Lempp, Manuel Lerman, and D. Reed Solomon. Embedding finite lattices into the computably enumerable degrees—a status survey. In *Logic Colloquium '02*, volume 27 of *Lect. Notes Log.*, pages 206–229. Assoc. Symbol. Logic, La Jolla, CA, 2006.

[55] Manuel Lerman. The embedding problem for the recursively enumerable degrees. In *Recursion theory (Ithaca, N.Y., 1982)*, volume 42 of *Proc. Sympos. Pure Math.*, pages 13–20. Amer. Math. Soc., Providence, RI, 1985.

[56] Donald A. Martin. Classes of recursively enumerable sets and degrees of unsolvability. *Z. Math. Logik Grundlagen Math.*, 12:295–310, 1966.

[57] Michael McInerney. *Topics in Algorithmic Randomness and Computability Theory*. PhD thesis, Victoria University of Wellington, 2016.

[58] Yiannis N. Moschovakis. Many-one degrees of the predicates $H_a\,(x)$. *Pacific J. Math.*, 18:329–342, 1966.

[59] André Nies. Lowness properties and randomness. *Adv. Math.*, 197(1):274–305, 2005.

[60] André Nies. *Computability and randomness*, volume 51 of *Oxford Logic Guides*. Oxford University Press, Oxford, 2009.

[61] André Nies, Richard A. Shore, and Theodore A. Slaman. Interpretability and definability in the recursively enumerable degrees. *Proc. London Math. Soc. (3)*, 77(2):241–291, 1998.

[62] Emil L. Post. Recursively enumerable sets of positive integers and their decision problems. *Bull. Amer. Math. Soc.*, 50:284–316, 1944.

[63] Richard A. Shore. The recursively enumerable α-degrees are dense. *Ann. Math. Logic*, 9(1-2):123–155, 1976.

[64] Richard A. Shore. Natural definability in degree structures. In *Computability theory and its applications (Boulder, CO, 1999)*, volume 257 of *Contemp. Math.*, pages 255–271. Amer. Math. Soc., Providence, RI, 2000.

[65] Theodore A. Slaman. The density of infima in the recursively enumerable degrees. *Ann. Pure Appl. Logic*, 52(1-2):155–179, 1991. International Symposium on Mathematical Logic and its Applications (Nagoya, 1988).

[66] Robert I. Soare. Recursive theory and Dedekind cuts. *Trans. Amer. Math. Soc.*, 140:271–294, 1969.

[67] Clifford Spector. Recursive well-orderings. *J. Symb. Logic*, 20:151–163, 1955.

[68] Frank Stephan and Guohua Wu. Presentations of k-trivial reals and kolmogorov complexity. In *Computability in Europe 2005: New Computational Paradigms*, volume 3526 of *Lecture Notes in Comput. Sci.*, pages 461–469. Springer, 2005.

[69] Stephen M. Walk. *Towards a definition of the array computable degrees*. PhD thesis, University of Notre Dame, 1999.

[70] Yongge Wang. *Randomness and Complexity*. PhD thesis, University of Heidelberg, 1996.

[71] Barry Weinstein. *On Embedding of the Lattice 1-3-1 into the Recursively Enumerable Degrees*. PhD thesis, University of California, Berkeley, 1988.

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

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

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

E-mail address: `greenberg@msor.vuw.ac.nz`